

IBM SPSS Collaboration and
Deployment Services 5 –
Anpassungs-Referenzhandbuch



Hinweis: Lesen Sie vor der Verwendung dieser Informationen und des zugehörigen Produkts die allgemeinen Informationen unter Hinweise auf S. 90.

Diese Ausgabe gilt für IBM SPSS Collaboration and Deployment Services 5 und alle nachfolgenden Versionen und Abwandlungen, bis in neuen Ausgaben anderweitig angegeben.

Screenshots von Adobe-Produkten nachgedruckt mit Genehmigung durch Adobe Systems Incorporated.

Screenshots von Microsoft-Produkten nachgedruckt mit Genehmigung durch Microsoft Corporation.

Lizenziertes Material – Eigentum von IBM

© **Copyright IBM Corporation 2000, 2012.**

Eingeschränkte Rechte für Mitarbeiter der US-Regierung – Benutzung, Duplizierung und Veröffentlichung beschränkt durch GSA ADP Schedule-Vertrag mit IBM Corp.

Vorwort

IBM® SPSS® Collaboration and Deployment Services ist eine Anwendung auf Unternehmensebene, die die weit verbreitete Verwendung von Vorhersageanalysen ermöglicht. IBM SPSS Collaboration and Deployment Services bietet eine zentrale, sichere und Audit-fähige Speicherung von Analyseeinrichtungen, erweiterte Funktionen für die Verwaltung und Steuerung von Analyseprozessen zur Vorhersage sowie ausgereifte Mechanismen zur Bereitstellung der Ergebnisse der analytischen Verarbeitung für die Endbenutzer. Zu den Vorteilen von IBM SPSS Collaboration and Deployment Services gehören die Sicherung des Werts von Analyseeinrichtungen, die Gewährleistung der Einhaltung von Bestimmungen, die Verbesserung der Produktivität der Analysten sowie die Minimierung der IT-Kosten für die Verwaltung von Analytiken.

Technischer Support

Registrierte Kunden von IBM Corp. können den IBM Corp. Technischen Support in Anspruch nehmen. Kunden können sich an den technischen Support wenden, wenn sie Hilfe bei der Arbeit mit IBM Corp.-Produkten oder bei der Installation in einer der unterstützten Hardware-Umgebungen benötigen. Informationen zum Technischen Support finden Sie auf der IBM Corp.-Website unter <http://www.spss.com> oder wenden Sie sich an Ihr regionales Büro, das Sie auf der IBM Corp.-Website unter <http://www.spss.com/worldwide> finden. Beachten Sie, dass Sie nach Ihrem Namen, dem Namen Ihrer Organisation und Ihrer Seriennummer gefragt werden.

Kundenmeinungen

Ihre Meinung ist uns wichtig. Teilen Sie uns bitte Ihre Erfahrungen mit IBM Corp.-Produkten mit. Senden Sie uns eine E-Mail an suggest@us.ibm.com oder schreiben Sie an: SPSS Inc., Attn: Director of Product Planning, 233 South Wacker Drive, 11th Floor, Chicago, IL 60606-6412.

Inhalt

1	Überblick über Anpassungsmöglichkeiten	1
	Voraussetzungen	1
2	IBM SPSS Collaboration and Deployment Services Deployment Portal-Anpassung	3
	Anpassen der Benutzeroberfläche	3
	Anpassen der Authentifizierung	4
3	URL-Parameter	6
	URL-Basispfad	6
	Abfragestring	6
	Häufige Parameter	7
	Berichtparameter	13
	Scoring-Parameter	16
	Parameter für benutzerdefinierte Dialogfelder	19
	HTML-Verfahren.	27
4	IBM SPSS Collaboration and Deployment Services Tag Library	30
	JavaServer Pages-Architektur	31
	Unterstützte Elemente	32
	Berichte	32
	Jobs	33
	Scoring-Modelle	33
	Benutzerdefinierte Dialogfelder	34
	Aufbauen einer Anwendung	35
	Implementierungsdetails	38
	Öffentliche JavaScript-API	39
	Die Funktion "runRepositoryItem"	39
	Die Funktion "getBookmarkedValues"	40
	Die Funktion "retrievePromptValues"	40

IBM SPSS Collaboration and Deployment Services Tag Library-Tag-Referenz	41
Das Tag "credential"	42
Das repositoryItem-Tag	44
Das Tag "repositoryItemPrompt"	50
Das Tag "report"	51
Das Tag "reportPrompt"	51
Das Tag "outputLocation"	51
Das Tag "sourceLinkPrompt"	53
Das Tag "sourceLinkRepositoryItem"	56
Das Tag "sourceLinkReport"	57
Das Tag "sourceLinkVariable"	57
Das Tag "actionHandler"	58
Das Tag "actionParameter"	59
Tab-Bibliothek-Beans	60
Credential-Bean	61
ReportBean-Bean	61
SearchBean bean	62
PevMetaDataBean-Bean	63
ScoringBean-Bean	64
JavaServer Pages-Beispiele.	66

5 Portalintegration 67

Installation	68
Konfiguration	68

6 HTML-Archiv 74

Dateistruktur	74
Erstellung von HTMLC-Dateien	75
Beispiel für eine benutzerdefinierte HTMLC-Datei	75

7 Anpassungsbeispiel 76

IBM SPSS Collaboration and Deployment Services Tag Library	76
Berichtdefinitionen	76
Ausführen von BIRT Report Designer for IBM SPSS-Berichten	77
Ausführen von Visualisierungsberichten	80

Javascript-API	80
Interaktivität von Visualisierungsberichten.	81
URL-Fragmente.	82
Framework "Tab Extension"	82
8 Erstellen von benutzerdefinierten Datenservicetreibern	84
Datenservice-API.	84
Erstellen eines benutzerdefinierten Treibers	86
Beispiel eines benutzerdefinierten Treibers	86
Anhang	
A Hinweise	90
Index	93

Überblick über Anpassungsmöglichkeiten

IBM® SPSS® Collaboration and Deployment Services Deployment Portal dient als Thin-Client-Schnittstelle mit dem IBM® SPSS® Collaboration and Deployment Services Repository und ermöglicht jedem Benutzer mit einem Browser und gültigen Anmeldeinformationen die Arbeit mit Inhalten, die im Repository gespeichert sind. Das Standard-Erscheinungsbild und die Standard-Funktionen sind jedoch unter Umständen nicht für alle Benutzer optimal. So kann es beispielsweise erforderlich sein, das Erscheinungsbild der Browser-Oberfläche zu ändern, damit es besser auf den Unternehmensstandard abgestimmt ist. Außerdem können Sie Ihre eigene Benutzeroberfläche für Repository-Inhalte erstellen.

IBM® SPSS® Collaboration and Deployment Services bietet eine Vielzahl an Anpassungsverfahren für die Interaktion mit Inhalten, die im Repository gespeichert sind. Hier einige Beispiele:

- Bearbeiten von Paketkomponenten, wie beispielsweise Bildern und Stylesheets, um das Erscheinungsbild von Deployment Portal zu steuern. Weitere Informationen finden Sie unter [IBM SPSS Collaboration and Deployment Services Deployment Portal-Anpassung](#).
- Verweisen auf Repository-Inhalte direkt über URL-Parameter (Uniform Resource Locators). Weitere Informationen finden Sie unter [URL-Parameter](#).
- Erstellen von benutzerdefinierten Webseiten basierend auf Informationen aus Berichten und Abfragen, die im Repository gespeichert sind, über Java Server Page-Tags. Weitere Informationen finden Sie unter [IBM SPSS Collaboration and Deployment Services Tag Library](#).
- Einbetten von Repository-Inhalten, wie beispielsweise Berichten, in Portalseiten. Weitere Informationen finden Sie unter [Portalintegration](#).
- Durchführen von Batch-Verarbeitungsvorgängen für Repository-Inhalte mithilfe von Python-Skripts. Weitere Informationen finden Sie in der IBM® SPSS® Collaboration and Deployment Services - Essentials for Python-Dokumentation.

Voraussetzungen

Zur korrekten Verarbeitung von benutzerdefinierten Dialogen müssen die folgenden Anforderungen erfüllt sein:

- Ein Fernausführungsserver für IBM® SPSS® Statistics muss in IBM® SPSS® Collaboration and Deployment Services Deployment Manager eingerichtet und dann als Standardserver für die Ausführung von benutzerdefinierter Dialogsyntax im browserbasierten Deployment Manager festgelegt werden. Es ist auch möglich, individuelle benutzerdefinierte Dialogfelder zur Verwendung eines bestimmten SPSS Statistics-Servers zu konfigurieren, der vom Systemstandard abweicht.

- Der Benutzer muss über die Berechtigung *Benutzerdefinierte Dialogfelder ausführen* verfügen, um benutzerdefinierte Dialogfelder ausführen zu können.
- SPSS Statistics-Save-Dateizugriff wird vom SPSS Statistics Data File Driver Service aktiviert; dieser muss als Treiber für SPSS Statistics-Daten mit browserbasiertem IBM® SPSS® Collaboration and Deployment Services Deployment Manager installiert, ausgeführt und angegeben sein. IBM Corp.-Kunden können sich diese Software herunterladen.

Wichtig! SPSS Statistics Data File Driver Service muss auf einem Host mit demselben Betriebssystemtyp ausgeführt werden wie der Repository-Host. Es ist z. B. nicht möglich, ein Repository auf einem Linux-Server in Verbindung mit Data File Driver Service auf einem Windows-Server zu verwenden. Informationen zu IBM® SPSS® Collaboration and Deployment Services-Systemkonfiguration und -Aktionen finden Sie in der Administratordokumentation.

IBM SPSS Collaboration and Deployment Services Deployment Portal-Anpassung

Administratoren können bestimmte Elemente der IBM® SPSS® Collaboration and Deployment Services Deployment Portal-Benutzeroberfläche anpassen, indem sie verschiedene Dateien in einem Repository-Paket (*peb-webcontent.package*) ändern und das Paket mithilfe des IBM® SPSS® Collaboration and Deployment Services Package Manager erneut bereitstellen. Erfahrungen im Umgang mit Stylesheets (.css) sind empfehlenswert.

Das System kann außerdem so konfiguriert werden, dass es einen benutzerspezifischen Authentifizierungsmechanismus verwendet, damit beim Zugriff auf das System über einen Browser keine Anmeldeinformationen manuell eingegeben werden müssen.

Anpassen der Benutzeroberfläche

So passen Sie die IBM® SPSS® Collaboration and Deployment Services Deployment Portal-Benutzeroberfläche an:

1. Kopieren Sie im Verzeichnis *staging* Ihrer Repository-Installation die Datei *peb-webcontent.package* und speichern Sie sie in einem separaten Verzeichnis. Verwenden Sie diese Datei als Sicherungskopie für den Fall, dass Sie später wieder zur Standard-Benutzeroberfläche zurückkehren möchten.
2. Öffnen Sie im Verzeichnis *staging* Ihrer Repository-Installation das Paket *peb-webcontent.package* mit einem Komprimierungsdienstprogramm und extrahieren Sie seinen Inhalt in ein temporäres Verzeichnis. Der extrahierte Inhalt umfasst die Verzeichnisse *images* und *css*, die den Großteil der verwendeten Grafiken und Stylesheets enthalten.
3. Ändern oder ersetzen Sie im extrahierten Verzeichnis *images* beliebige Grafiken wie gewünscht. Um die Darstellung von Seiten zu bearbeiten, ändern Sie den Inhalt der Datei *spssStyles.css* mit einem einfachen Texteditor. Berücksichtigen Sie folgende Vorschläge als Hilfestellung beim Ändern der Darstellung dieser Elemente:
 - Um zu ermitteln, welches Bild oder welcher Stil bearbeitet werden muss, laden Sie die Webseite in einem Browser und verwenden Sie die Funktion *Quelle anzeigen*. Die HTML-Quelle ermittelt die verwendeten Bilder und Stile.
 - Möglicherweise werden einige Bilder von Regeln in der Datei *css* angegeben.
 - Geben Sie die Schriftgröße 0 pt an, um den Produktnamen im Banner zu unterdrücken.
 - Ersetzen Sie Bilder mit transparenten Bildern, um diese zu unterdrücken.
4. Öffnen Sie im extrahierten Verzeichnis *config* die Datei *UIConfig.xml* in einem Texteditor. Ändern Sie die Einstellungen, um bestimmte Elemente der Benutzeroberfläche wie gewünscht

zu unterdrücken. Speichern und schließen Sie die Datei zum Abschluss. Folgende Elemente werden häufig unterdrückt:

- `Footer`: Die untere Fußzeile, die das “Powered by”-Logo enthält (standardmäßig unterdrückt). Ändern Sie den Wert von `false` in `true`, um das Element einzublenden.
- `FileHeader`: Die graue Dateiinformationsleiste, die den Dateinamen und das letzte Änderungsdatum samt Uhrzeit enthält und beim Anzeigen eines Dokuments eingeblendet wird. Ändern Sie den Wert von `true` in `false`, um das Element zu unterdrücken.

```
<component-configuration>
  <component-name>Footer</component-name>
  <component-visible>false</component-visible>
</component-configuration>

<component-configuration>
  <component-name>FileHeader</component-name>
  <component-visible>true</component-visible>
</component-configuration>
```

5. Komprimieren Sie mithilfe eines Komprimierungsdienstprogramms alle Ordner und angepassten Dateien, die zuvor extrahiert wurden. Speichern Sie die Datei als *peb-webcontent.package*. Stellen Sie sicher, dass Sie die Dateierweiterung *.package* und nicht *.zip* verwenden.
6. Installieren Sie mithilfe von IBM® SPSS® Collaboration and Deployment Services Package Manager die angepasste Datei *peb-webcontent.package*, die Sie gespeichert haben. Hinweise finden Sie in der IBM SPSS Collaboration and Deployment Services Package Manager-Dokumentation im IBM® SPSS® Collaboration and Deployment Services Repository-Installations- und Konfigurationshandbuch. Melden Sie sich nach der Fertigstellung ab und schließen Sie das Dienstprogramm.
7. Starten Sie den Repository-Server neu.
8. Warten Sie mehrere Minuten und öffnen Sie dann Deployment Portal (<http://<hostname>:<port>/peb>), um Ihre Änderungen zu überprüfen.

Anpassen der Authentifizierung

Falls an Ihrem Standort die Einzelanmeldung konfiguriert und aktiviert ist, ist für den Zugriff auf die IBM® SPSS® Collaboration and Deployment Services Deployment Portal keine manuelle Eingabe von Anmeldeinformationen erforderlich. Falls die Einzelanmeldung nicht aktiviert ist, kann das System jedoch immer noch so angepasst werden, dass eine manuelle Eingabe von Anmeldeinformationen nicht erforderlich ist.

Die Architektur umfasst eine Java-Schnittstelle mit der Bezeichnung `AuthenticationCriteriaInterface`. Diese Schnittstelle umfasst die folgenden Methoden:

```
public String getUsername()
public String getPassword()
public String getProvider()
```

Um die Authentifizierung anzupassen, müssen Sie zuerst eine Java-Klasse erstellen, die diese Schnittstelle implementiert, beispielsweise *com.spss.AuthenticationCriteriaImpl*. Ihre Klasse muss den Benutzernamen, das Passwort und die Authentifizierungsinformationen des Anbieters angeben. Diese Werte können über verschiedene Quellen bereitgestellt werden, z. B. über eine

zuvor authentifizierte Intranet-Site oder ein Portal. Die Klasse sollte im Klassenpfad für den Anwendungsserver platziert werden.

Konfigurieren Sie das System anschließend so, dass es Ihre Klasse verwendet. Ändern Sie im browserbasierten IBM® SPSS® Collaboration and Deployment Services Deployment Manager die folgenden Deployment Portal-Konfigurationseinstellungen:

- *Konfigurierte Authentifizierungskriterienklasse.* Geben Sie den Namen Ihrer benutzerdefinierten Klasse an.
- *Konfigurierte Authentifizierungskriterien verwenden.* Wählen Sie diese Option, um die Verwendung Ihrer Klasse zu aktivieren.

Nach der Konfiguration Ihrer Authentifizierungsklasse sollten Benutzer in der Lage sein, die Deployment Portal ohne Eingabe von Anmeldeinformationen aufzurufen, vorausgesetzt die von der benutzerdefinierten Klasse bereitgestellten Anmeldeinformationen sind korrekt. Darüber hinaus müssen URL-Referenzierungen auf Repository-Elemente keine Anmeldeinformationen enthalten.

URL-Parameter

Sie können über direkte URLs (Uniform Resource Locators) auf IBM® SPSS® Collaboration and Deployment Services Deployment Portal-Berichte und andere Objekte im Repository zugreifen. Mit URLs können Sie Berichtinformationen auf unterschiedliche Arten direkt freigeben. So können Sie sie beispielsweise in Ihre externen Websites und Anwendungen einbetten. Dieses Referenzdokument führt unterschiedliche URL-Parameter auf und enthält einige Tipps zur Erstellung und Verwendung von Deployment Portal-URL-Abfragestrings. Hilfe erhalten Sie beim technischen Support.

Die in diesem Dokument behandelten URL-Parameter haben nichts mit den URLs zu tun, die in IBM® SPSS® Collaboration and Deployment Services Deployment Manager verfügbar sind.

URL-Basispfad

Der Basispfad für alle Anforderungen lautet:

`http://<Hostname>:<Port>/peb/view/<Pfad zum Content-Repository>`

oder

`http://<Hostname>:<Port>/peb/view?id=<Objekt-ID>`

Dabei gilt:

- `<Hostname>` ist der Name oder die IP-Adresse des Repository-Servers.
- `<Port>` ist die Portnummer für die Verbindung mit dem Repository.
- `<Pfad zum Content-Repository>` ist der Ressourcenpfad des Repository-Objekts, das behandelt werden soll.
- `<Objekt-ID>` ist die Ressourcen-ID des Repository-Objekts, das behandelt werden soll.

Beispiele

`http://yourserver:8080/peb/view/sample/employee.dbq`

`http://yourserver:8080/peb/view?id=0a58c3461e885d240000010f4cc607188375`

Abfragestring

An den Basispfad der URL-Referenzierung kann sich ein Abfragestring mit Parametern anschließen, die zusätzliche Verarbeitungsinformationen enthalten. Der Abfragestring beginnt mit einem Fragezeichen und enthält Parameter-/Wertepaare, die durch das Kaufmanns-Und (&) getrennt sind. Beachten Sie Folgendes: wenn ein Repository-Element von seinem Ressourcenbezeichner referenziert wird, ist das den Abfragestring einleitende Fragezeichen für den Parameter `id` bereits vorhanden und sollte für andere Parameter nicht wiederholt werden.

Ein URL muss mindestens den Content-Repository-Pfad im Basispfad oder den Parameter `id` enthalten. Andere Parameter sind optional. Sofern nicht anders angegeben, werden für Parameter und ihre Werte Groß- und Kleinschreibung unterschieden. Einige Parameter wie `username` und `password` werden praktisch in allen URL-Abfragen verwendet, wohingegen die Verwendung anderer Parameter vom in der Abfrage referenzierten Elementtyp abhängig sein kann. Beachten Sie, dass das System so konfiguriert sein kann, dass es einen benutzerspezifischen Authentifizierungsmechanismus verwendet, damit im Abfragestring keine sicheren Anmeldeparameter angegeben werden müssen. [Für weitere Informationen siehe Thema Anpassen der Authentifizierung in Kapitel 2 auf S. 4.](#)

Reservierte Zeichen wie `&` und ausgeschlossene US-ASCII-Zeichen wie `#` sollten URL-codiert werden, bevor sie als Parameterwert im Abfragestring verwendet werden. Zeichen im reservierten Satz sind jedoch nicht in allen Kontexten reserviert. Im Allgemeinen ist ein Zeichen reserviert, wenn sich die Semantik des URI ändert, wenn das Zeichen mit seiner vorangestellten US-ASCII-Codierung ersetzt wird. Daher sind einige Zeichen (wie `?`, `=` und `:`) nicht in den Parameterwerten reserviert, während Zeichen wie `&` und `#` es sind und folglich URL-codiert werden müssen.

Das Zeichen `&` sollte beispielsweise als `%26` URL-codiert werden. Beispiel für die Angabe des folgenden URL:

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=testuser&admin
```

sollte angegeben werden als

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=testuser%26admin
```

In den folgenden Abschnitten wird jeder Parameter einzeln beschrieben.

Häufige Parameter

Häufige Parameter werden in praktisch allen URL-Referenzierungen oder für mehrere Typen von Repository-Elementen verwendet.

Der Parameter "id"

Der Parameter `id` legt den Repository-Bezeichner für das Element fest, das behandelt werden soll.

Syntax

```
id=<Bezeichner>
```

Der Wert von `<Bezeichner>` entspricht dem Bezeichner des Repository-Objekts.

Beispiele

```
http://yourserver:8080/peb/view?id=0a58c3461e885d240000010f4cc607188375
```

Der Parameter "version"

Der Parameter `version` legt die Version des Repository-Objekts fest, das unter Verwendung des Versionskennzeichens bzw. des Versionslabels behandelt werden soll. Sonderzeichen (z. B. Leerzeichen) muss ein umgekehrter Schrägstrich vorangestellt werden. Lassen Sie diesen Parameter weg, um die AKTUELLSTE Version anzuzeigen.

Syntax

```
version=m.<Versionskennzeichen>  
version=l.<Label>
```

Der Wert von `<Versionskennzeichen>` entspricht der Version des Repository-Objekts. Alternativ kennzeichnet der Wert von `<Label>` das Versionslabel des Repository-Objekts.

Beispiele

```
http://yourserver:8080/peb/view/sample/employee.dbq?version=m.1:2006-12-04%2020:39:17.995  
http://yourserver:8080/peb/view/sample/employee.dbq?version=l.firstVersion
```

Der Parameter "username"

Der Parameter `username` definiert den Benutzer, mit dem die Anmeldung beim System erfolgt.

Syntax

```
username=<Benutzer-ID>  
  
username=<Benutzer-ID>
```

Der Wert von `<Benutzer-ID>` ist die Benutzer-ID der Person, die sich beim Repository-Server anmeldet.

Beispiel

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=validUser
```

Der Parameter "password"

Der Parameter `password` definiert das Passwort, mit dem die Anmeldung beim System erfolgt.

Syntax

```
password=<password>
```

Der Wert von `<Passwort>` ist das Passwort der Person, die sich beim Repository-Server anmeldet.

Beispiel

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=validUser&password=pass
```

Der Parameter "provider"

Der Parameter `provider` definiert den Sicherheits-Provider, mit dem die Anmeldeinformationen validiert werden. Für `provider` muss ein Wert angegeben werden, falls die Parameter `username` und `password` verwendet werden.

Syntax

`provider=<provider>`

Der Wert von `<Provider>` gibt den Sicherheits-Provider an. Gültige Werte sind:

- *Nativ* für den integrierten Anbieter.
- *AD_<name>/<domain>* für Active Directory, wobei `<name>` dem Namen des Sicherheits-Providers im System und `<domain>` dem DNS-Namespace entspricht
- *ADL_<name>/<domain>* für Active Directory mit lokaler Überschreibung, wobei `<name>` dem Namen des Sicherheits-Providers im System und `<domain>` dem DNS-Namespace entspricht
- *iSeries* für IBM i
- *ldap_<name>* für OpenLDAP, wobei `<name>` dem Namen des Sicherheits-Providers im System entspricht

Sonderzeichen (z. B. Leerzeichen) muss ein umgekehrter Schrägstrich vorangestellt werden.

Beispiel

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=validUser&password=pass&provider=Native
```

Der Parameter "promptstate"

Der Parameter `promptstate` legt fest, ob das Dialogfeld "Prompt", das zur Eingabe von Variablenwerten auffordert, die nicht im Abfragestring angegeben werden, unterdrückt werden soll.

Syntax

`promptstate=<x>`

Der Wert *1* für `<x>` unterdrückt das Dialogfeld mit der Eingabeaufforderung während der Laufzeit und verwendet den angegebenen Variablenwert für angeforderte Variablen, die nicht angegeben wurden. Beim Wert *2* wird das Dialogfeld mit der Eingabeaufforderung für alle angeforderten Variablen angezeigt, die nicht angegeben wurden. Alternativ können Sie diesen Parameter weglassen, um die Anzeige des Eingabeaufforderungs-Dialogfelds zuzulassen.

Beispiel

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=validUser&password=pass&provider=Native&fragment=true&outputtype=html&var_EmployeeID=1&promptstate=1
```

Der Parameter "waitstate"

Der Parameter `waitstate` legt fest, ob das Fenster "Warten" angezeigt wird, wenn ein Bericht ausgeführt wird.

Syntax

`waitstate=<x>`

Beim Wert *1* für `<x>` wird das Fenster "Warten" unterdrückt. Lassen Sie diesen Parameter weg, um das Fenster "Warten" anzuzeigen.

Beispiel

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=validUser
&password=pass&provider=Native&fragment=true&outputtype=html
&var_EmployeeID=1&promptstate=1&waitstate=1&fragment=true
```

Der Parameter "partId"

Der Parameter `partId` identifiziert einen bestimmten Teil des referenzierten Repository-Objekts. Bei HTMLC-Dateien kann dieser Parameter eine bestimmte Datei innerhalb des Archivs referenzieren. Bei IBM® SPSS® Statistics-Ausgabedateien (*.spw*) entspricht der Parameter dem Index gemäß der Anzeige in der Gliederung der Datei. Geben Sie also `partId=0` an, um den ersten Teil abzurufen.

Syntax

`partId=<Referenz-ID>`

`<Referenz-ID>` kann folgende Werte annehmen:

- Den relativen Pfad und den Namen einer Datei innerhalb einer HTMLC-Datei.
- Den Index der gewünschten Ausgabe in einer *.spw*-Datei

Beispiel

```
http://yourserver:8080/peb/view/output.htmlc?username=validUser
&password=pass&provider=Native&partId=img/chart.png
```

```
http://yourserver:8080/peb/view/output.spw?username=validUser
&password=pass&provider=Native&partId=1
```

Der Parameter "outputtype"

Der Parameter `outputtype` legt den Dateityp des Ergebnis-Sets fest.

Syntax

`outputtype=<Dateityp>`

dabei entspricht `<Dateityp>` einem der Werte in folgender Tabelle.

Tabelle 3-1
Ausgabetypen

Berichtstyp	Wert	Rückgabe
IBM® ShowCase®	<i>html</i>	HTML
	<i>pdf</i>	Portable Document Format
	<i>wk4</i>	Lotus 1-2-3
	<i>Text</i>	Text
	<i>csv</i>	Durch Kommas getrennte Werte
	<i>biff8</i>	Microsoft Excel
	<i>xlsx</i>	Microsoft Excel 2007 XML-Format
	<i>rptdocument</i>	BIRT Report Designer for IBM® SPSS®-Berichtdokument
BIRT Report Designer for IBM SPSS	<i>HTML</i>	HTML
	<i>Excel 97/2003</i>	Microsoft Excel
	<i>PowerPoint</i>	Microsoft PowerPoint
	<i>Word-Dokument</i>	Microsoft Word
	<i>PDF</i>	Portable Document Format
	<i>PDF - An Seitenbreite anpassen</i>	Portable Document Format mit Breiten-Vergrößerung
	<i>PDF - Nur Seitennummerierung bei Seitenumbruch</i>	Portable Document Format mit Seitennummerierung bei Seitenumbruch
	<i>PDF - In ganze Seite einpassen</i>	Portable Document Format mit Seitenvergrößerung
	<i>PostScript</i>	PostScript-Format
	<i>PostScript - An Seitenbreite anpassen</i>	PostScript-Format mit Breiten-Vergrößerung
	<i>PostScript - Nur Seitennummerierung bei Seitenumbruch</i>	PostScript-Format mit Seitennummerierung bei Seitenumbruch
	<i>PostScript - In ganze Seite einpassen</i>	PostScript-Format mit Seitenvergrößerung
	<i>BIRT-RPT-Dokument</i>	BIRT Report Designer for IBM SPSS-Berichtdokument
	<i>Nachfragen</i>	Ein Prompt, der den Benutzer während der Laufzeit zur Angabe eines Ausgabeformats auffordert
Visualisierung	<i>png</i>	Portable Network Graphics-Format
	<i>emf</i>	Enhanced Metafile-Format
	<i>jpeg</i>	JPEG
	<i>html</i>	HTML. Dies ist für Visualisierungsberichte nur ein gültiges Ausgabeformat, wenn es sich bei der Ausgabe um eine Tabelle handelt. Wenn HTML als Format für einen Visualisierungsbericht verwendet wird, der keine Tabelle erzeugt,

Berichtstyp	Wert	Rückgabe
		wird die Ausgabe in ein PNG-Bild konvertiert.
	<i>pdf</i>	PDF
	<i>Nachfragen</i>	Ein Prompt, der den Benutzer während der Laufzeit zur Angabe eines Ausgabeformats auffordert
Benutzerdefinierte Dialogfelder	<i>SPW</i>	IBM® SPSS® Statistics-Web-Ausgabe-Viewer
	<i>HTML</i>	HTML

Beispiel

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=validUser
&password=pass&provider=Native&fragment=true&outputtype=html
```

Der Parameter "format"

Der Parameter `format` legt fest, ob die ursprüngliche, im Repository gespeicherte Datei zurückgegeben oder ausgeführt werden soll.

Syntax

```
format=raw
```

Der Wert `raw` gibt die ursprüngliche Datei zurück. Beispielsweise wird bei IBM® ShowCase® Query-Definitionen über den Parameter `format` die ursprüngliche `*.dbq`-Datei zurückgegeben und die Abfrage nicht dynamisch ausgeführt.

Beispiel

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=validUser
&password=pass&provider=Native&format=raw
```

Der Parameter "fragment"

Der Parameter `fragment` legt fest, ob die IBM® SPSS® Collaboration and Deployment Services Deployment Portal-Benutzeroberflächenelemente (also die Kopf- und Fußzeile und der Content-Repository-Baum) mit den Berichtsergebnissen angezeigt werden sollen.

Syntax

```
fragment=true
```

Der Wert `true` unterdrückt die Elemente der Deployment Portal-Oberfläche. Lassen Sie diesen Parameter weg, um die Oberfläche anzuzeigen.

Beispiel

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=validUser
&password=pass&provider=Native&fragment=true
```

Parameter für Variablen

Bei nicht für einen Bericht bestimmten Repository-Elementen, die Variablen verwenden (z. B. Jobs), kann der Wert einer Variable angegeben werden, indem Name und Wert der Variable im Abfragestring des URL mit angegeben werden. Bei benutzerdefinierten Dialogen, Jobs und Scoring werden Prompts für Variablenwerte für alle oder für keine Variable angezeigt. Dies ist vom Parameter abhängig.

Bei Berichtselementen muss dem Variablennamen das Präfix `var_` vorangestellt werden. [Für weitere Informationen siehe Thema Parameter für Variablen auf S. 15.](#)

Syntax

`<Variable>=<Wert>`

Der Wert von `<Variable>` entspricht dem Namen der Variablen, die erfüllt werden muss. `<Wert>` ist der Eintrag, der verwendet werden muss, um die angegebene Berichtvariable zu erfüllen.

Beispiel

```
http://yourserver:8080/peb/view/sample/myJob?username=validUser  
&password=pass&provider=Native&region=1
```

Berichtparameter

Berichtparameter werden in Referenzen auf im IBM® SPSS® Collaboration and Deployment Services Repository gespeicherte Berichte verwendet. Bei den Berichten kann es sich um Visualisierungsberichte, BIRT Report Designer for IBM® SPSS®-Berichte oder IBM® ShowCase®-Berichte handeln.

Der Parameter "dbcredential_datasourcename"

Der Parameter `dbcredential_datasourcename` legt die Anmeldeinformationen fest, mit denen die Anmeldung in der Datenquelle erfolgt. Dieser Parameter wird verwendet, wenn die Benutzer-ID für die Datenquelle nicht mit der IBM® SPSS® Collaboration and Deployment Services Deployment Portal-Benutzer-ID identisch ist.

Syntax

`dbcredential_<datasourcename>=<Anmelde-ID>`

Der Wert von `<datasourcename>` ist der Name der angegebenen Datenquelle. Der Wert von `<Anmelde-ID>` ist die ID des Anmeldeinformationsobjekts, das für die Verbindung mit der Datenquelle verwendet wird.

Beispiel

```
http://yourserver:8080/peb/view/sample/employee.dbq  
?dbcredential_yourDS=0a58c346cd5b72010000010f3df6d5e28130
```

Der Parameter "dbuser_datasourcename"

Der Parameter `dbuser_datasourcename` legt die Benutzer-ID fest, mit der die Anmeldung in der Datenquelle erfolgt. Er wird verwendet, wenn die Benutzer-ID für die Datenquelle von der Benutzer-ID für IBM® SPSS® Collaboration and Deployment Services Deployment Portal abweicht.

Syntax

`dbuser_<datasourcename>=<Benutzer-ID>`

Der Wert von `<datasourcename>` ist der Name der angegebenen Datenquelle. Der Wert von `<Benutzer-ID>` ist die Benutzer-ID der Person, die sich bei der Datenquelle anmeldet.

Beispiel

`http://yourserver:8080/peb/view/sample/employee.dbq?dbuser_yourDS=sa`

Der Parameter "dbpwd_datasourcename"

Der Parameter `dbpwd_datasourcename` legt das Kennwort fest, mit dem die Anmeldung in der Datenquelle erfolgt. Dieser Parameter wird verwendet, wenn die Benutzer-ID für die Datenquelle nicht mit der IBM® SPSS® Collaboration and Deployment Services Deployment Portal-Benutzer-ID identisch ist.

Syntax

`dbpwd_<datasourcename>=<password>`

Der Wert von `<datasourcename>` ist der Name der angegebenen Datenquelle. Der Wert von `<Password>` ist das Passwort der Person, die sich bei der Datenquelle anmeldet.

Beispiel

`http://yourserver:8080/peb/view/sample/employee.dbq
?dbuser_yourDB=sa&dbpwd_yourDB=sa`

Anmerkung

Wenn der Parameter `dbcredential_datasourcename` angegeben wurde, wird dieser Parameter vor den Parametern `dbuser_datasourcename` und `dbpwd_datasourcename` für die Herstellung einer Verbindung zu der Datenquelle berücksichtigt.

Der Parameter "width"

Der Parameter `width` legt die Breite eines Bilds oder Diagramms fest. Dieser Parameter wird vor allem für Visualisierungsberichte verwendet.

Für Berichte mit Höhen- und Breitenangaben müssen die Parameter `height` und `width` angegeben werden. Wenn einer der Parameter fehlt, wird das Diagramm mit der Standardhöhe und -breite angezeigt.

Syntax

`width=<x>`

Der Wert `<x>` gibt den ganzzahligen Wert für die Breite in Pixeln an.

Beispiel

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=validUser
&password=pass&provider=Native&fragment=true&outputtype=html
&var_EmployeeID=1&promptstate=1&waitstate=1&width=500&height=1000
```

Der Parameter "height"

Der Parameter `height` legt die Höhe eines Bilds oder Diagramms fest. Dieser Parameter wird vor allem für Visualisierungsberichte verwendet.

Für Berichte mit Höhen- und Breitenangaben müssen die Parameter `height` und `width` angegeben werden. Wenn einer der Parameter fehlt, wird das Diagramm mit der Standardhöhe und -breite angezeigt.

Syntax

`height=<x>`

Der Wert `<x>` gibt den ganzzahligen Wert für die Höhe in Pixeln an.

Beispiel

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=validUser
&password=pass&provider=Native&fragment=true&outputtype=html
&var_EmployeeID=1&PROMPTSTATE=1&waitstate=1&width=500&height=1000
```

Der Parameter "var_variable"

Der Parameter `var_variable` legt den Wert fest, der die angegebene Berichtvariable erfüllt.

Syntax

`var_<Variable>=<Wert>`

Der Wert von `<Variable>` ist der Name der Variablen, die erfüllt werden muss. Um den Variablennamen zu suchen, wählen Sie in IBM® ShowCase® Query oder IBM® ShowCase® Report Writer im Menü "Abfrage" die Option Variable. Es wird eine Liste der Variablennamen für den aktuellen Bericht angezeigt.

`<Wert>` ist der Eintrag, der verwendet werden muss, um die angegebene Berichtvariable zu erfüllen.

Beispiel

```
http://yourserver:8080/peb/view/sample/employee.dbq?username=validUser
&password=pass&provider=Native&fragment=true&outputtype=html&var_EmployeeID=1
```

Anmerkungen

- Bei Berichten unterdrückt die Angabe eines Variablenwerts im URL das Prompt-Dialogfeld für diese Variable.
- Um einen einzelnen Variablenwert anzugeben (=), verwenden Sie die Syntax
var_Lastname=Curtis.
- Um mehrere Variablenwerte anzugeben (IN), verwenden Sie die Syntax
var_Lastname=Curtis&var_Lastname=McLind.
- Um einen Bereich von Variablenwerten anzugeben (BETWEEN), verwenden Sie die Syntax
var_Dateship=3-1-2007&var_Dateship=3-31-2007.
- Um Werte für mehrere Variablen anzugeben, verwenden Sie die Syntax
var_Lastname=Curtis&var_Dateship=3-1-2007&var_Dateship=3-31-2007.

Scoring-Parameter

Scoring-Parameter werden bei der Referenzierung von Scoring-Konfigurationen zum Generieren von Scores verwendet.

Der Parameter "dataset"

Der Parameter `dataset` gibt den Ort einer SQL-Daten-Provider-Definition an, die für das Batch-Scoring verwendet werden soll. Der Wert dieses Parameters ist ein relativer Pfad im IBM® SPSS® Collaboration and Deployment Services Repository.

Syntax

```
dataset=<DPD-Speicherort>
dataset.<tableID>=<DPD-Speicherort>
```

Der Wert von `<DPD-Speicherort>` ist der Repository-Pfad zur Daten-Provider-Definition.

Wenn das Daten-Set mehrere Tabellen enthält, hängen Sie an den Parameter einen Punkt, gefolgt von der Tabellen-ID für die gewünschte Tabelle, an. Der Wert von `<tableID>` kann durch Untersuchung des Scoring-Modells in IBM® SPSS® Modeler ermittelt werden.

Beispiel

```
http://yourserver:8080/peb/view/myPMML.xml?username=validUser
&password=pass&scoring_configuration=testConfig
&dataset=/datasets/dataset.sqldpd
```

Der Parameter "dataset_label"

Der Parameter `dataset_label` ermöglicht dem Benutzer die Angabe der entsprechenden Version des Daten-Sets. Die angegebene Version des Daten-Sets muss mit dem in der Scoring-Konfiguration definierten Daten-Provider kompatibel sein. Ohne Angabe wird die *AKTUELLSTE* Version verwendet.

Syntax

```
dataset_label=<meinLabel>  
dataset_label.<tableID>=<meinLabel>
```

Der Wert von `<meinLabel>` ist die Bezeichnung für die gewünschte Version des Daten-Sets.

Wenn das Daten-Set mehrere Tabellen enthält, hängen Sie an den Parameter einen Punkt, gefolgt von der Tabellen-ID für die gewünschte Tabelle, an. Der Wert von `<tableID>` kann durch Untersuchung des Scoring-Modells in IBM® SPSS® Modeler ermittelt werden.

Beispiel

```
http://yourserver:8080/peb/view/myPMML.xml?username=validUser  
&password=pass&scoring_configuration=testConfig  
&dataset=/datasets/dataset.sqldpd&dataset_label=PRODUCTION
```

Der Parameter "dataset_table"

Der Parameter `dataset_table` ermöglicht dem Benutzer die Angabe einer Tabelle innerhalb eines Daten-Sets.

Syntax

```
dataset_table=<meineTabelle>  
dataset_table.<tableID>=<meineTabelle>
```

Der Wert von `<meineTabelle>` ist der Name für die gewünschte Daten-Set-Tabelle.

Wenn das Daten-Set mehrere Tabellen enthält, hängen Sie an den Parameter einen Punkt, gefolgt von der Tabellen-ID für die gewünschte Tabelle, an. Der Wert von `<tableID>` kann durch Untersuchung des Scoring-Modells in IBM® SPSS® Modeler ermittelt werden.

Beispiel

```
http://yourserver:8080/peb/view/myPMML.xml?username=validUser  
&password=pass&scoring_configuration=testConfig&fragment=true&  
dataset=/data/mySet&  
dataset_label=PRODUCTION&  
dataset_rowlimit=2&  
dataset_table=myTable&  
promptstate=1
```

Der Parameter "dataset_rowlimit"

Der Benutzer kann die vom Daten-Set für das Batch-Scoring verarbeitete Datenmenge einschränken. Auf diese Weise lassen sich Prozesse mit langen Laufzeiten vermeiden. Der Parameter `dataset_rowlimit` legt die Anzahl der Datenzeilen fest, die auf dem Daten-Set extrahiert werden.

Syntax

```
dataset_rowlimit=<x>  
dataset_rowlimit.<tableID>=<x>
```

Der Wert `<x>` gibt die Anzahl der aus dem Daten-Set zu extrahierenden Zeilen an.

Wenn das Daten-Set mehrere Tabellen enthält, hängen Sie an den Parameter einen Punkt, gefolgt von der Tabellen-ID für die gewünschte Tabelle, an. Der Wert von `<tableID>` kann durch Untersuchung des Scoring-Modells in IBM® SPSS® Modeler ermittelt werden.

Beispiel

```
http://yourserver:8080/peb/view/myPMML.xml?username=validUser  
&password=pass&scoring_configuration=testConfig  
&dataset=/datasets/dataset.sqldpd&dataset_rowlimit=1000
```

Der Parameter "scoring_configuration"

Der Parameter `scoring_configuration` gibt die Scoring-Konfiguration an, die von der Scoring-Engine für das Scoring des angegebenen Modells verwendet wird.

Syntax

```
scoring_configuration=<Konfig-Name>
```

Der Wert von `<Konfig-Name>` ist der Name der für das Scoring zu verwendenden Scoring-Konfiguration. Die angegebene Konfiguration muss in der Lage sein, eine Scoring-Anforderung zu verarbeiten. Ein Verweis auf eine ausgesetzte Konfiguration kann keine Scores erzeugen.

Beispiel

```
http://yourserver:8080/peb/view/myPMML.xml?username=validUser  
&password=pass&scoring_configuration=testConfig  
&dataset=/datasets/dataset.sqldpd
```

Der Parameter "batch_type"

Der Parameter `batch_type` gibt an, welche Scoring-Eingabeaufforderungen angezeigt werden sollen. Falls der Parameter *Daten-Set* angibt, erzeugt die Scoring-Benutzeroberfläche die Eingabeaufforderungen für Daten-Set und Bezeichnung. Falls der Parameter `batch_type` nicht angegeben wird und keine Parametereingaben definiert werden, wird die auf den Scoring-Parametern basierende Benutzeroberfläche verwendet.

Syntax

batch_type=<Eingabeaufforderung>

Der Wert von <Eingabeaufforderung> gibt die Quelle für die Eingabeaufforderungen an. Derzeit wird nur *Daten-Set* als Quelle unterstützt. Vermeiden Sie diesen Parameter, um den Benutzer zur Eingabe von auf Parametern basierenden Werten aufzufordern.

Beispiel

```
http://yourserver:8080/peb/view/myPMML.xml?username=validUser
&password=pass&scoring_configuration=testConfig&batch_type=dataset
```

Parameter für benutzerdefinierte Dialogfelder

Parameter für benutzerdefinierte Dialogfelder werden zur Referenzierung von Dateien für benutzerdefinierte Dialogfelder (*.spd*) verwendet.

Anmerkung: Für diese Funktionalität sind IBM® SPSS® Statistics-Adapter in der IBM® SPSS® Collaboration and Deployment Services-Umgebung erforderlich. Weitere Informationen finden Sie in der SPSS Statistics-Installationsdokumentation.

Der Parameter "dataset.uri"

Der URI des Daten-Sets, das vom benutzerdefinierten Dialogfeld verwendet werden soll. Für DPDs und *.sav*-Dateien im IBM® SPSS® Collaboration and Deployment Services Repository kann der URI als Repository-Pfad oder die Ressourcen-ID angegeben sein. Wenn der URI eine Datei im Dateisystem referenziert, muss der Pfad zur Datei vom IBM® SPSS® Statistics Data File Driver Server gültig sein, der zum Abrufen der Variablen-Metadaten verwendet wird. Außerdem muss es ein gültiger Pfad auf dem SPSS Statistics-Server sein, der die Syntax ausführen wird. Falls ein Repository-Daten-Set-Objekt verwendet wird, kann die Version des Objekts entweder als Versionserzeuger oder als Bezeichnung an den URI angehängt werden.

Syntax

dataset.uri=<meinURI>

Der Wert von <meinURI> gibt den URI für das Daten-Set an.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd
?dataset.uri=spsscr:///Datasets/SpecificURI.sav
```

```
http://yourserver:8080/peb/view/myDialog.spd
?dataset.uri=spsscr:///?id=0a30063bc975ede40000011cafb8deda8327.
```

```
http://yourserver:8080/peb/view/myDialog.spd
?dataset.uri=file:///C:/Programme/SPSSInc/Samples/accidents.sav
```

Der Parameter "dataset.table"

Für IBM® SPSS® Collaboration and Deployment Services Enterprise View-Datenquellen die Tabelle, die vom benutzerdefinierten Dialogfeld verwendet werden soll. Wenn kein Name angegeben wird, wird der Benutzer aufgefordert, eine Tabelle aus der Liste der in der DPD verfügbaren Tabellen auszuwählen.

Syntax

```
dataset.table=<meineTabelle>
```

Der Wert von <meineTabelle> gibt die zu verwendende Tabelle an.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd  
?dataset.uri=spsscr:///DPDs/myDPD&dataset.table=myTableName
```

Der Parameter "dataset.prompt"

Gibt an, dass der Benutzer gezwungen wird, ein Daten-Set für die benutzerdefinierten Dialogfelder auszuwählen. Anderenfalls wird das Daten-Set, das der Benutzer während der Sitzung für das erste Dialogfeld geöffnet hat und das den Suchkriterien entspricht, für jedes nachfolgende benutzerdefinierte Dialogfeld verwendet, das nicht für die Verwendung eines speziellen Daten-Sets konfiguriert wurde.

Syntax

```
dataset.prompt=<Indikator>
```

Der Wert von <Indikator> ist entweder *true* oder *false*.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd?dataset.prompt=true
```

Der Parameter "dataset.search.criteria"

Die zu verwendenden Suchkriterien für die Erstellung einer Liste von Daten-Sets während der Ausführungszeit. Die gesamte Suchzeichenfolge muss in einer einzigen Zeile eingegeben werden. Mehrere Bedingungen können mit Hilfe von Klammern und Und-/Oder-Logik kombiniert werden.

Suchkriterien

```
$$repository/title_field_name=< Objektname>
```

```
$$search/mimetype=<Objekt-MIME-Typ>
```

```
$$repository/version_created_by_field=<Stempel des Benutzers, der das Objekt erstellt hat>
```

\$\$repository/version_created_date_field=< Datum der Versionserstellung>

\$\$repository/description_field_name=< Objektbeschreibung>

\$\$repository/object_last_modified_by=< Stempel des Benutzers, der das Objekt erstellt hat>

Syntax

dataset.search.criteria=<meineKriterien>

Der Wert von <meineKriterien> entspricht dem Suchausdruck.

Beispiel

```
# lokalisiert alle DPDs
http://yourserver:8080/peb/view/myDialog.spd
?dataset.search.criteria='$$search/mimetype%3Dapplication/x-vnd.spss-data-provider'

# lokalisiert alle SAV-Dateien
http://yourserver:8080/peb/view/myDialog.spd
?dataset.search.criteria='$$search/mimetype%3Dapplication/x-vnd.spss-spss-data%20or%20
$$search/mimetype%3Dapplication/x-vnd.spss-statistics-data'

# lokalisiert alle Dateien, die dem Schlüsselwort SPECIAL_DATASET entsprechen
http://yourserver:8080/peb/view/myDialog.spd
?dataset.search.criteria='$$repository/keyword_field_name%3D%3DSPECIAL_DATASET'
```

Der Parameter “variable.display”

Der Parameter `variable.display` gibt an, ob Variablennamen bzw. -labels angezeigt werden sollen oder nicht.

Syntax

`variable.display=<Typ>`

Der Wert von <Typ> ist entweder *names*, womit Variablennamen angezeigt werden, oder *labels*, womit Variablenlabels angezeigt werden.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd
?dataset.uri=spsscr:///Datasets/SpecificURI.sav&variable.display=labels
```

Der Parameter “variable.sort”

Der Parameter `variable.sort` gibt das zur Sortierung der Variablen verwendete Suchkriterium an.

Syntax

`variable.sort=<meineKriterien>`

<meineKriterien> kann folgende Werte annehmen:

- *none*, um keine zusätzliche Sortierung der ursprünglichen Reihenfolge der Daten durchzuführen
- *alphanumeric* für eine alphanumerische Sortierung von Feldnamen oder -bezeichnungen (je nach Anzeige)
- *measurement* für eine Sortierung nach dem Messniveau des Felds

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd  
?dataset.uri=spsscr:///Datasets/SpecificURI.sav&variable.sort=alphanumeric
```

Der Parameter "stylesheet.url"

Wenn Sie ein CSS-Stylesheet verwenden, das im Repository gespeichert ist, der Repository-URL des Stylesheets.

Syntax

```
stylesheet.url=<meinURL>
```

Der Wert von <meinURL> ist der URL für das Stylesheet.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd  
?stylesheet.url=/peb/view/EditText_pes.css&fragment=true
```

Der Parameter "stylesheet.name"

Wenn Sie ein CSS-Stylesheet verwenden, das in die Datei des benutzerdefinierten Dialogfelds eingebettet ist, der Name des Stylesheets. Das Stylesheet kann der benutzerdefinierten Dialogdatei hinzugefügt werden, indem Komprimierprogramme wie WinZip verwendet werden.

Syntax

```
stylesheet.name=<meineStile>
```

Der Wert von <meineStile> gibt den Namen des Stylesheets an.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd  
?stylesheet.name=EditText.css
```

Der Parameter "javascript.url"

Wenn Sie ein JavaScript verwenden, das im Repository gespeichert ist, der Repository-URL der Skriptdatei.

Syntax

```
javascript.url=<meinURL>
```

Der Wert von <meinURL> ist der URL für die JavaScript-Datei.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd  
?javascript.url=/peb/view/EditBox_pes.js&fragment=true
```

Der Parameter "javascript.name"

Wenn Sie ein JavaScript-Sheet verwenden, das in die Datei des benutzerdefinierten Dialogfelds eingebettet ist, der Name der Skriptdatei.

Syntax

```
javascript.name=<meineDatei>
```

Der Wert von <meineDatei> ist der Name der JavaScript-Datei.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd?javascript.name=EditBox.js
```

Der Parameter "validate.method"

Ein Validierungsverfahren aus der angegebenen JavaScript-Datei, das aufgerufen wird, bevor die Seite angezeigt wird. Das übergebene Formular sollte der einzige Parameter für die Methode sein. Nach Auswertung der Eingaben gibt das Verfahren einen Boole'schen Wert zurück. Die Methode sollte "true" zurückgeben, falls alles korrekt ist, und "false", falls die Übertragung abgebrochen werden soll.

Syntax

```
validate.method=<meineMethode>
```

Der Wert von <meineMethode> ist der Name der für die Validierung zu verwendenden Methode in der JavaScript-Datei.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd?javascript.name=EditBox.js  
&validate.method=myValidate
```

Der Parameter "output.format"

Das Format der zu erzeugenden Ausgabe. Standardformat ist das IBM® SPSS® Statistics-Web-Ausgabe-Viewer-Format (*.spw*). In einigen Fällen kann es sinnvoll sein, stattdessen HTML zu erzeugen. Beim Ausgabeformat wird zwischen Groß- und Kleinschreibung unterschieden.

Dieser Parameter gibt dieselben Informationen wie der Parameter `outputtype` an, wird jedoch nur für benutzerdefinierte Dialogfelder berücksichtigt.

Syntax

```
output.format=<meinFormat>
```

Der Wert von `<meinFormat>` ist das Format für die Ausgabe. Folgende Werte sind gültig:

- *SPW* für den SPSS Statistics-Web-Ausgabe-Viewer
- *HTML* für eine HTML-Ausgabe

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd?output.format=SPW
```

Der Parameter "output.filename"

Der Name der Ausgabedatei. Wenn nicht angegeben, erhält die generierte Ausgabedatei den Namen der Datei des benutzerdefinierten Dialogfelds, jedoch ohne *.spw*-Erweiterung.

Syntax

```
output.filename=<meineDatei>
```

Der Wert von `<meineDatei>` ist der Name für die Ausgabedatei.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd  
?output.filename=MyOutputName.spw
```

Der Parameter "showOutline"

Gibt an, ob die Gliederung angezeigt werden soll. Die Standardeinstellung lautet *true*.

Syntax

```
showOutline=<Indikator>
```

Der Wert von `<Indikator>` ist entweder *true* oder *false*.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd?showOutline=true
```

Der Parameter "allowPivoting"

Gibt an, ob es möglich sein soll, die Tabelle zu bearbeiten. Wenn die Option deaktiviert ist, kann der Benutzer Schichten nicht drehen, vertauschen oder ändern, Ansichten nicht speichern und Daten nicht in einem neuen Fenster öffnen. Die Standardeinstellung lautet *true*.

Syntax

`allowPivoting=<Indikator>`

Der Wert von `<Indikator>` ist entweder *true* oder *false*.

Beispiel

`http://yourserver:8080/peb/view/myDialog.spd?allowPivoting=true`

Der Parameter "allowPrinterFriendly"

Gibt an, ob eine druckerfreundliche Version für eine bestimmte Tabelle geöffnet werden kann. Die Standardeinstellung lautet *true*.

Syntax

`allowPrinterFriendly=<Indikator>`

Der Wert von `<Indikator>` ist entweder *true* oder *false*.

Beispiel

`http://yourserver:8080/peb/view/myDialog.spd?allowPrinterFriendly=true`

Der Parameter "allowDownload"

Gibt an, ob die Daten in eine lokale Datei heruntergeladen werden können. Die Standardeinstellung lautet *true*.

Syntax

`allowDownload=<Indikator>`

Der Wert von `<Indikator>` ist entweder *true* oder *false*.

Beispiel

`http://yourserver:8080/peb/view/myDialog.spd?allowDownload=true`

Der Parameter "showLogs"

Gibt an, ob Protokolleinträge in der Ausgabe angezeigt werden sollen. Die Standardeinstellung lautet *true*.

Syntax

showLogs=<Indikator>

Der Wert von <Indikator> ist entweder *true* oder *false*.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd?showLogs=true
```

Der Parameter "statistics.server"

IBM® SPSS® Statistics-Server, der zur Ausführung der Syntax des benutzerdefinierten Dialogfelds verwendet wird. Der Wert kann ein URI oder ein Name sein, der auf einen in IBM® SPSS® Collaboration and Deployment Services definierten Server verweist. Bei mehreren Servern kann dieser Wert der URI oder Name eines Server-Clusters sein.

Syntax

statistics.server=<Server-ID>

Der Wert von <Server-ID> gibt den für die Ausführung zu verwendenden Server an.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd?
statistics.server=spsscr:///?id=0a30063bc975ede4000011cafb8deda8327

http://yourserver:8080/peb/view/myDialog.spd
?statistics.server=localStatisticsServer

http://yourserver:8080/peb/view/myDialog.spd
?statistics.server=copServerCluster
```

Der Parameter "statistics.server.credential"

Die Anmeldedaten, die verwendet werden sollen, um beim Ausführen der Syntax eine Verbindung zum IBM® SPSS® Statistics-Server aufzubauen. Der Wert kann ein URI oder ein Name sein, der auf Anmeldedaten für IBM® SPSS® Collaboration and Deployment Services verweist.

Syntax

statistics.server.credential=<meineAnmeldeinfos>

Der Wert von <meineAnmeldeinfos> gibt die Anmeldeinformationen an, unter denen die Ausführung stattfindet.

Beispiel

```
http://yourserver:8080/peb/view/myDialog.spd?statistics.server=localStatisticsServer
&statistics.server.credential=spsscr:///?id=0a30063bc975ede4000011cafb8deda8327.

http://yourserver:8080/peb/view/myDialog.spd?statistics.server=localStatisticsServer
&statistics.server.credential=administrator
```

HTML-Verfahren

Verwenden Sie einen HTML-Editor.

Viele HTML-Editoren können die Erstellung von URL-Abfragestrings vereinfachen und die korrekten Trennzeichen zwischen Parameter setzen.

Verwenden Sie HTML-Formulare, um Anforderungen zu übergeben.

IBM® SPSS® Collaboration and Deployment Services Deployment Portal-Anforderungen können über HTML-Formulare übergeben werden, die in eine Webseite integriert sind. Beispielsweise können einem Benutzer folgende Aktionen über ein Formular ermöglicht werden:

- Auswahl aus einer Liste verfügbarer Berichte
- Auswahl eines Ausgabe-Dateityps
- Angabe von abgefragten Variablen vor der Übergabe der Berichts-anforderung
- Angabe einer ID und eines Kennworts für die Ausführung eines Berichts

Das folgende Beispiel referenziert ein benutzerdefiniertes Dialogfeld in der Aktion für ein Formular.

```
<form name='AnalyzeOptions' method='POST' target='iframe_1'
  action='/peb/view/SamplesStatistics/SPD/Simple.spd?fragment=true&promptstate=1&waitstate=1'>
  <input type='hidden' name='username' value='userA'/>
  <input type='hidden' name='password' value='passwordA'/>
  <input type='hidden' name='provider' value='Native'/>
  <input type='hidden' name='dataset.uri' value='spsscr:///SamplesStatistics/SAV/multipleResponseData.sav'/>
  <input type='hidden' name='allowPivoting' value='false'/>

  <input name='PromptParameter1' type='checkbox' value='true'/>
  Check the box to select parameter 1

  <br>
  <input type='submit' value='Run Report' />
</form>
```

Verwenden Sie das IBM SPSS Collaboration and Deployment Services Repository, um benutzerdefinierte Seiten mit relativen Pfaden zu speichern.

Das Repository kann als zentraler Speicherort für alle Dateien für eine benutzerdefinierte Website verwendet werden. Relative und absolute Pfade können auf der benutzerdefinierten Website verwendet werden, um Verknüpfungen mit Objekten wie .css-Stylesheets, Bilder, Deployment Portal-Berichtobjekte und JavaScript herzustellen.

So können Sie beispielsweise einen Ordner *MeineWebseite* im Repository speichern, der die benutzerdefinierte Webseite *MeineWebseite.htm* und Ressourcen wie Bilder, Stylesheets und JavaScript-Dateien enthält. *MeineWebseite.htm* kann **relative** Verweise auf die Ressourcen wie folgende enthalten:

```

```

```
<script language="javascript" src="MyJS.js?fragment=true">
</script>
<LINK REL="StyleSheet" HREF="MyStyles.css?fragment=true"
TYPE="text/css" MEDIA="screen" />
```

Damit derartige relative Verweise ordnungsgemäß funktionieren, muss über den Parameter `fragment=true` im URL auf die Webseite zugegriffen werden. Beispiel:

```
http://yourserver:port/peb/view/MyWebPage/MyWebPage.htm?
username=validUser&password=pass&provider=Native&fragment=true
```

Wenn Sie die Ressourcen für Ihre Website an einem anderen Ort im Repository als dem Ort, an dem auch die Webseite gespeichert ist, speichern möchten, können Sie über **absolute** Pfade Verweise von Ihrer Webseite (zum Beispiel *MeineWebseite.htm*) erstellen:

```

<script language="javascript"
src="/peb/view/MyWebPage/js/MyJS.js?fragment=true">
</script>
<LINK REL="StyleSheet"
HREF="/peb/view/MeineWebseite/CSS/MyStyles.css?fragment=true"
TYPE="text/css" MEDIA="screen" />
```

Oder Sie verweisen über den vollständigen Hostnamen und Port im Pfad darauf:

```

<script language="javascript" src="http://yourserver:8080/peb/view/
MeineWebseite/js/MyJS.js?fragment=true">
</script>
<LINK REL="StyleSheet" HREF="http://yourserver:8080/peb/view/
MeineWebseite/CSS/MyStyles.css?fragment=true" TYPE="text/css"
MEDIA="screen"/>
```

Ein Hinweis zu Drittanbieter-Cookies

Wenn Sie eine HTML-Seite erstellen, die auf veröffentlichte IBM® SPSS® Statistics-Berichte (*.spw*) verweist, die in einen HTML IFRAME eingebettet sind, müssen Sie die Datenschutzeinstellungen des Internet Explorer für Cookies von Drittanbietern anpassen, um zu vermeiden, dass der Anmeldebildschirm angezeigt wird.

Dieses Szenario tritt nur auf, wenn Sie die *.spw*-Berichte von einer externen Website aus aufrufen und dabei IFRAMEs zur Einbettung und Anzeige mehrerer Berichte auf einer Webseite verwendet werden. Beim Zugriff auf die HTML-Seite werden die Berichte ausgeführt und in den IFRAMEs angezeigt. Bei der Einbettung des Deployment Portal-URL-Parameters in einen IFRAME werden die Cookies als Drittanbieter-Cookies betrachtet. In den Datenschutzeinstellungen des Internet Explorer (Extras/Internetoptionen/Datenschutz) sind Drittanbieter-Cookies standardmäßig deaktiviert. Dies führt dazu, dass beim Zugriff auf die Webseite der Anmeldebildschirm angezeigt

wird (selbst wenn die URL den richtigen Benutzernamen und das richtige Kennwort enthält). Sie können dieses Verhalten in den Datenschutzeinstellungen des Internet Explorer ändern. Fügen Sie den Domännennamen bzw. die IP-Adresse von IBM® SPSS® Collaboration and Deployment Services mit "Zulassen" zu den verwalteten Sites hinzu. Dadurch wird sichergestellt, dass sämtliche Cookies von dieser Adresse akzeptiert werden und kein Anmeldebildschirm angezeigt wird.

Im Folgenden sehen Sie ein Beispiel für die HTML, die für den Verweis auf einen im Repository veröffentlichten *.spw*-Bericht verwendet wird.

```
<iframe frameborder=1 src="http://yourserver:8080/peb/view  
/jba/accidents.spw?partId=5&fragment=true&username=admin&password=  
yourpwd&provider=Native" style="WIDTH: 800px;HEIGHT: 280px" name="I1">  
</iframe>
```

Beachten Sie, dass dieses Problem nur bei Internet Explorer auftritt, nicht bei anderen unterstützten Browsern.

IBM SPSS Collaboration and Deployment Services Tag Library

Mit IBM® SPSS® Collaboration and Deployment Services wird für Administratoren und erfahrene Benutzer, die Beziehungen zwischen Repository-Elementen oder benutzerdefinierte Webseiten (*.jsp*-Seiten) mit Elementen erstellen möchten, die sich gegenseitig mit Werten versorgen können, eine JavaServer Pages (JSP) -Tag-Bibliothek bereitgestellt. Die Tag-Bibliothek bietet folgende grundlegenden Funktionen:

Authentifizierung: Sie können den Benutzer, das Passwort und den Sicherheits-Provider festlegen und Elemente oder Prompts freigeben, die auf der Seite definiert sind. Für den Zugriff auf Elemente im IBM® SPSS® Collaboration and Deployment Services Repository und für die Datenquellen-Authentifizierung ist eine Authentifizierung erforderlich.

Elemente: Sie können die Definition von Elementen einschließlich des Ziel-”Containers” (<div>- oder <iframe>-Element) festlegen. Die Elemente werden für IFRAME-Ziele über eine POST-Anforderung und für DIV-Ziele über eine AJAX-Anforderung (Asynchronous JavaScript und XML) ausgeführt.

Prompts: Sie können Prompts verwenden, um die Parameter für die Ausführung von Elementen dynamisch anzupassen. Die Prompt-Position ist nur auf eine Position auf der aktuellen Seite beschränkt. Prompts können entweder benutzerdefiniert oder ein ausgewählter Parameter aus einer vorhandenen Elementdefinition sein.

Verknüpfungsbeziehungen: Sie können Beziehungen zwischen folgenden Elementen definieren:

- Quell- und Zielberichtelementen, Job-, Scoring- oder benutzerdefinierten Dialogfeldobjekten
- Einer Liste von Prompts und einem Zielelement. Sowohl der Aktivierungsspeicherort (DIV oder IFRAME) als auch der Zeitpunkt (ONDEMAND, ONLOAD oder NONE) werden unterstützt.

Die Tag-Bibliothek setzt sich aus den folgenden wesentlichen Bestandteilen zusammen:

- der öffentlichen JavaScript-API
- benutzerdefinierten Tags und ihren Interaktionen untereinander
- den Tag-Bibliothek-Beans für den Abruf von Daten-Sets [Für weitere Informationen siehe Thema Tab-Bibliothek-Beans auf S. 60.](#)

Dieses Dokument beschreibt jede Tag-Funktion, die in der JSP-Tag-Bibliothek verfügbar ist, und enthält einige Verwendungsbeispiele. Wir empfehlen, nach dem Lesen dieses Dokuments die Beispiel *.jsp*-Dateien durchzusehen, die mit der Tag-Bibliothek bereitgestellt wurden, bevor Sie beginnen, eigene benutzerdefinierte Seiten zu erstellen. [Für weitere Informationen siehe Thema JavaServer Pages-Beispiele auf S. 66.](#)

Aktualisieren auf IBM SPSS Collaboration and Deployment Services Tag Library

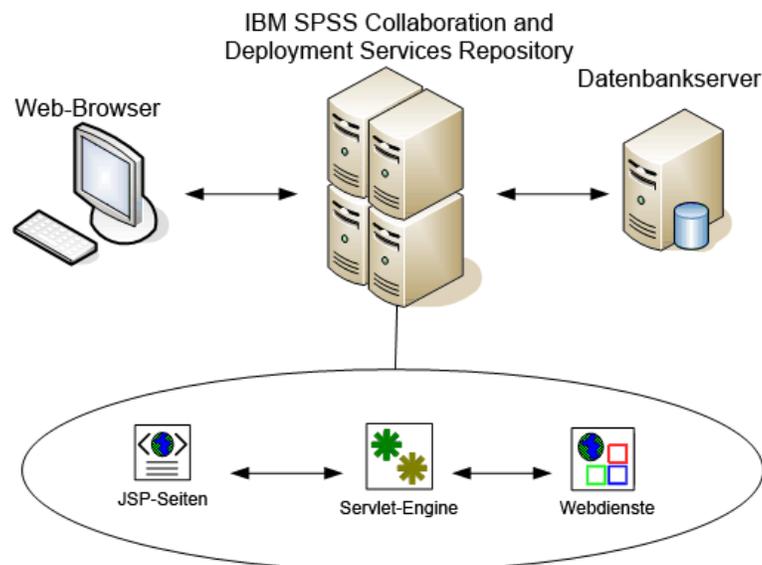
Beachten Sie, dass bei früheren Versionen von IBM SPSS Collaboration and Deployment Services eine *.tld*-Datei mit der Bezeichnung *reporting-taglib.tld* oder *pasw-taglib.tld* verwendet wurde. Alle vorhandenen JSP-Seiten, die einen dieser Namen verwenden, sollten so aktualisiert werden, dass sie auf *IBMSPSSTaglib.tld* verweisen.

Außerdem sollten Sie sich vergewissern, dass etwaige benutzerdefinierte Seiten, die mit früheren Versionen von IBM SPSS Collaboration and Deployment Services erstellt wurden, in der aktuellen Version erwartungsgemäß arbeiten. Möglicherweise sind einige Abänderungen erforderlich. So machen beispielsweise Änderungen bei der Berichtsverarbeitung Änderungen an Seiten erforderlich, die Berichte beinhalten.

JavaServer Pages-Architektur

Die [“JSP-Architektur” Abbildung](#) stellt die Architektur für die Nutzung der Tag-Bibliothek dar.. Der Anwendungsserver, auf dem das IBM® SPSS® Collaboration and Deployment Services Repository gehostet ist, enthält eine Servlet-Engine, die die Informationen in den Bibliothek-Tags in Eingaben für Webdienste konvertiert, die in IBM® SPSS® Collaboration and Deployment Services enthalten sind.

Abbildung 4-1
JSP-Architektur



Die Ausführung von Elementen über die Tag-Bibliothek gestaltet sich in der Regel folgendermaßen:

- Der JSP-Entwickler verwendet benutzerdefinierte Tags, um Anmeldeinformationen, Elemente und Elementbeziehungen in einer *.jsp*-Datei zu definieren, und speichert die Datei auf dem Anwendungsserver, der das IBM SPSS Collaboration and Deployment Services Repository hostet.

- ▶ Wenn ein Client auf die *.jsp*-Seite zugreift, wertet der Server die Tags aus und erzeugt je nach Bedarf XML-Dateninseln oder HTML-Elemente, die von den JavaScript-Komponenten des Frameworks verwendet werden, um Beziehungen zwischen Elementen und Prompts zu erkennen und zu verwalten.
- ▶ In jedem Tag-Handler werden Validierungen durchgeführt und die entsprechenden Fehlermeldungen auf der Seite angezeigt, so dass der Benutzer in jeder Phase der Seitenerstellung über Fehler in Kenntnis gesetzt wird.
- ▶ Ein Servlet unterstützt die Ausführung von Elementen und die Verarbeitung und Rückgabe der Ausgabe.
- ▶ Der entsprechende Webdienst für den Elementtyp wird aufgerufen, um das Element auszuführen und verschiedene Validierungen durchzuführen.

Unterstützte Elemente

Über die IBM® SPSS® Collaboration and Deployment Services Tag Library können verschiedenen Repository-Elemente in JSP-Seiten referenziert werden. Bei der Verarbeitung der Seite bestimmt der MIME-Typ des Elements, wie es verarbeitet wird. Folgende Elemente sind gültig:

- Berichte
- Jobs
- Scoring-Modelle
- Benutzerdefinierte Schnittstellendefinitionen

Berichte

Bei einem Bericht muss das Repository-Element einen der folgenden Dateitypen referenzieren:

- BIRT Report Designer for IBM® SPSS®-Bericht-Design (**.rptdesign*)
- IBM® ShowCase®-Bericht (**.dbq* oder **.rpt*)
- Visualisierungsdefinition (**.viz*)

Folgende Eigenschaften sollten bei der Arbeit mit Berichtselementen berücksichtigt werden:

Ausgabe. Ein Berichtselement generiert für gewöhnlich eine einzelne Ausgabe.

Visualisierungsberichte jedoch generieren neben der Visualisierung eine verweissensitive Grafik. Die Ausgabe für das Element kann, abhängig vom Berichtstyp, in unterschiedlichen Formaten erfolgen. Folgende Formate werden unterstützt:

- HyperText Markup Language (**.html*)
- Portable Document Format (**.pdf*)
- Berichtsdocument (**.rptdocument*)
- HTML Complete (**.htmlc*)
- MIME HTML (**.mht*)

- Microsoft Word-Dokument (*.doc)
- Microsoft PowerPoint (*.ppt)
- Portable Network Graphic (*.png)
- Enhanced Metafile (*.emf)
- Joint Photographic Experts Group (*.jpeg)

Prompts. Bei der Verarbeitung wird das Element Werte für alle Variablen anfordern, die im Bericht definiert sind.

Beschränkungen hinsichtlich des Anzeigeorts. Ausgaben vom Typ **.rptdocument* können nur in IFRAME angezeigt werden.

Verknüpfung von Elementen. Berichtselemente können als Quellen für nachfolgende Elemente oder als Ziele für andere Elemente verwendet werden.

Unterstützte Tags. Berichtselemente unterstützen nicht das `outputLocation`-Tag. Alle anderen Tags in der Tag-Bibliothek werden unterstützt.

Das Element kann zusätzliche Informationen enthalten, die die Darstellung der Ausgabe beeinflussen, z. B. Titel des Fensters oder Vorhandensein einer Symbolleiste.

Jobs

Im Fall eines Jobs muss das Repository-Element einen Job in IBM® SPSS® Collaboration and Deployment Services referenzieren, dessen MIME-Typ *application/x-vnd.spss-prms-job* lautet. Folgende Eigenschaften sollten bei der Arbeit mit Job-Elementen berücksichtigt werden:

Ausgabe. Ein Job-Element kann eine unbegrenzte Anzahl von Ausgaben verschiedener Typen generieren. Welche Ausgabe generiert wird, hängt von den Schritten ab, die im Job enthalten sind.

Prompts. Bei der Verarbeitung wird das Element Werte für alle Job-Parameter anfordern, die im Job definiert sind.

Beschränkungen hinsichtlich des Anzeigeorts. Die Ausgabe der einzelnen Schritte innerhalb des Jobs muss explizit definiert werden.

Verknüpfung von Elementen. Job-Elemente können als Ziele für andere Elemente verwendet werden, nicht jedoch als Quellen.

Unterstützte Tags. Job-Elemente unterstützen nicht das `actionHandler`-Tag. Alle anderen Tags in der Tag-Bibliothek werden unterstützt.

Scoring-Modelle

Bei einem Scoring-Modell muss das Repository-Element eine für das Scoring definierte Datei referenzieren. Gültige Dateitypen sind:

- Szenario (*.scn)

- IBM® SPSS® Modeler-Stream (*.str)
- Predictive Model Markup Language (PMML)
- Definition einer Echtzeit-Prognoseanwendung

Folgende Eigenschaften sollten bei der Arbeit mit Scoring-Elementen berücksichtigt werden:

Ausgabe. Ein Scoring-Element generiert eine HTML-Ausgabe.

Prompts. Bei der Verarbeitung kann das Element Werte für Parameter, eine Datendatei, eine Daten-Provider-Definition und einen Modellnamen anfordern.

Verknüpfung von Elementen. Scoring-Elemente können als Ziele für andere Elemente verwendet werden, nicht jedoch als Quellen.

Unterstützte Tags. Scoring-Elemente unterstützen nicht das `outputLocation`- und das `actionHandler`-Tag. Alle anderen Tags in der Tag-Bibliothek werden unterstützt.

Benutzerdefinierte Dialogfelder

Anmerkung: Für diese Funktionalität sind IBM® SPSS® Statistics-Adapter in der IBM® SPSS® Collaboration and Deployment Services-Umgebung erforderlich. Weitere Informationen finden Sie in der SPSS Statistics-Installationsdokumentation.

Bei einer benutzerdefinierten Weboberfläche muss das Repository-Element eine Dialogfelddefinition (*.spd) referenzieren. Folgende Eigenschaften sollten bei der Arbeit mit benutzerdefinierten Dialogfeldern berücksichtigt werden:

Ausgabe. Ein benutzerdefiniertes Dialogfeld erzeugt eines der folgenden Elemente:

- Eine einzelne Ausgabedatei (*.spw), die einem Frame oder Fenster zugewiesen werden muss.
- Eine HTML-Datei, die einem Frame bzw. Fenster oder einem DIV-Ziel zugewiesen werden kann.

Prompts. Bei der Verarbeitung wird das Element Werte für alle Prompts anfordern, die in der Dialogfelddefinition definiert sind. Das Element kann auch Daten-Sets anfordern. Jegliche in der .spd-Datei definierte Hilfe für Prompts wird jedoch nicht verwendet. Die Anwendung sollte eigene Hilferenzen enthalten.

Beschränkungen hinsichtlich des Anzeigeorts. Die Ausgabe kann in einem Frame, DIV oder einem neuen Fenster angezeigt werden.

Verknüpfung von Elementen. Dialogfeld-Elemente können als Ziele für andere Elemente verwendet werden, nicht jedoch als Quellen.

Unterstützte Tags. Dialogfeld-Elemente unterstützen nicht das `actionHandler`-Tag. Alle anderen Tags in der Tag-Bibliothek werden unterstützt.

Die zur Verwendung in einem URL beschriebenen Web Deployment-Eigenschaften, der ein benutzerdefiniertes Dialogfeld referenziert, können in der Tag-Bibliothek entweder als im `repositoryItem`-Tag verschachtelte Eigenschaften oder mit Hilfe des `sourceLinkPrompt`-Tags angegeben werden. [Für weitere Informationen siehe Thema Parameter für benutzerdefinierte Dialogfelder in Kapitel 3 auf S. 19.](#)

Die Eigenschaften `dataset.uri` und `dataset.table` sollten immer definiert werden, wobei letztere nur für Daten-Provider-Definitionen gilt. Die Eigenschaften `javascript.url`, `javascript.name`, `stylesheet.url` und `stylesheet.name` hingegen werden alle ignoriert. Werte für diese Eigenschaften sollten innerhalb der JSP selbst definiert werden.

Aufbauen einer Anwendung

Jede JSP-Seite in einer benutzerdefinierten Anwendung muss einige Standard-Aufbauregeln definieren, damit die Tag-Bibliothek verwendet und fehlerfrei referenziert werden kann. Die erste, die Aufbauregel `page`, legt Eigenschaften für die gesamte Seite fest. Zu diesen Eigenschaften gehören:

- Das `language`-Attribut, das die die Skriptsprache definiert, die von der Seite verwendet wird
- Das `contentType`-Attribut, das den MIME-Typ und den Zeichensatz angibt, der für Reaktionen auf Clients verwendet wird
- Das `session`-Attribut, das angibt, ob die Tag-Bibliothek Informationen über die Sitzung speichert

Die zweite Aufbauregel `taglib` legt fest, welche Tags von der JSP-Seite verwendet werden. Folgende Eigenschaften sind für diese Aufbauregel definiert:

- Das `uri`-Attribut, das den richtigen Pfad zu `IBMSPSSTaglib.tld` angibt
- Das `prefix`-Attribut, das einen Bereich für die Tags definiert

Beachten Sie, dass bei früheren Versionen von IBM® SPSS® Collaboration and Deployment Services eine `.tld`-Datei mit der Bezeichnung `reporting-taglib.tld` oder `pasw-taglib.tld` verwendet wurde. Alle vorhandenen JSP-Seiten, die einen dieser Namen verwenden, sollten so aktualisiert werden, dass sie auf `IBMSPSSTaglib.tld` verweisen.

Im folgenden Beispiel wird die `page`-Aufbauregel verwendet, um den Inhaltstyp als `text/html` mit dem Zeichensatz UTF-8, der Skriptsprache Java und dem `session`-Objekt "true" zu definieren. Die `taglib`-Aufbauregel legt den Speicherort der Berichtsdatei `.tld` und ein Präfix `r` für alle darin definierten Tags fest.

```
<%@ page contentType="text/html;charset=utf-8"
  language="java" session="true" %>

<%@ taglib uri="/WEB-INF/tlds/IBMSPSSTaglib.tld" prefix="r" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  http://www.w3.org/TR/html4/loose.dtd">

<html>
<!-- Rest of HTML / JSP goes here -->
</html>
```

Um Ihre Anwendung in die Produktion zu geben, sollten Sie die Erstellung einer Archivdatei für die Webanwendung (.war) mit den .jsp-Dateien planen und sie als separate Webanwendung auf Ihrem Anwendungsserver bereitstellen. Dies ist die bevorzugte Methode.

Die Struktur des erweiterten Anwendungsarchivs der Beispiel-Tag-Bibliothek "Reporting" (IBMSPSSTaglib.tld.war) in der Standardinstallation von IBM SPSS Collaboration and Deployment Services lautet beispielsweise folgendermaßen:

```

IBMSPSSTaglib
• index.html
• setup.html
•••js
• <JavaScript-Dateien>•
•••jsp
• <Java Server Page-Dateien>
•••META-INF
• MANIFEST.MF
••••WEB-INF
• • web.xml
• • weblogic.xml
• •••lib
• • <Java-Archivdateien>
• •••tlds
• IBMSPSSTaglib.tld
• reporting-taglib.tld
••••xsl
  <Extensible Stylesheet Language-Dateien>

```

Beachten Sie, dass die TLD- (Tag Library Descriptions) Datei und die TLD-Bibliotheken (.jar-Dateien) in der bereitgestellten .war-Datei enthalten sind. Die TLD-Datei wird auch in der Deskriptor-Datei der Anwendung (web.xml) referenziert:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
  <filter>
    <filter-name>Common Authentication Filter</filter-name>
    <filter-class>com.spss.report.taglib.filter.AuthFilter</filter-class>
  <init-param>
    <param-name>PARAMETER_ENCODING</param-name>
    <param-value>UTF-8</param-value>
    <description>Parameter Encoding</description>
  </init-param>
  <init-param>
    <param-name>SSO_ADAPTER_CLASS</param-name>
    <param-value>com.spss.er.sso.authenticator.SessionAuthenticatorImpl</param-value>
    <description>SSO Authenticator Impl class</description>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>Common Authentication Filter</filter-name>
  <url-pattern>/reportingTaglib/*</url-pattern>

```

```

</filter-mapping>
<filter-mapping>
  <filter-name>Common Authentication Filter</filter-name>
  <url-pattern>/tagLib/*</url-pattern>
</filter-mapping>
<servlet>
  <servlet-name>ReportingTaglibServlet</servlet-name>
  <display-name>
    Servlet verantwortlich für Erfüllung aller Anforderungen von
    Berichts-Tag-Bibliotheken
  </display-name>
  <servlet-class>
    com.spss.report.taglib.servlet.ReportingTaglibServlet
  </servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>ReportingTaglibServlet</servlet-name>
  <url-pattern>/reportingTagLib/*</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>ReportingTaglibServlet</servlet-name>
  <url-pattern>/tagLib/*</url-pattern>
</servlet-mapping>

<!-- Start : Taglib Node -->

<taglib>
  <taglib-uri>/reporting-taglib.tld</taglib-uri>
  <taglib-location>/WEB-INF/tlds/IBMSPSSTaglib.tld</taglib-location>
</taglib>

<!-- Ende : Taglib-Knoten -->

<!-- Start : Security-Constraint-Knoten -->
<!-- Ende : Security-Constraint-Knoten -->
</web-app>

```

Der Deskriptor der Anwendung gibt ebenfalls an, dass das Servlet *ReportingTaglibServlet* den URL-Mustern */taglib* und */reportingTagLib* zugewiesen wird und beide URLs das Servlet aufrufen können. Die Java-Klasse des Servlet lautet *com.spss.report.taglib.servlet.ReportingTaglibServlet*. Die optionale Einzelanmeldung wird durch einen Servlet-Filter *Common Authentication Filter* aktiviert, der die Klasse *com.spss.report.taglib.filter.AuthFilter* verwendet und dem Servlet über einen URL zugewiesen ist. Der Filter wird über Parameter zur Verschlüsselung und für die Adapterklasse zur Einzelanmeldung initialisiert.

Weitere Informationen zu *.war*-Dateien finden Sie in Online-Ressourcen wie http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/WebComponents3.html. Weitere Informationen und Anforderungen finden Sie zudem in der Dokumentation Ihres Anwendungsservers.

Wichtig: Für *.war*-Anwendungsdateien, die nicht von IBM SPSS Collaboration and Deployment Services-Installationskripts oder IBM® SPSS® Collaboration and Deployment Services Package Manager bereitgestellt werden, wie Tag-Bibliothek- oder benutzerdefinierte Anwendungen, muss evtl. die Klassenlader-Reihenfolge geändert werden. Beispielsweisemuss sie für Bericht- und Scoring-Tag-Bibliothek-Anwendungen auf WebSphere auf *Klassen mit Anwendungsklassenlader zuerst geladen* und die *.war*-Klassenladerrichtlinie auf *Einzelner Klassenlader für Anwendung* gesetzt sein.

Implementierungsdetails

Benutzer, die benutzerdefinierte *.jsp*-Seiten erstellen, sollten sich über folgende Punkte im Klaren sein:

- Jedes Mal, wenn der Server anhält, gehen *.jsp*-Dateien, die direkt im Verzeichnis *tmp/deploy* gespeichert sind, verloren. Um Ihre *.jsp*-Dateien beizubehalten, speichern Sie Sicherungskopien auf einer lokalen Festplatte und kopieren Sie sie nach jedem Neustart wieder auf den Server. Produktionsanwendungen sollten in *.war*-Dateien verpackt werden. [Für weitere Informationen siehe Thema Aufbauen einer Anwendung auf S. 35.](#)
- Internet Explorer 6.0 und Mozilla Firefox™ 1.5 werden unterstützt. Für Firefox gelten folgende Einschränkungen:
 - Für Berichte mit Bildern bzw. Diagrammen kombiniert das *.mht*-Ausgabeformat alle Bilder und Diagramme und HTML-Elemente in einer einzelnen Datei, die nur zu Internet Explorer kompatibel ist.
 - IBM® ShowCase® Report Writer-Berichte (*.rpt*) verwenden eine spezielle Technologie für Diagramme. Diese Technologie ist nur mit Internet Explorer kompatibel.
- Der Server validiert alle Tags soweit wie möglich vor und schreibt eventuell festgestellte Fehlermeldungen an die HTTP-Antwort. Das unterstützt den JSP-Entwickler bei der Behebung von Problemen mit der Erstellung einer Seite. Es werden beispielsweise folgende Elemente validiert: Überprüfung, ob alle Anmeldeinformationen definiert sind; Überprüfung der Anmeldeinformationen; Überprüfung, ob Berichtparameter vorhanden sind; Überprüfung, ob Spaltennamen für ein Berichtobjekt vorhanden sind; Überprüfung, ob die Verknüpfung zwischen Elementen logisch schlüssig sind.
- Für die Tags ist ein Repository-Server erforderlich.
- Die Tag-Bibliothek unterstützt Verknüpfungen zwischen Prompts und Elementen, zwischen dem Klicken auf eine Zeile und Zielelementen, zwischen Berichten und Drill-Down-Berichten im selben Frame und zwischen Prompt-/Tabellenzeilen und Zielelementen, die in separaten Fenstern geöffnet sind.
- Verknüpfungsverhalten wird dem Benutzer nicht angezeigt und über `sourceLinkRepositoryItem`- und/oder `sourceLinkPrompt`-JSP-Tags definiert. Der Benutzer muss die Technologie hinter der JSP-Tag-Verwendung nicht verstehen.

- Alle Zielelemente müssen mit Parametern vordefiniert werden, um die weitergegebenen Parameter zu erhalten.
- Für benutzerdefinierte Dialogfelder definiert das Standard-CSS Standardstile, die bei Verwendung des `repositoryItem`-Tags automatisch mit einbezogen werden. Um diese Stile zu überschreiben, müssen Sie nach dem `repositoryItem`-Tag ein benutzerdefiniertes Stylesheet einfügen. Beispiel:

```
<r:repositoryItem name="sample" inputURI="spsscr:///myDialog.spd"
  ...more here...
</r:repositoryItem>
```

```
<link rel="stylesheet" type="text/css" href="MyStyles.css">
```

Öffentliche JavaScript-API

Das Framework stellt JavaScript-Funktionen zur Verarbeitung von Repository-Elementen, für den Abruf von mit Lesezeichen versehenen Berichtswerten und von Weiterleitungs-Prompt-Werten bereit.

Die Funktion "runRepositoryItem"

Die öffentliche JavaScript-Funktion `runRepositoryItem` wird vom Framework für das Ausführen eines Elements bereitgestellt. Mit ihr kann der Entwickler ein Element ausführen, indem er dieses JavaScript mit einem Ereignis-Handler verknüpft, und ein Element über Prompts aktivieren. Wenn ein Element jedoch direkt über diese Funktion ausgeführt wird, wird die normale Prompt-Validierung umgangen. Die Anwendung ist dafür zuständig, Parameter zu validieren, bevor die Funktion aufgerufen wird.

Die Funktion nimmt folgende drei Argumente an.

- Eine Zeichenfolge für den Namen eines Elements, das ausgeführt werden soll. Der Name muss über das `name`-Attribut des `repositoryItem`-Tags definiert worden sein.
- Ein Array von Datenwerten, die bei der Ausführung des Elements als Parameterwerte verwendet werden sollen. Das Array besitzt die folgende Struktur:

```
var thisVar = new Object();
thisVar.value = "param_value";
thisVar.columnName = "param_name";
var linkedData = new Array(thisVar);
```

- Einen optionalen Parameter, der den Zielspeicherort für die Elementausgabe angibt. Er folgt denselben Regeln wie das `location`-Attribut des `repositoryItem`-Tags. Dabei kann es sich um die ID eines DIV-Elements, den Namen eines IFRAME- oder FRAME-, **NEW*- oder **windowName*-Elements handeln. Wenn dieser Parameter nicht angegeben wird, wird der Standardspeicherort des `repositoryItem` verwendet.

Das Datenwert-Array kann folgendermaßen definiert werden:

- Als NULL (oder weggelassen). In diesem Fall werden die erforderlichen Werte über `sourceLinkPrompt`-Tags abgerufen, die für das Element definiert sind.

- Als Verknüpfungsdaten aus `actionHandler`. Die `actionHandler`-Tags definieren Funktionen, die aufgerufen werden sollen, und die Parameterwerte, die im Rahmen des Ereignisses erfasst werden. Diese Parameter können direkt an die `runRepositoryItem`-API weitergegeben werden.
- Als benutzerdefiniert. Das JavaScript, das die `runRepositoryItem`-API aufruft, kann die Werte des obigen Arrays bei Bedarf definieren. `columnName` ist der Name der Spalte, die im `sourceLinkVariable`-Tag definiert wird. Der Wert ist der Wert, der an den Parameter weitergegeben wird.

Die Funktion “getBookmarkedValues”

Die Funktion `getBookmarkedValues` ruft die Werte von Zellen ab, die in einem BIRT Report Designer for IBM® SPSS®-Bericht mit einem Lesezeichen versehen wurden. Das Attribut “id” der Zelle sollte auf den Lesezeichenwert eingestellt werden. Diese Funktion kann zur Verknüpfung komplexer BIRT Report Designer for IBM SPSS-Berichte mit Kreuztabellen und verschachtelten Tabellen verwendet werden.

Die Funktion `getBookmarkedValues` nimmt die folgenden Argumente an:

- Einen übergeordneten Knoten im DOM des Berichts, den die Funktion benötigt, um die Zellenwerte abrufen zu können, die den im Array des Lesezeichens angegebenen Elementen entsprechen.
- Ein Array mit Lesezeichen, die im Bericht definiert sind, dessen Werte benötigt werden. Beispiel: [“bookid1”, “bookid2”].

Die Funktion gibt ein Array von Datenwerten aus, die bei der Ausführung des Berichts als Parameterwerte verwendet werden sollen. Das Array besitzt die folgende Struktur:

```
var thisVar = new Object();
  thisVar.value = "cell_value";
  thisVar.columnName = "bookmark";
```

`columnName` ist der Name des Lesezeichens. `value` ist der Wert der angegebenen Zelle, die mit einem Lesezeichen versehen wird.

Die Funktion “retrievePromptValues”

Die Funktion `retrievePromptValues` sollte aufgerufen werden, wenn Parameter mit benutzerdefinierten Steuerelementen verwendet werden. Sie unterstützt sowohl Weiterleitungs- als auch Nicht-Weiterleitungs-Prompts. Rufen Sie diese Funktion im Textkörper des Handlers `onLoad` auf, um die Anfangswerte des Prompts zu laden (oder des übergeordneten Prompts bei Weiterleitungs-Prompts). Rufen Sie diese Funktion im Handler `onChange` des für die Definition des Weiterleitungsparameters verwendeten Steuerelements auf. Die Funktion ruft vom Server die Prompt-Werte zum Auffüllen der Parametersteuerelemente mit aktualisierten Werten abhängig vom ausgewählten übergeordneten Parameterwert ab.

Die Funktion `retrievePromptValues` nimmt die folgenden Argumente an:

- Eine Zeichenfolge mit dem Namen des Berichts, der die Definitionen der Weiterleitungsparameter enthält. Der Name muss als das Attribut `Name` eines `repositoryItem`-Tags definiert worden sein.
- Eine Zeichenfolge, die dem Namen des Parameters im Bericht entspricht. Für Weiterleitungsparameter ist diese Zeichenfolge der Name der Weiterleitungsparametergruppe. Die Weiterleitungsgruppe muss im Bericht vorhanden sein.
- Eine benutzerdefinierte Funktion, die ein Array von Werten akzeptiert und Text für die neuen Optionen anzeigt. Das Array kann "Null" sein. In diesem Fall sollte die Funktion das Steuerelement löschen. Diese Funktion wird über `retrieveCascadingPromptValues` aufgerufen, um die Parametersteuerelemente mit neuen Werten zu bevölkern.

```
function callback(options) {
  // Logik zum Löschen des Steuerelements
  // Logik zum Hinzufügen von Werten und Anzeigetext zum Steuerelement
  for(var i = 0; i < options.length; i++) {
    control.value = options[i].value;
    Display Text for control = options[i].displayText;
  }
}
```

- Ein Array der ausgewählten vorangehenden Werte in der Weiterleitungsgruppe. Dieses Array wird nur für Weiterleitungsparameter benötigt und sollte für einen Nicht-Weiterleitungsparameter ausgelassen werden. Die Parameter müssen in sequenzieller Reihenfolge sein. Um die Liste des übergeordneten Weiterleitungsparameters abzurufen, geben Sie die vorangehenden Werte an:

```
var precedingvals= new Array();
```

Das Array mit den vorangehenden Werten besitzt die folgende Struktur: Beispielsweise zum Abrufen einer Liste der Städte in MN:

```
precedingvals= new Array();
precedingvals[0]= "USA";
precedingVals[1]="MN";
```

IBM SPSS Collaboration and Deployment Services Tag Library-Tag-Referenz

Die verschiedenen Tags in der IBM® SPSS® Collaboration and Deployment Services Tag Library sind voneinander abhängig und zu Validierungszwecken muss sichergestellt sein, dass die Referenzen korrekt eingehalten werden. Außerdem müssen die Tags in der korrekten Reihenfolge definiert werden. In den folgenden Abschnitten wird jedes verfügbare Tag ausführlich beschrieben.

Diese Tag-Bibliothek basiert auf JSP 1,2.

Das Tag "credential"

Das `credential`-Tag definiert sowohl die Anmeldeinformationen für eine Datenquelle, als auch für ein Repository. Das `credential`-Tag wird für alle auf der Seite definierten Elemente und/oder Eingabeaufforderungen nach Namen referenziert. Es sollte vor allen anderen Tags definiert werden, die das `credential`-Tag eventuell referenzieren. Bei normaler Verwendung wäre es das erste in der JSP referenzierte Tag.

Das `credential`-Tag kann `properties`-Elemente enthalten. Im Fall von J.-D.-Edwards- (JDE) Datenquellen sieht das `credential`-Tag beispielsweise folgendermaßen aus:

```
<credential>
  <properties>
    <property name="JDE_LIBRARY_LIST_SELECTED" value="liblist_name"/>
  </properties>
</credential>
```

Tag-Verschachtelung

Keines

Erwartete Ausgabe

Keine. Dieses Tag enthält Authentifizierungsinformationen. Das Tag liefert keine Ausgabe, sondern nimmt die Anmeldeinformationen mit Hilfe des `name`-Attributs als Schlüssel zur späteren Verwendung mit einem `report`- oder `prompt`-Tag in den Cache auf.

Tabelle 4-1

Attribute für das `credential`-Tag

Name	Erforderlich	Beschreibung
Name	true	<p>Entweder ein interner Name für die Anmeldeinformationen für das IBM® SPSS® Collaboration and Deployment Services Repository oder der Name einer in einem Repository-Objekt verwendeten Datenquelle. Wird verwendet, um Elemente und Eingabeaufforderungen mit diesen Anmeldeinformationen zu verknüpfen und für alle erforderlichen Anmeldungen von referenzierten Elementen bei Datenquellen.</p> <ul style="list-style-type: none"> ■ Für Anmeldeinformationen für ein Repository muss dieser Name mit dem im <code>repositoryCredentialName</code>-Attribut des <code>repositoryItem</code>-Tags angegebenen Namen übereinstimmen. ■ Für Datenbank-Anmeldeinformationen muss der Name dem Namen der Datenquelle entsprechen, da

Name	Erforderlich	Beschreibung
		<p>er vom Element referenziert wird, das diese Datenquelle verwendet.</p> <p>Dieser Name wird verwendet, um die Anmeldeinformationen in einer Sitzungsvariablen zu speichern. Jedes credential-Tag muss einen eindeutigen Namen besitzen.</p>
useSSO	false	<p>Gibt an, ob Einzelanmeldungs-Anmeldeinformationen für Kerberos verwendet werden sollen. Falls dieses Attribut auf "true" eingestellt ist, müssen die Attribute "Benutzername", "Passwort" und "Provider" nicht angegeben werden.</p> <p>Wenn die Einzelanmeldung verwendet wird, muss der Authentifizierungsfilter in der Datei web.xml konfiguriert werden. Für weitere Informationen siehe Thema Aufbauen einer Anwendung auf S. 35.</p>
credentialDefinitionName	false	<p>Der Name von Anmeldeinformationen, die im Repository als Ressource definiert sind. Falls dieser Wert angegeben ist, müssen die Attribute <code>username</code>, <code>password</code> und <code>provider</code> nicht angegeben werden, da die Anmeldeinformations-Ressource diese Informationen bereits enthält.</p>
provider	false	<p>Für Anmeldeinformationen für ein Repository ist dies der optionale Name eines Sicherheitsanbieters. Gültige Werte sind:</p> <ul style="list-style-type: none"> ■ <i>Nativ</i> für den integrierten Anbieter. ■ <i>AD_<name>/<domain></i> für Active Directory, wobei <i><name></i> dem Namen des Sicherheits-Providers im System und <i><domain></i> dem DNS-Namespace entspricht ■ <i>ADL_<name>/<domain></i> für Active Directory mit lokaler Überschreibung, wobei <i><name></i> dem Namen des Sicherheits-Providers im System und <i><domain></i> dem DNS-Namespace entspricht

Name	Erforderlich	Beschreibung
		<ul style="list-style-type: none"> ■ <i>iSeries</i> für IBM i ■ <i>ldap_<name></i> für OpenLDAP, wobei <i><name></i> dem Namen des Sicherheits-Providers im System entspricht <p>Falls kein Name angegeben ist, wird der integrierte Repository-Sicherheitsanbieter verwendet. Dieses Attribut wird für Anmeldeinformationen für Datenbanken ignoriert.</p>
username	false	Der Benutzername, der für die Authentifizierung verwendet wird.
password	false	Das Passwort für den angegebenen Benutzernamen. Das Passwort wird intern von der Tag-Bibliothek verwendet. Es wird NICHT in das JSP-Ergebnis geschrieben.

Verwendungsbeispiel

In folgendem Beispiel sind drei credential-Tags angegeben: Das erste dient zum Zugriff auf das IBM SPSS Collaboration and Deployment Services Repository über einen vorgegebenen Benutzernamen und ein vorgegebenes Passwort. Der Wert *Native* für *provider* zeigt an, dass Benutzername und Passwort für die Validierung im nativen (bzw. lokalen) Sicherheits-Provider definiert sind. Das zweite credential-Tag ermöglicht die Einzelanmeldung für das IBM SPSS Collaboration and Deployment Services Repository unter Verwendung der zuvor authentifizierten Anmeldeinformationen des Benutzers. Das dritte credential-Tag gilt für eine Datenquelle mit der Bezeichnung *Northwind*.

```
<r:credential name="repositoryCredential" provider="Native"
  username='admin' password='password' />
<r:credential name="repositorySSO" useSSO="true" />
<r:credential name="Northwind" username='sa' password='sa' />
```

Das repositoryItem-Tag

Das *repositoryItem*-Tag ist das wichtigste Tag zur Definition von Repository-Element-Definitionen, die von der Anwendung verwendet werden. Das *repositoryItem*-Tag kann Berichte (IBM® ShowCase®-Berichte, BIRT Report Designer for IBM® SPSS®-Berichte oder Visualisierungsberichte), Jobs, Scoring-Elemente oder SPD-Dateien referenzieren. Die Repository-Elemente können direkt ausgeführt, zur Bereitstellung von Eingabeaufforderungen verwendet oder programmatisch ausgeführt werden.

sourceLinkPrompt- und *sourceLinkRepositoryItem*-Tags sollten wie folgt im *repositoryItem*-Tag verschachtelt sein.

- Verwenden Sie ein verschachteltes `sourceLinkRepositoryItem`-Tag, falls dieses Element ausgeführt wird, wenn der Benutzer auf ein anderes Element klickt.
- Verwenden Sie das `sourceLinkPrompt`-Tag, wenn die Parameterwerte von auf der Seite oder direkt im Element definierten Eingabeaufforderungen stammen.

Optional können Sie zusätzliche Eigenschaften angeben, die für einen bestimmten Repository-Elementtyp spezifisch sind. Damit sie im Firefox-Browser verwendet werden können, müssen die Eigenschaftsnamen in Kleinbuchstaben stehen. Diese Eigenschaftswerte werden an den URL übergeben, damit das Repository-Element ausgeführt wird. Die Eigenschaften werden als verschachtelter XML-Block angegeben.

Tag-Verschachtelung

Dieses Tag kann ein `sourceLinkRepositoryItem`-Tag und/oder mehrere `sourceLinkPrompt`- und `outputLocation`-Tags enthalten.

Tabelle 4-2
Attribute für das `repositoryItem`-Tag

Name	Erforderlich	Beschreibung
Name	true	Definiert einen eindeutigen Namen für das Element. Der Name kann danach von anderen Tags oder über die <code>runRepositoryItem()</code> JavaScript API referenziert werden.
inputURI	true	Die für das Rendering der Berichtsausgabe zu verwendende Elementdefinition. Dieser Wert muss einen URI angeben, der zur Suche nach der Elementdefinition verwendet werden kann. Die folgenden URI-Schemata werden unterstützt: <ul style="list-style-type: none"> ■ file: Referenziert eine spezielle Datei auf dem Anwendungsserver und/oder an einem Netzwerkspeicherort. ■ spsscr: Referenziert einen Knoten im IBM® SPSS® Collaboration and Deployment Services Repository. Dieses Schema ermöglicht die Referenzierung von Dateien durch Bezeichner oder hierarchische Pfade im Repository. Es können spezielle Versionskennzeichnungen angegeben werden. Falls keine Version oder Bezeichnung angegeben

Name	Erforderlich	Beschreibung
		<p>wird, wird die letzte Version verwendet.</p> <ul style="list-style-type: none"> ■ Scoring: Referenziert eine Modellkonfiguration aus dem Repository. Scoring-Konfigurationen werden nach dem Namen aus den Tag-Bibliotheken referenziert. Falls eine Scoring-Konfiguration umbenannt wird, muss auch die Tag-Bibliothek-Referenz geändert werden.
activate	true	<p>Gibt an, wann das Element aktiviert wird. Zu den Optionen gehören:</p> <ul style="list-style-type: none"> ■ ONDEMAND: Führt das Element aus, wenn er durch Klicken auf eine Zeile in einem Quellbericht aktiviert wird. ■ ONLOAD: Führt das Element aus, wenn die Seite zum ersten Mal geladen wird. ■ NONE: Das Element wird nicht automatisch ausgeführt. In diesem Fall wird das Element zur Bereitstellung von Eingabeaufforderungen oder von Werten für Eingabeaufforderungen verwendet. <p>Unabhängig von der Aktivierungseinstellung kann jeder Bericht durch Verwendung der öffentlichen JavaScript <code>runRepositoryItem()</code>-API programmatisch ausgeführt werden.</p>
Speicherort	false	<p>Das Ziel für die Ausgabe des ausgeführten Elements. Die Verwendung variiert etwas abhängig vom Zieltyp.</p> <ul style="list-style-type: none"> ■ Bei DIV-Zielen sollte das <code>location</code>-Attribut die ID des DIV-tags angeben, in dem die Ausgabe platziert werden soll. ■ Bei IFRAME-Zielen muss das <code>location</code>-Attribut den Namen des Frames angeben.

Name	Erforderlich	Beschreibung
		<ul style="list-style-type: none"> ■ Um die Ausgabe in einem neuen Fenster zu öffnen, geben Sie als Speicherort *NEU an. ■ Um die Ausgabe an ein benanntes Fenster umzuleiten, verwenden Sie ein Sternchen (*) gefolgt vom Fensternamen. Beispielsweise öffnet *MYWINDOW ein neues Fenster mit dem Namen <i>MYWINDOW</i> und verwendet dieses Fenster erneut bei jeder Aktivierung der Verknüpfung.
repositoryCredentialName	true	Der Name der zu verwendenden Anmeldeinformationen, wenn vom Repository aus auf das Element zugegriffen wird und wenn das Element über den Reporting-Dienst ausgeführt wird. Die Anmeldeinformationen sollten zuvor über das credential-Tag definiert worden sein.
outputType	false	<p>Der zu erstellende Ausgabotyp. Die unterstützten Ausgabotypen variieren je nach Elementtyp. Normalerweise erfolgt die Ausgabe im HTML- oder PNG-Format. Es sind jedoch auch folgende Optionen möglich:</p> <p>BIRT Report Designer for IBM SPSS Berichte: HTML, PDF, RPTDocument, PowerPoint, Word Document, HTMLC</p> <p>Visualisierungsberichte: PNG, EMF, JPEG, HTML.</p> <p>IBM® ShowCase® Query: HTML, PDF, CSV, WK4, Excel 97/2003, xlsx</p> <p>IBM® ShowCase® Report Writer: HTML, PDF, MHT</p> <p>Ohne Angabe erfolgt die Ausgabe standardmäßig im HTML-Format (oder im PNG-Format für Visualisierungsberichte). Wenn ein BIRT Report Designer for IBM SPSS- oder ShowCase-Bericht</p>

Name	Erforderlich	Beschreibung
		<p>als Verknüpfungsquelle verwendet wird, wird das <code>outputType</code>-Attribut ignoriert und eine Ausgabe im HTML-Format erstellt, da die anderen Ausgabetypen keine Verknüpfungen unterstützen.</p> <p>Um Berichte mit dem Viewer anzeigen zu können, geben Sie den Typ <i>RPTDocument</i> an. Für diesen Typ muss das Ziel ein IFRAME-Ziel sein.</p>
showTitle	false	<p>Gibt an, ob die Titelleiste des BIRT-Viewers angezeigt werden soll. Geben Sie als Wert entweder "true" oder "false" an. Diese Einstellung gilt nur dann, wenn für das <code>outputType</code>-Attribut das <i>RPTDocument</i>-Format angegeben ist, das im Bericht-Viewer angezeigt wird. Die Standardeinstellung lautet "true".</p>
title	false	<p>Gibt den Titel für den BIRT-Viewer an. Diese Einstellung gilt nur dann, wenn für das <code>outputType</code>-Attribut das <i>RPTDocument</i>-Format angegeben ist, das im Bericht-Viewer angezeigt wird. Falls kein Titel angegeben ist, wird der Standardtitel angezeigt.</p>
showToolBar	false	<p>Gibt an, ob die Symbolleiste des BIRT-Viewers angezeigt werden soll. Geben Sie als Wert entweder "true" oder "false" an. Diese Einstellung gilt nur dann, wenn für das <code>outputType</code>-Attribut das <i>RPTDocument</i>-Format angegeben ist, das im Bericht-Viewer angezeigt wird. Die Standardeinstellung lautet "true".</p>
showNavigationBar	false	<p>Gibt an, ob die Navigationsleiste des BIRT-Viewers angezeigt werden soll. Geben Sie als Wert entweder "true" oder "false" an. Diese Einstellung gilt nur dann, wenn für das <code>outputType</code>-Attribut das <i>RPTDocument</i>-Format angegeben ist, das im Bericht-Viewer angezeigt wird. Die Standardeinstellung lautet "true".</p>

Name	Erforderlich	Beschreibung
Breite	false	Die Breite des Ausgabebilds. Die Breite muss größer als 0 sein und zusammen mit der Höhe angegeben werden. Falls keine Breite angegeben ist, werden die Standardbreite und -höhe verwendet.
Größe	false	Die zu verwendende Höhe, falls es sich bei der Ausgabe um ein Bild handelt. Ohne Angabe einer Breite hat diese Einstellung keinerlei Auswirkungen. Der Wert muss größer als 0 sein.

Verwendungsbeispiel

Das folgende Beispiel definiert ein Element mit dem Namen *AllCountries* für einen im IBM SPSS Collaboration and Deployment Services Repository gespeicherten BIRT Report Designer for IBM SPSS-Bericht.

```
<r:repositoryItem name="AllCountries"
  inputURI="spsscr:///SampleReports/BIRT/CountrySales.rptdesign"
  repositoryCredentialName="repositoryCredential"
  outputType="HTML" width="400" height="300"
  activate="ONLOAD" location="ReportDIV">
</r:repositoryItem>
```

Um Parameterwerte für ein Element anzufordern, geben Sie ein *sourceLinkPrompt*-Tag mit an. Im folgenden Beispiel wird mit Hilfe der JavaScript-Funktion *getValue* ein Wert für den Parameter *ShipCountry* abgerufen.

```
<r:repositoryItem name="CountrySales"
  inputURI="spsscr:///SampleReports/BIRT/CountrySalesByCity.rptdesign"
  repositoryCredentialName="repositoryCredential"
  outputType="HTML" activate="ONDEMAND" location="ReportDIV">
  <r:sourceLinkPrompt targetNameParameter="ShipCountry"
    getValueJSFunction="getValue('IDFilter')"/>
</r:repositoryItem>
```

Um einen zweites Element als Reaktion auf eine Benutzeraktion auszuführen, geben Sie ein *sourceLinkRepositoryItem*-Tag mit an. Im folgenden Beispiel wird der Visualisierungsbericht *CityDetails* als Reaktion auf eine Aktion im Quellbericht *AllCountries* ausgeführt.

```
<r:repositoryItem name="CityDetails"
  inputURI="spsscr:///SampleReports/Vis/CitiesBarChart.viz"
  repositoryCredentialName="repositoryCredential"
  outputType="png" width="400" height="300"
  activate="ONDEMAND" location="SecondReportDIV">
  <r:sourceLinkRepositoryItem sourceReportName="AllCountries">
    <r:sourceLinkVariable columnName="ShipCountry" targetNameParameter="ShipCountry"/>
  </r:sourceLinkRepositoryItem>
</r:repositoryItem>
```

Das Tag "repositoryItemPrompt"

Das `repositoryItemPrompt`-Tag generiert die HTML für eine Prompt-Variable, die im referenzierten Element definiert ist. Das Element, das die Eingabeaufforderung referenziert, muss über das `repositoryItem`-Tag definiert werden, bevor dieses Tag verwendet werden kann. Verwenden Sie dieses Tag, wenn in Ihrer Anwendung Eingabeaufforderungen wie die in IBM® SPSS® Collaboration and Deployment Services Deployment Portal verwendet werden sollen.

Dieses Tag generiert die HTML-Eingabeaufforderung an dem Ort, an dem das Tag verwendet wird. Das Tag muss einem bestimmten Parameter eines Elements zugeordnet sein, um von Nutzen zu sein. Die Zuordnung zu Parametern geschieht über das `sourceLinkPrompt`-Tag, wobei die `promptID` von `sourceLinkPrompt` mit der `promptID` dieses Tags übereinstimmen muss.

Tag-Verschachtelung

Keines

Erwartete Ausgabe

Ein HTML-Element, das es dem Benutzer ermöglicht, je nach `promptType`, der als `parameterName` ausgewählt wurde, persönliche Werte auszuwählen und/oder einzugeben. Das Tag `repositoryItemPrompt` unterstützt alle Parameter, die in Deployment Portal unterstützt werden. Daher werden alle Prompt-Typen unterstützt und das entsprechende HTML-Element wird erzeugt.

Tabelle 4-3
Attribute für das `repositoryItemPrompt`-Tag

Name	Erforderlich	Beschreibung
<code>promptId</code>	false	Eine eindeutige ID, die über das Attribut <code>promptId</code> des <code>sourceLinkPrompt</code> -Tags referenziert werden kann.
<code>repositoryItemName</code>	true	Ein Verweis auf den Namen des Elements wie im Attribut <code>name</code> des <code>repositoryItem</code> -Tags definiert.
<code>parameterName</code>	false	Name der Prompt-Variable wie im Element definiert.

Verwendungsbeispiel

In folgendem Beispiel wird die Eingabe eines Werts für den Parameter `EmployeeID` im Bericht `Employees` angefordert.

```
<repositoryItem name="Employees"
  inputURI="file:///d:/yourDS/ReportTaglib/Employees.dbq"
  repositoryCredentialName="localhost" activate="NONE" />

<repositoryItemPrompt promptId="EmployeeIdPrompt"
  repositoryItemName="Employees" parameterName="EmployeeID" />
```

Das Tag "report"

Dieses Tag wurde verworfen. Verwenden Sie stattdessen das `repositoryItem`-Tag

Das Tag "reportPrompt"

Dieses Tag wurde verworfen. Verwenden Sie stattdessen das `repositoryItemPrompt`-Tag

Das Tag "outputLocation"

Dieses Tag verknüpft die generierte Ausgabe, die im Repository vorhanden ist, mit dem Ort auf der Seite, an dem die Ausgabe angezeigt wird. Bei Ausführung des Elements wird die Ausgabe aus dem Repository abgerufen und am angegebenen Zielort auf der Seite angezeigt.

Dieses Tag muss immer in einem `repositoryItem`-Tag verschachtelt sein.

Tag-Verschachtelung

Keines

Tabelle 4-4

Attribute für das `outputLocation`-Tag

Name	Erforderlich	Beschreibung
outputId	false	Dies ist der Pfad der im Repository vorhandenen Ausgabe. Für benutzerdefinierte Dialogfelder sollte dieses Attribut ausgelassen werden. Die Ausgabe aus der Ausführung der Syntax wird automatisch erkannt.
Speicherort	true	<p>Dieses Attribut gibt an, wo auf der Seite die Ausgabe eingefügt werden soll.</p> <ul style="list-style-type: none"> ■ Bei DIV-Zielen sollte das location-Attribut die ID des DIV-tags angeben, in dem die Ausgabe platziert werden soll. ■ Bei IFRAME-Zielen muss das location-Attribut den Namen des Frames angeben. ■ Um die Berichtsausgabe in einem neuen Fenster zu öffnen, geben Sie als Speicherort <code>*NEU</code> an. ■ Um die Ausgabe an ein benanntes Fenster umzuleiten, verwenden Sie ein Sternchen (*) gefolgt vom Fensternamen. Beispielsweise öffnet <code>*MEINEBERICHTE</code> ein neues Fenster mit dem Namen <code>MEINEBERICHTE</code>

Name	Erforderlich	Beschreibung
		<p>und verwendet diesen Link erneut bei jeder Aktivierung der Verknüpfung.</p> <p>Ausgaben im HTML-Format können ein DIV-Ziel haben. Alle anderen Ausgaben sollten IFRAME-Ziele oder ein Fenster zum Ziel haben.</p>
partId	false	<p>Dies wird verwendet, um den spezifischen Teil oder das spezifische Element der Ausgabe des SPW-Archivs zu kennzeichnen.</p> <p>Anmerkung: Für diese Funktionalität sind IBM® SPSS® Statistics-Adapter in der IBM® SPSS® Collaboration and Deployment Services-Umgebung erforderlich. Weitere Informationen finden Sie in der SPSS Statistics-Installationsdokumentation.</p>

Verwendungsbeispiel

Im folgenden Beispiel wird mithilfe des *ChartFRAME*-IFRAME-Tags ein Ausgabeort für eine Tabelle angegeben, die im IBM® SPSS® Collaboration and Deployment Services Repository gespeichert ist.

```
<outputLocation outputId="spsscr:///output/output_chart.png"
  location="ChartFRAME"/>
```

Falls die Attributwerte von Parameterwerten abhängig sein, verwenden Sie das *sourceLinkPrompt*-Tag, um Übereinstimmungen für die Parameter zu definieren. Falls eine Übereinstimmung gefunden wird, wird der Parameter durch sie ersetzt. Das folgende Beispiel definiert zum Beispiel zwei *outputLocation*-Tags mit Dateinamen, die abhängig von Parametern sind.

```
<repositoryItem name= "Call Center Score"
  inputURI= "spsscr:///job/Call Center"
  repositoryCredentialName="localhost"
  activate="ONDEMAND"/>
  <outputLocation outputId="spsscr:///output/output_tab_${JobParam1}.png"
    location="ChartFRAME"/>
  <outputLocation outputId="/output/output_chart_${JobParam2}.html"
    location="ReportDIV"/>
  <sourceLinkPrompt promptId="JobParam1" parameterValue="Jan" />
  <sourceLinkPrompt promptId="JobParam2"
    targetNameParameter="html_id_for_the_value" />
</repositoryItem>
```

Bei *JobParam1* wird im Namen ein Wert von *Jan* substituiert, wodurch *output_chart_Jan.png* bei *ChartFRAME* angezeigt wird.

Bei *JobParam2* wird der mit dem html-Steuerelement verknüpfte Wert für den Parameter im Namen substituiert. Falls dieser Wert *Illinois* lautet, erscheint bei *ReportDIV* die Datei *output_tab_Illinois.html*.

Das Tag "sourceLinkPrompt"

Das Tag `sourceLinkPrompt` weist die Elementparameter den Prompts zu, die ihre Werte bereitstellen. Dabei kann es sich um benutzerdefinierte HTML-Elemente, JavaScript-Funktionen, über das Tag `repositoryItemPrompt` erstellte Prompts oder direkt angegebene Werte handeln.

Das `sourceLinkPrompt`-Tag muss immer in einem `repositoryItem`-Tag verschachtelt sein. Wenn das Element ausgeführt wird, werden die Parameterwerte über die `sourceLinkPrompts` abgerufen.

Tag-Verschachtelung

Keines

Durchgeführte Validierungen

Keines

Erwartete Ausgabe

Keines

Tabelle 4-5
Attribute für das `sourceLinkPrompt`-Tag

Name	Erforderlich	Beschreibung
targetNameParameter	true	Name des Parameters wie im Repository-Element definiert. Geben Sie bei Scoring-Modellen mit mehreren Tabellen den Wert als <i>tabelle.parameter</i> an, wobei es sich bei <i>tabelle</i> um den Namen der Tabelle und bei <i>parameter</i> um den Namen des Parameters handelt.
promptId	false	Die <code>promptId</code> könnte die ID eines <code>reportPrompt</code> -Tags oder der Name eines HTML-Steuerelements sein. Wenn ein Prompt-Wert benötigt wird, wird das <code>reportPrompt</code> -Tag oder das HTML-Steuerelement verwendet, um den Prompt-Wert zu ermitteln. Entweder <code>promptId</code> , <code>parameterValue</code> oder <code>getValueJSFunction</code> sollte angegeben werden.

Name	Erforderlich	Beschreibung
parameterValue	false	Legt den Wert für den Parameter fest, anstatt ihn anzufordern. Dieser Tag sollte verwendet werden, wenn die Anwendung den Parameterwert kennt, wenn die JSP verarbeitet wird. In diesem Fall kann der Wert über dieses Attribut direkt angegeben werden. Wenn parameterValue angegeben ist, sollte promptId und getValueJSFunction nicht verwendet werden.
getValueJSFunction	false	Legt eine Funktion fest, die aufgerufen wird, um die Prompt-Werte abzurufen. Die Funktion sollte entweder einen einzelnen Wert oder ein Werte-Array zurückgeben. Dieses Attribut sollte den Funktionsnamen, Klammern und bei Bedarf auch Parameter enthalten. Für die Funktion <i>MyGetValues</i> , die einen Parameter aufnimmt, setzen Sie das Attribut beispielsweise auf <code>MyGetValues('myPromptID')</code> .
validateJSFunction	false	Legt eine Funktion fest, die aufgerufen wird, um eine Validierung des Prompts bereitzustellen. Die Funktion sollte <code>true</code> zurückgeben, wenn die Prompts gültig sind. Dieses Attribut sollte den Funktionsnamen, Klammern und bei Bedarf auch Parameter enthalten. Für die Funktion <i>MyValidate</i> , die einen Parameter aufnimmt, setzen Sie das Attribut beispielsweise auf <code>MyValidate('myPromptID')</code> .

Verwendungsbeispiel

Folgendes Berichtsbeispiel fordert die Eingabe zweier Parameterwerte über `repositoryItemPrompt`-Tags an. Die `sourceLinkPrompt`-Tags für den Bericht *CountrySales* verwenden die IDs für diese Prompts, um ihre Werte an den Bericht weiterzugeben.

```
<r:repositoryItem name="CountrySales"
  reportDefinitionURI="spsscr:///SampleReports/BIRT/CountryCity_cascadingParameter.rptdesign"
  repositoryCredentialName="repositoryCredential"
  outputType="HTML" activate="ONDEMAND" location="ReportDIV">
  <r:sourceLinkPrompt targetNameParameter="ShipCountry" promptId="IDFilter"/>
  <r:sourceLinkPrompt targetNameParameter="ShipCity" promptId="IDFilter1"/>
</r:repositoryItem>
```

```
<table width="95%" cellspacing="1" bgcolor="black">
  <tr bgcolor="white">
    <r:repositoryItemPrompt promptId="IDFilter" repositoryItemName="CountrySales1"
      parameterName="ShipCountry"/>
  </tr>
  <tr bgcolor="white">
    <r:repositoryItemPrompt promptId="IDFilter1" repositoryItemName="CountrySales1"
      parameterName="ShipCity"/>
  </tr>
</table>
```

Ein ähnlicher Ansatz kann bei Scoring-Modellen verfolgt werden. Folgendes Scoring-Beispiel fordert die Eingabe von fünf Parameterwerten über `input`-Tags an. Die `sourceLinkPrompt`-Tags für das Element *Configuration* verwenden die IDs für diese Prompts, um ihre Werte weiterzugeben.

```
<table>
  <tr>
    <td>Age</td>
    <td><input name="Age" id="Age" type="text"/></td>
  </tr>
  <tr>
    <td>Blood Pressure</td>
    <td><input name="BP" id="BP" type="text"/></td>
  </tr>
  <tr>
    <td>Cholesterol</td>
    <td><input name="Cholesterol" id="Cholesterol" type="text"/></td>
  </tr>
  <tr>
    <td>K</td>
    <td><input name="K" id="K" type="text"/></td>
  </tr>
  <tr>
    <td>Na</td>
    <td><input name="Na" id="Na" type="text"/></td>
  </tr>
</table>

<r:repositoryItem name="MyConfiguration" inputURI="scoring:///KMeans"
  repositoryCredentialName="repositoryCredential" outputType="HTML"
  activate="ONDEMAND" location="ReportIframe">
  <r:sourceLinkPrompt targetNameParameter="Age" promptId="Age"/>
  <r:sourceLinkPrompt targetNameParameter="BP" promptId="BP"/>
  <r:sourceLinkPrompt targetNameParameter="Cholesterol" promptId="Cholesterol"/>
  <r:sourceLinkPrompt targetNameParameter="Drug" parameterValue="DrugX"/>
  <r:sourceLinkPrompt targetNameParameter="K" promptId="K"/>
  <r:sourceLinkPrompt targetNameParameter="Na" promptId="Na"/>
</r:repositoryItem>
```

Der Wert für den Parameter *Drug* wird auf der Seite als *DrugX* angegeben, indem das Attribut `parameterValue` verwendet wird.

Wenn die Parameter, die von der Scoring-Konfiguration verwendet werden, in verschiedenen Tabellen definiert sind, stellen Sie dem Namen des Parameters den Namen der Tabelle voran und trennen Sie die beiden Namen mit einem Punkt voneinander. Beispiel: Wenn die Parameter *Age* und *BP* in *Table1* und die restlichen Parameter in *Table2* definiert werden, werden die Elemente `sourceLinkPrompt` wie folgt angegeben:

```
<r:repositoryItem name="MyConfiguration" inputURI="scoring:///KMeans"
  repositoryCredentialName="repositoryCredential" outputType="HTML"
  activate="ONDEMAND" location="Reportframe">
  <r:sourceLinkPrompt targetNameParameter="Table1.Age" promptId="Age"/>
  <r:sourceLinkPrompt targetNameParameter="Table1.BP" promptId="BP"/>
  <r:sourceLinkPrompt targetNameParameter="Table2.Cholesterol" promptId="Cholesterol"/>
  <r:sourceLinkPrompt targetNameParameter="Table2.Drug" parameterValue="DrugX"/>
  <r:sourceLinkPrompt targetNameParameter="Table2.K" promptId="K"/>
  <r:sourceLinkPrompt targetNameParameter="Table2.Na" promptId="Na"/>
</r:repositoryItem>
```

Das Tag “*sourceLinkRepositoryItem*”

Das `sourceLinkRepositoryItem`-Tag legt das Quellelement und die Quellvariablen fest, die verwendet werden, um die für das Element definierten Parameter zu erfüllen. Bei diesem Verfahren wird das übergeordnete Element mit den in den verschachtelten `sourceLinkVariable`-Tags definierten Parametern ausgeführt, wenn auf das Quellelement geklickt wird.

Dieses Tag muss immer in einem `repositoryItem`-Tag verschachtelt sein. Es sollte mindestens ein verschachteltes `sourceLinkVariable`-Tag enthalten.

Tag-Verschachtelung

Das `sourceLinkRepositoryItem`-Tag enthält mindestens ein `sourceLinkVariable`-Tag, das die Namen der Quellspalte und der Zielparameter angibt.

Erwartete Ausgabe

Keines

Tabelle 4-6
Attribute für das `sourceLinkRepositoryItem`-Tag

Name	Erforderlich	Beschreibung
sourceName	true	Name des <code>repositoryItem</code> , das als Quelle für die Beziehung verwendet wird
linkType	false	Legt fest, welche Aktion im Quellbericht die Ausführung des aktuellen Berichts auslöst. Derzeit wird nur ein <code>linkType</code> , <code>row</code> , unterstützt. Für diesen Typ wird der Zielbericht ausgeführt, wenn auf eine Zeile im Quellbericht geklickt wird. In künftigen Versionen werden gegebenenfalls weitere <code>linkTypes</code> unterstützt.

Verwendungsbeispiel

Im folgenden Beispiel wird `CityDetails` als Bericht festgelegt, der aufgrund einer Benutzeraktion im Bericht `AllCountries` ausgeführt wird.

```
<r:repositoryItem name="CityDetails"
  inputURI="spsscr:///SampleReports/BIRT/CountrySalesByCity.rptdesign"
  repositoryCredentialName="repositoryCredential"
  outputType="HTML" width="400" height="300"
  activate="ONDEMAND" location="SecondReportDIV">
  <r:sourceLinkRepositoryItem sourceReportName="AllCountries">
    <r:sourceLinkVariable columnName="ShipCountry"
      targetNameParameter="ShipCountry" />
  </r:sourceLinkRepositoryItem>
</r:repositoryItem>
```

Das Tag "`sourceLinkReport`"

Dieses Tag wurde verworfen. Verwenden Sie stattdessen das `sourceLinkRepositoryItem`-Tag

Das Tag "`sourceLinkVariable`"

Das `sourceLinkVariable`-Tag definiert die Zuordnung zwischen der Variablen oder Spalte, die im Quellelement verwendet werden soll, und dem im Zielelement definierten Parameter. Dieses Tag muss immer in einem `sourceLinkRepositoryItem`-Tag verschachtelt sein.

Tag-Verschachtelung

Keines

Durchgeführte Validierungen

Keines

Erwartete Ausgabe

Keines

Tabelle 4-7

Attribute für das `sourceLinkVariable`-Tag

Name	Erforderlich	Beschreibung
columnName	true	Für IBM® ShowCase®-Berichte legt dieses Attribut den Namen der Spalte im Quellbericht fest. Für Visualisierungsberichte enthält dieses Attribut die ID des <code>sourceVariable</code> - oder <code>derivedVariable</code> -Elements der Visualisierungsspezifikation. Derzeit werden nur kategoriale Variablen unterstützt.
targetNameParameter	true	Name des Parameters in der Zielabfrage

Verwendungsbeispiel

Im folgenden Beispiel wird die Variable `ShipCountry` im Bericht `AllCountries` dem Parameter `ShipCountry` im Bericht `CityDetails` zugeordnet.

```
<r:repositoryItem name="CityDetails"
  inputURI="spsscr:///SampleReports/Vis/CitiesBarChart.viz"
  repositoryCredentialName="repositoryCredential"
  outputType="png" width="400" height="300"
  activate="ONDEMAND" location="SecondReportDIV">
  <r:sourceLinkRepositoryItem sourceName="AllCountries">
    <r:sourceLinkVariable columnName="ShipCountry" targetNameParameter="ShipCountry"/>
  </r:sourceLinkRepositoryItem>
</r:repositoryItem>
```

Das Tag "actionHandler"

Definiert die Aktionshandler, die auf das Element angewendet werden sollen. Wenn Aktionshandler definiert werden, ist die automatische Verknüpfungseinstellung über `sourceLinkRepositoryItem` nicht mehr wirksam. Der Application Builder ist für die Ausführung der Zielelemente über die öffentliche Java-Skript-API `runRepositoryItem` verantwortlich.

Tag-Verschachtelung

Datenwerte, die als Parameter an die JavaScript-Funktion weitergegeben werden müssen, sollten über verschachtelte `actionParameter`-Tags definiert werden.

Tabelle 4-8
Attribute für das `actionHandler`-Tag

Name	Erforderlich	Beschreibung
event	true	Der Name des Ereignisses. Gültige Ereignisse sind: <ul style="list-style-type: none"> ■ <code>onclick</code> ■ <code>onmouseover</code> ■ <code>onmouseout</code>
function	true	Der Name der JavaScript-Funktion, die aufgerufen wird, wenn das Ereignis eintritt. Hier darf nur der Funktionsname ohne Klammern oder Parameter angegeben werden.
partId	false	Dies wird verwendet, um den spezifischen Teil des Berichts zu identifizieren, für den die Aktionen gelten sollen.

Verwendungsbeispiel

Das folgende `repositoryItem`-Tag definiert drei Aktionshandler, einen für jeden Ereignistyp, der eintreten könnte. Jeder Handler ruft eine eindeutige JavaScript-Funktion auf, die die anschließende Verarbeitung definiert.

```
<r:repositoryItem name="AllCountries"
  inputURI="spsscr:///SampleReports/BIRT/CountrySales.rptdesign"
  repositoryCredentialName="repositoryCredential"
  outputType="HTML"
  width="400" height="300"
  activate="ONLOAD" location="ReportDIV">
  <r:actionHandler event="onclick" function="myOnClick">
    <r:actionParameter name="ShipCountry"/>
  </r:actionHandler>
  <r:actionHandler event="onmouseover" function="myOnOver">
    <r:actionParameter name="ShipCountry"/>
  </r:actionHandler>
  <r:actionHandler event="onmouseout" function="myOnOut" />
</r:repositoryItem>
```

Das Tag "actionParameter"

Für jeden Datenwert im Element, der an die JavaScript-Funktion `actionHandler` weitergegeben werden muss, sollte ein `actionParameter` vorhanden sein. Dieses Tag muss immer im `actionHandler`-Tag verschachtelt sein.

Tag-Verschachtelung

Keines

Tabelle 4-9
Attribute für das `actionParameter`-Tag

Name	Erforderlich	Beschreibung
Name	true	<p>Name der Spalte oder Variable, die festlegt, welcher Wert aus den Berichtsergebnissen an die Funktion weitergegeben werden soll.</p> <ul style="list-style-type: none"> ■ Für Visualisierungsberichte ist der Name das <code>id</code>-Attribut des <code>sourceVariable</code>- oder <code>derivedVariable</code>-Elements. Derzeit werden nur kategoriale Variablen unterstützt. ■ Für IBM® ShowCase®-Berichte ist dies der Name der Spalte.

Verwendungsbeispiel

Folgendes Beispiel definiert einen `actionParameter` mit dem Namen `ShipCountry`, der an die JavaScript-Funktion `myOnClick` weitergeben wird, wenn der Benutzer auf den Bericht klickt.

```
<r:actionHandler event="onclick" function="myOnClick">
  <r:actionParameter name="ShipCountry"/>
</r:actionHandler>
```

Tab-Bibliothek-Beans

Das Framework enthält Tag-Bibliothek-Beans, die zusammen für verschiedene Zwecke verwendet werden können. Sie können beispielsweise zum Abrufen eines Daten-Sets verwendet werden, das dann verwendet werden kann, um benutzerdefinierte HTML-Steuerelemente zu erstellen.

Um die Beans zu verwenden, müssen Sie zunächst auf der JSP Verweise auf sie erstellen. Das geschieht über das `import`-Attribut der `page`-Aufbauregel.

```
<%@ page contentType="text/html;charset=utf-8"
  language="java"
  session="true"
  import="java.util.Map"
  import="java.util.HashMap"
  import="com.spss.report.taglib.bean.ReportBean"
  import="com.spss.report.taglib.bean.Credential"
%>
```

Diese Code-Beispiele für Beans verwenden die Tag-Bibliothek JavaServer Pages Standard Tag Library (JSTL), die über die `taglib`-Aufbauregel integriert werden sollte.

```
<%@ taglib uri="http://java.sun.com/jstl/core" prefix="c" %>
```

Weitere Informationen über JSTL finden Sie in der [Sun-Dokumentation](http://java.sun.com/products/jsp/jstl/) (<http://java.sun.com/products/jsp/jstl/>).

Credential-Bean

Die Credential-Bean definiert die Anmeldeinformationen, die von anderen Beans verwendet werden. Das nachstehende Code-Beispiel erstellt zwei Anmeldeinformationen und speichert sie in HashMap. Im nachstehenden Beispiel liefern die Anmeldeinformationen *localhost* die Anmeldeinformationen für das IBM® SPSS® Collaboration and Deployment Services Repository. Die Anmeldeinformationen *ps4008* gelten für die SQL Server-Datenquelle "ps4008", die von der Berichtsdefinition referenziert wird.

```
<%
Map credentialMap = new HashMap();
Credential repositoryCredential = new Credential("localhost","Native","admin","spss",null);
Credential datasourceCredential = new Credential("ps4008",null,"sa","sa",null);
credentialMap.put("localhost",repositoryCredential);
credentialMap.put("ps4008",datasourceCredential);
%>
```

ReportBean-Bean

Die ReportBean-Bean wird verwendet, um die Daten für ein Daten-Set abzurufen, das in einer Berichtsdefinition definiert ist. Der nachstehende Code verwendet die zuvor erstellte *credentialMap*, um ein Daten-Set abzurufen. Visualisierungsberichte unterstützen diese Funktion nicht.

```
<%-- Creating JavaBeans --%>
<jsp:useBean id="report" class="com.spss.report.taglib.bean.ReportBean">
  <jsp:setProperty name="report" property="reportDefinitionURI"
    value="file:///d:/SPSS/ps4008/Test.dbq" />
  <jsp:setProperty name="report" property="repositoryCredentialName"
    value="localhost" />
  <jsp:setProperty name="report" property="host" value="localhost" />
  <jsp:setProperty name="report" property="port" value="8080" />
  <jsp:setProperty name="report" property="dataSetName"
    value="DataSet1" />
  <jsp:setProperty name="report" property="credentialMap"
    value="<%=credentialMap%>" />
</jsp:useBean>
```

Folgende Eigenschaften werden in diesem Code verwendet:

- Die Eigenschaft *reportDefinitionURI*, die den Speicherort des Berichts angibt.
- Die Eigenschaft *repositoryCredentialName*, die den Host angibt.
- Die Eigenschaft *port*, die den Port definiert.
- Die Eigenschaft *dataSetName*, die den Namen des Daten-Sets angibt, wie er in der Berichtsdefinition für BIRT Report Designer for IBM® SPSS®-Berichte definiert ist. Diese Eigenschaft gilt nicht für IBM® ShowCase®-Berichte und sollte ausgelassen werden.
- Die Eigenschaft *credentialMap*, die einen Verweis auf eine HashMap mit den zu verwendenden Anmeldeinformationen definiert.

Die `ReportBean`-Bean kann anschließend ausgeführt werden, um das Daten-Set zurückzugeben. Die Daten können verwendet werden, um ein Listensteuerelement wie im nachstehenden Code dargestellt zu erstellen.

```
<SELECT style="WIDTH :250 px" ID="EmployeeID_Prompt" NAME="EmployeeID_Prompt"
  TABINDEX="2">
<c:forEach var="row" items="{report.rows}">
  <c:forEach var="column" items="{row.columns}">
    <c:if test="{column.name == "EmployeeID"}">

      <OPTION VALUE='<c:out value="{column.value}" />'>
        <c:out value="{column.value}" />
      </OPTION>
    </c:if>
  </c:forEach>
</c:forEach>
</SELECT>
```

SearchBean bean

Die Bean `SearchBean` bietet einen Abfragemechanismus für das Auffinden von Inhalt im Repository, der bestimmten Kriterien entspricht. Die Bean kann zum Beispiel eine Liste mit Daten-Provider-Definitionen und IBM® SPSS® Statistics-Datendateiquellen (*.sav*) in IBM® SPSS® Collaboration and Deployment Services Repository abrufen, die einem bestimmten Suchkriterium entsprechen. Der folgende Code definiert Bean-Eigenschaften zur Abfrage aller Daten-Provider-Definitionen und SPSS Statistics-Datenquellen, die die MIME-Typen verwenden, die diesen Quellen zugeordnet sind.

```
<jsp:useBean id="data_sources"
  class="com.spss.report.taglib.bean.SearchBean" scope="page">
  <jsp:setProperty name="data_sources" property="request" value="<%= request %>" />
  <jsp:setProperty name="data_sources" property="credentialName"
    value="AuthenticationCredential" />
  <jsp:setProperty name="data_sources" property="searchQuery"
    value="<%= '('$$$search/mimetype='application/x-vnd.spss-spss-data' or "
      + "'$$$search/mimetype='application/x-vnd.spss-statistics-data' or "
      + "'$$$search/mimetype='application/x-vnd.spss-data-provider')' %>" />
</jsp:useBean>
```

Folgende Eigenschaften werden in diesem Code verwendet:

- Die Eigenschaft *request*, die ein `HttpServletRequest`-Objekt definiert.
- Die Eigenschaft *credentialName*, die die für eine Verbindung zum Repository erforderlichen Anmeldeinformationen angibt. In diesem Fall entspricht der Wert dem Credential *AuthenticationCredential*, das mit Hilfe des `credential`-Tags definiert wurde.
- Die Eigenschaft *searchQuery*, die eine Zeichenfolge angibt, die das Suchkriterium enthält. Die Struktur dieser Zeichenfolge entspricht der Syntax für den Parameter `dataset.search.criteria`, der für URL-Zeichenfolgen verwendet wird.

Die SearchBean-Bean kann anschließend ausgeführt werden, um die übereinstimmenden Datenquellen zurückzugeben. Der folgende Code gibt Name, Änderungsdatum, Versionsbeschriftung und Autor-Metadaten für die Datenquellen in einer Tabelle an.

```
<Table border="0" height="100%" width="100%" cellpadding="0" cellspacing="0">
<tr>
<td align="center" bgcolor="#EEEEEE" style="padding-top:5px; padding-bottom:5px;">
  Datenquelle
</td>
<td align="center" bgcolor="#EEEEEE" style="padding-top:5px; padding-bottom:5px;">
  Änderungsdatum
</td>
<td align="center" bgcolor="#EEEEEE" style="padding-top:5px; padding-bottom:5px;">
  Versionsbeschriftung
</td>
<td align="center" bgcolor="#EEEEEE" style="padding-top:5px; padding-bottom:5px;">
  Autor
</td>
</tr>
<c:forEach var="data_source" items="{data_sources.records}" varStatus="status"
  begin="0" end="3" step="1">
<tr>
<td align="center" bgcolor="#EEEEEE" style="padding-top:5px; padding-bottom:5px;">
  <c:out value="{data_source.title}" />
</td>
<td align="center" bgcolor="#EEEEEE" style="padding-top:5px; padding-bottom:5px;">
  <c:out value="{data_source.modifiedDate}" />
</td>
<td align="center" bgcolor="#EEEEEE" style="padding-top:5px; padding-bottom:5px;">
  <c:out value="{data_source.versionLabel}" />
</td>
<td align="center" bgcolor="#EEEEEE" style="padding-top:5px; padding-bottom:5px;">
  <c:out value="{data_source.author}" />
</td>
</tr>
</c:forEach>
</Table>
```

PevMetaDataBean-Bean

Die PevMetaDataBean-Bean ruft Variablen-Metadaten aus Daten-Provider-Definitionen und IBM® SPSS® Statistics-Datendateiquellen (.sav) ab. Der folgende Code definiert Eigenschaften für die Bean zur Abfrage einer .sav-Datei.

```
<jsp:useBean id="variables"
  class="com.spss.report.taglib.bean.PevMetaDataBean" scope="page">
<jsp:setProperty name="variables" property="request" value="<%= request %>" />
<jsp:setProperty name="variables" property="dataseturi"
  value="spsscr:///sav_files/demo.sav" />
<jsp:setProperty name="variables" property="credentialName"
  value="AuthenticationCredential" />
```

```
</jsp:useBean>
```

Folgende Eigenschaften werden in diesem Code verwendet:

- Die Eigenschaft *request*, die ein `HttpServletRequest`-Objekt definiert.
- Die Eigenschaft *dataseturi*, die den URI für die Datendatei oder die Daten-Provider-Definition, die die Variablen enthält, angibt.
- Die Eigenschaft *credentialName*, die die für eine Verbindung zum Repository erforderlichen Anmeldeinformationen definiert. In diesem Fall entspricht der Wert dem Credential *AuthenticationCredential*, das mit Hilfe des `credential`-Tags definiert wurde.

Die `PevMetaDataBean`-Bean kann dann ausgeführt werden, um die Metadaten für die Variablen im Daten-Set zurückzugeben. Der folgende Code zeigt die Metadaten in der Tabelle an.

```
<Table border="0" height="100%" width="100%" cellpadding="0" cellspacing="0">
  <tr>
    <td align="center" bgcolor="#EEEEEE" style="padding-top:5px;
      padding-bottom:5px;">
      Variablenname
    </td>
  </tr>
  <c:forEach var="group" items="{variables.variablesMetaData}" >
    <c:forEach var="v" items="{group.variableMetaData}" varStatus="status"
      begin="0" end="3" step="1">
      <tr>
        <td align="center" bgcolor="#EEEEEE" style="padding-top:5px;
          padding-bottom:5px;">
          <c:out value="{status.count}" /> <c:out value="{v.name}" />
        </td>
      </tr>
    </c:forEach>
  </c:forEach>
</Table>
```

ScoringBean-Bean

Die `ScoringBean`-Bean ruft eine Liste von Scoring-Konfigurationen für ein angegebenes Modell ab, die auf eine Scoring-Anforderung reagieren können. Die `getScoringConfigurations`-Methode des Beans akzeptiert die folgenden Parameter:

- Den Parameter *credential*, der die Anmeldeinformationen für den Zugriff auf das IBM® SPSS® Collaboration and Deployment Services Repository angibt, die mit der `Credential`-Bean definiert wurden.
- Den Parameter *modelLocationUri*, der den URI für ein Modell im IBM SPSS Collaboration and Deployment Services Repository angibt.

Statt einem `Credential`-Bean-Objekt können die folgenden beiden Parameter zur Angabe von Anmeldeinformationen verwendet werden:

- Den Parameter `request`, der ein `HttpServletRequest`-Objekt angibt.
- Den Parameter `credentialName`, der die Anmeldeinformationen zur Herstellung einer Verbindung zum IBM SPSS Collaboration and Deployment Services Repository angibt, die mit dem `credential`-Tag definiert wurden.

Der folgende Code ruft für das Modell `KMeans.xml` die Scoring-Information ab, die auf eine Scoring-Anforderung mit einem durch das `credential`-Tag definierte Anmeldeinformationen antworten kann.

```
<r:credential name="repositoryCredential" provider="Native"
  username='<%= request.getParameter("userid")%>'
  password='<%= request.getParameter("password")%>' />

<%
  String[] configurations = ScoringBean.getScoringConfigurations(request,
    "repositoryCredential", "spsscr:///Sample/KMeans.xml");
%>
```

Das von der Bean zurückgegebene Array kann verwendet werden, um ein Formular zu füllen, aus dem ein Benutzer eine für künftige Scoring-Vorgänge zu verwendende Scoring-Konfiguration auswählen kann.

```
<form id="selectConfigurationForm" target="ScoringIframe" method="POST">
  <div style="display:none">
    <input name="userid" type="text" value="<%= request.getParameter("userid")%>"/>
    <input name="password" type="text" value="<%= request.getParameter("password")%>"/>
  </div>
  Scoring-Konfiguration auswählen:
  <select name="selectedConfiguration" onchange="onSelectConfiguration(this)">
    <option></option>
    <%
      for (int i=0; i < configurations.length; i++)
      {
    %>
      <option value="<%= configurations[i].replaceAll(" ", "%20")%>"
        <%= configurations[i] %></option>
    <%
      }
    %>
  </select>
</form>
```

JavaServer Pages-Beispiele

IBM® SPSS® Collaboration and Deployment Services enthält eine Vielzahl an JSP-Beispielen, die die Verwendung der Tag-Bibliothek illustrieren. Die Beispiele sind in folgenden Kategorien zusammengefasst:

- **Reporting.** Interaktive Nutzung von BIRT Report Designer for IBM® SPSS®- und Visualisierungsberichten einschließlich der Ausführung eines zweiten Berichts aufgrund einer Auswahl. Über folgenden Link können Sie auf die Beispiele zugreifen:

`http://<server-name>:<port>/IBMSPSSTagLib/index.html`

- **Scoring.** Generieren von Scores für ein Vorhersagemodell, das zum Scoring konfiguriert wurde. Schließt mehrere Ansätze zum Bereitstellen konfigurierter Modelle mit Daten zum Scoring ein. Über folgenden Link können Sie auf die Beispiele zugreifen:

`http://<server-name>:<port>/scoringTagLib/index.html`

Wenn der URL für ein Beispielsatz keine Einführungsseite liefert, wurde das Paket oder die war-Datei mit den Beispielen nicht auf dem IBM® SPSS® Collaboration and Deployment Services Repository-Server bereitgestellt. Verwenden Sie das Tool “Package Manager”, um das gewünschte Paket oder die war-Datei gemäß der Dokumentation für Ihren Anwendungsserver bereitzustellen. Die bereitzustellenden war-Dateien und erforderlichen Beispieldateien befinden sich im Verzeichnis `./components/taglib/Samples/TagLib` der Repository-Installation.

Klicken Sie auf der Einführungsseite für die Beispiele für ein beliebiges Beispiel auf Quelle anzeigen, um dessen Quellcode anzuzeigen. Um die Funktionalität kennenzulernen, führen Sie die Beispiele über die Seite aus, indem Sie auf Ausführen klicken. Um eine erfolgreiche Ausführung zu ermöglichen, sind jedoch folgende Elemente erforderlich:

- Beispielressourcen in einer bestimmten Ordnerstruktur im IBM SPSS Collaboration and Deployment Services Repository.
- Gültige Anmeldedaten für den Zugriff auf die Ressourcen, auf die die Beispiele verweisen.

Instruktionen zur Konfiguration der Umgebung für erfolgreiche Beispielausführung stehen auf der Einführungsseite für die Beispiele zur Verfügung.

Portalintegration

Die IBM® SPSS® Collaboration and Deployment Services-Webdienstarchitektur bietet die Möglichkeit der Integration mit Portalservern. Dadurch können stark angepasste Inhalte über plugfähige Komponenten der Benutzeroberfläche bereitgestellt werden, die über Webdienste Markupcode-Fragmente produzieren, die auf einer Portalseite zusammengestellt werden. In der Regel wird eine Portalseite als eine Gruppe von nicht überlappenden Fenstern angezeigt, wobei jedes Fenster ein Inhaltssegment enthält. Beispiele für Portalanwendungen sind E-Mails, Wetterberichte, Diskussionsforen und Nachrichten. IBM SPSS Collaboration and Deployment Services-Portale können ebenfalls verwendet werden, um angepasste Inhalte wie Ausgaben von Berichten und analytischen Verarbeitungen, Tabellen, Diagrammen usw. bereitzustellen.

Das Repository unterstützt Portalintegration basierend auf dem Standard JSR 168. Der Standard JSR 168, der von der Java Community Process Group (<http://jcp.org>) vorgeschlagen wurde, ermöglicht die Interoperabilität für Portlets zwischen unterschiedlichen Webportalen. Diese Spezifikation definiert eine Reihe von APIs für die Zusammenarbeit zwischen dem Portlet-Container und dem Portlet und befasst sich mit Bereichen wie Personalisierung, Präsentation und Sicherheit. Die Implementierung von JSR 168 umfasst IBM Web Portal von WebSphere, Oracle Application Server Portal 10g, BEA WebLogic Portal, Vignette Portal, Sun Portal Server und JBoss.

Das Repository unterstützt zudem die Portalintegration mit Microsoft SharePoint Server und Web Parts.

Offiziell unterstützte Portalumgebungen sind:

- WebSphere Portal Server 6.1
- Weblogic Portal Server 10.0
- JBoss Portal Server 2.6.1
- Sun Java Enterprise System 5
- Microsoft Sharepoint 2007 Server

Das Repository kann auch mit anderen Portalumgebungen auf Basis von JSR 168 und J2SE 5.0 integriert werden.

IBM® SPSS® Collaboration and Deployment Services Portlet und IBM® SPSS® Collaboration and Deployment Services Web Part können verwendet werden, um Repository-Inhalte für Portalbenutzer bereitzustellen. Die Architektur ermöglicht zudem die Erstellung von benutzerdefinierten JSR 168-kompatiblen SharePoint Web Parts, die IBM SPSS Collaboration and Deployment Services-Webdienste verwenden.

Installation

IBM® SPSS® Collaboration and Deployment Services-Portalkomponenten werden auf Installationsdatenträger 2 in */PORTLET* als *IBMSPSSPortlet.war* (Portlet) und *IBMSPSSWebPart.wsp* (Web Part) bereitgestellt.

IBM SPSS Collaboration and Deployment Services Portlet-Installation

- Die Installation von *IBMSPSSPortlet.war* hängt vom jeweiligen Portalservertyp ab. Genauere Informationen finden Sie in der Dokumentation des Portalserveranbieters.

IBM SPSS Collaboration and Deployment Services Web Part-Installation

Installationsvoraussetzungen für SharePoint Web Part:

- Microsoft SharePoint 2007
- Microsoft Web Service Enhancement 2.0 (WSE 2.0 SP3)

So installieren Sie IBM® SPSS® Collaboration and Deployment Services Web Part:

1. Kopieren Sie die Datei *IBMSPSSWebPart.wsp* vom Repository-Installationsdatenträger an einen vordefinierten Speicherort auf dem SharePoint-Host, beispielsweise in das Verzeichnis *c:\temp*.
2. Führen Sie über das Verzeichnis */bin* der SharePoint-Serverinstallation folgende Befehle aus:

```
stsadm -o addsolution -filename c:\tmp\IBMSPSSwebpart.wsp  
stsadm -o deploysolution -name IBMSPSSwebpart.wsp -immediate -allowgacdeployment -url http://<hostname>
```

3. Verwenden Sie die Verwaltungsdienstprogramme von SharePoint, um Web Part der Web Part-Galerie hinzuzufügen und es anschließend bereitzustellen. Weitere Informationen finden Sie in der Microsoft SharePoint-Dokumentation.

Nach der Installation muss die Komponente für den Zugriff auf eine bestimmte Ressource im Repository konfiguriert werden. Zudem müssen Voreinstellungen für die Komponente eingerichtet werden.

Konfiguration

Nach der Installation der Portalkomponente und der Fertigstellung des Seitenlayouts für das Portal werden Sie aufgefordert, die Komponente für den Zugriff auf eine Repository-Ressource zu konfigurieren. Im Allgemeinen werden bei der Konfiguration des Portalzugriffs der Repository-Server definiert, die Repository-Anmeldedaten angegeben, die Ressource, die im Portal bereitgestellt wird, ausgewählt und bei Bedarf Datenquellen-Credentials und Standard-Prompt-Werte festgelegt. Zudem können Sie das Erscheinungsbild und das Verhalten der Komponenten konfigurieren, indem Sie die entsprechenden Voreinstellungen vornehmen.

Konfigurieren von IBM SPSS Collaboration and Deployment Services Portlet

Öffnen Sie die Portlet-Konfigurationsseite. Die Seite wird je nach Portalservertyp unterschiedlich geöffnet.

1. Geben Sie den Repository-Host und -Port an und ob für den Server eine sichere Verbindung erforderlich ist.

Abbildung 5-1

Portlet-Konfiguration: Angabe des Repositorys

The screenshot shows a configuration window titled "IBM SPSS Collaboration and Deployment Services Portlet – Konfiguration". The main heading is "Repositoryserver angeben". It contains three input fields: "Server:" with the value "10.96.41.194", "Port:" with the value "8080", and "Sicher:" with an unchecked checkbox. At the bottom, there is a "Nächster" button and a copyright notice: "Licensed Materials - Property of SPSS Inc., an IBM Company. © Copyright 2004, 2010 SPSS Inc., an IBM Company."

2. Geben Sie die Benutzer-Anmeldedaten für das Repository und den Sicherheits-Provider für die Anmeldeauthentifizierung an.

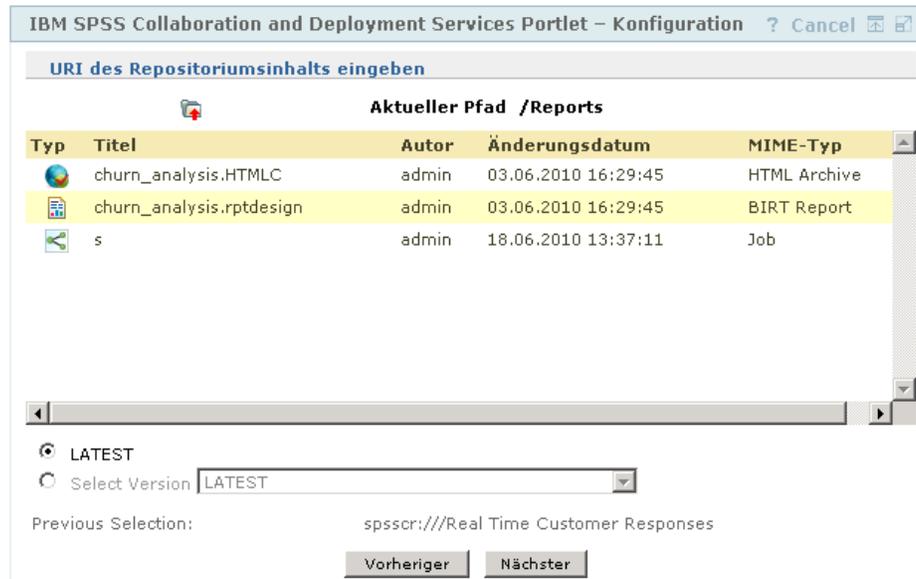
Abbildung 5-2

Portlet-Konfiguration: Angabe der Repository-Anmeldeinformationen

The screenshot shows the same configuration window, now at the "Anmeldeinformationen eingeben" step. It contains three input fields: "Anmeldename:" with the value "admin", "Kennwort:" with masked characters "****", and "Provider:" with a dropdown menu showing "Lokales Benutzer-Repos". At the bottom, there are "Vorheriger" and "Nächster" buttons.

3. Wählen Sie die Repository-Ressource, die im Portal bereitgestellt werden soll. Vergewissern Sie sich, dass die richtige Ressourcenversion angegeben ist.

Abbildung 5-3
Portlet-Konfiguration: Auswahl der Ressource



4. Geben Sie bei Bedarf die Credentials für die Datenquelle an, die von der Ressource referenziert wird. Wenn ein Bericht beispielsweise eine Datenbank verwendet, müssen die Credentials für die Datenbank angegeben werden. Beachten Sie, dass je nach Ressource gegebenenfalls die Credentials für mehrere Datenquellen angegeben werden müssen.

Abbildung 5-4
Portlet-Konfiguration: Angabe der Datenquellen-Credentials



5. Wenn die Ressource Eingabeaufforderungen enthält (zum Beispiel bietet einen Bericht unter Umständen die Möglichkeit der dynamischen Auswahl von Werten), geben Sie die Standard-Prompt-Einstellungen ein.

Abbildung 5-5
Portlet-Konfiguration: Angabe der Standard-Prompt-Werte

6. Vergewissern Sie sich, dass die Konfigurationsinformationen korrekt sind. Um von Neuem zu beginnen, klicken Sie auf Aktualisieren.

Abbildung 5-6
Portlet-Konfiguration: Bestätigungsseite

7. Klicken Sie auf Weiter, um mit der Anzeige der Ressource fortzufahren.

Abbildung 5-7
Portlet-Konfiguration: Abschlussmeldung

Die Portlet-Einstellungen können nach der ersten Konfiguration geändert werden: zum Beispiel kann auf eine andere Repository-Ressource verwiesen werden.

- Bestimmte Aspekte des Erscheinungsbilds und des Verhaltens des Portlets werden über seine Voreinstellungen festgelegt. Folgende Voreinstellungen stehen zur Verfügung:

Voreinstellung	Beschreibung
Ablauf-Cache	Ablaufzeitraum für den Portlet-Cache, d. h. die Zeit in Sekunden, nach der die Portlet-Ausgabe abläuft. -1 definiert, dass die Ausgabe nie abläuft. Der Standardwert lautet 600.
Protokollmeldungen	Gibt an, ob Portlet-Meldungen an die Portalserver-Protokolldatei angehängt werden. Der Standardwert ist NEIN.
erneut eingeben-ds-Anmeldung	Legt fest, ob der Benutzer bei jeder Anmeldung im Portal Datenquellen-Credentials für die Portlet-Instanz eingeben muss. Der Standardwert ist NEIN.
erneut eingeben-Parameter	Legt fest, ob der Benutzer bei jeder Anmeldung im Portal Prompt-Werte für die Portlet-Instanz eingeben muss. Der Standardwert ist NEIN.

Voreinstellung	Beschreibung
aktualisieren-Parameter	Legt fest, ob der Benutzer unterschiedliche Parameterwerte eingeben und den Inhalt basierend auf diesen Werten erneut anzeigen kann. Der Standardwert ist NEIN.
verwenden-Einzelanmeldung	Gibt an, ob das Portlet mit Einzelanmeldung verwendet wird. Der Standardwert ist NEIN.
validieren-Eingangsparameter	Aktiviert die Validierung von Eingabeparametern, um vor Site-übergreifenden Skript-Attacken zu schützen. Der Standardwert ist JA.
Fenster-Höhe	Die Höhe des Portlet-Fensters (in Pixel). Der Standardwert lautet 750.
Fenster-Titel	Beschreibender Name für die Portlet-Instanz.
Fenster-Breite	Die Breite des Portlet-Fensters (in Prozent). Der Standardwert ist 100 %.

Voreinstellungen werden über die Server-Administrationsdienstprogramme festgelegt. Der Zugriff auf diese Programme ist vom Servertyp abhängig.

Konfigurieren von IBM SPSS Collaboration and Deployment Services Web Part

Die Web Part-Konfiguration setzt sich aus denselben grundlegenden Schritten wie die Portlet-Konfiguration zusammen: Einrichten des Zugriffs auf die Repository-Ressource und der Konfigurationsoptionen. Beachten Sie, dass die Anzahl der angezeigten Elemente im Repository-Baum (bei der Auswahl der Ressource) über eine zusätzliche Konfigurationsoption gesteuert wird.

Einzelanmeldung

IBM® SPSS® Collaboration and Deployment Services ermöglicht Zugang über Einzelanmeldung, wobei eine spezielle Konfiguration des Portalserver erforderlich sein kann, um ihn für das Portlet oder den Web-Part zu aktivieren. Die Prozeduren für die Aktivierung der Einzelanmeldung unterscheiden sich je nach Portalserver. Beispiel für eine JBoss-Portalkonfiguration:

- Die Datei `<JBoss-Installationsordner>/bin/run.bat` muss so geändert werden, dass sie Java-Argumente für Kerberos-basierte Einzelanmeldungen wie im nachfolgenden Beispiel enthält.

```
set SSO_OPTS=-Djava.security.krb5.realm=SSOREALM.COM
-Djava.security.krb5.kdc=kdchost.ssorealm.com
-Djavax.security.auth.useSubjectCredsOnly=false
```

```
set JAVA_OPTS=%JAVA_OPTS% -Dprogram.name=%PROGNAME% %SSO_OPTS%
```

```
set JAVA_OPTS=%JAVA_OPTS% -Xms128m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=256m
```

- Der Datei `<JBoss-Installationsordner>/server/default/conf/login-config.xml` muss der folgende Abschnitt hinzugefügt werden.

```
<application-policy name="com.sun.security.jgss.initiate">
  <authentication>
    <login-module code = "com.sun.security.auth.module.Krb5LoginModule" flag = "required">
```

```
<module-option name="useTicketCache">true</module-option>
<module-option name="debug">false</module-option>
</login-module>
</authentication>
</application-policy>
```

Bekannte Probleme

- Wenn das Portlet mit JBoss-Portal verwendet wird, kann die Repository-Baumansicht möglicherweise nicht erweitert werden. Um dieses Problem zu beheben, ändern Sie in der Datei `<JBoss-Installationsordner>/bin/run.bat` (`run.sh` unter UNIX) die Neugenerierung und die permanente Generierungsgröße, indem Sie `JAVA_OPTS` die folgenden Argumente hinzufügen:

```
-XX:MaxNewSize=256m -XX:MaxPermSize=256m
```

- Cookie-Einstellungen im Safari-Browser können verhindern, dass einige Repository-Artefakte im Portlet angezeigt werden, ohne dass zuvor Anmeldeinformationen angefordert wurden. Die Cookie-Einstellung des Browsers sollte auf *Immer* anstelle von *Nur von Websites, die ich besuche* eingestellt sein, um wiederholte Anforderungen von Anmeldedaten zu verhindern.

HTML-Archiv

Ein HTML-Bericht beinhaltet normalerweise einige HTML-Dateien, die eine Reihe von referenzierten Bildern anzeigen, wobei Stylesheets verwendet werden, um das Erscheinungsbild der Ausgabe zu kontrollieren. Aufgrund der Anzahl der enthaltenen Dateien können die Verwaltung und die gemeinsame Verwendung dieser Ausgabe eine Herausforderung darstellen. Falls eine Datei fehlt oder falsch referenziert wird, werden die Seiten nicht korrekt dargestellt.

Das HTML Archive- bzw. HTMLC-Format widmet sich dem Problem der Verwaltung zahlreicher intern verknüpfter Dateien, indem es alle zugeordneten HTML-Artefakte in einer einzelnen, browserübergreifenden Datei zusammenfasst. Das IBM® SPSS® Collaboration and Deployment Services Repository enthält einen Viewer, der die Anzeige des Archiv-Inhalts über eine Reihe von Client-Anwendungen ermöglicht. Beim Zugriff auf eine im Repository gespeicherte HTMLC-Datei werden Querverweise innerhalb des Archivs im Hintergrund durch vollständige Pfade ersetzt, welche die Archiv-Datei referenzieren. Dies ermöglicht die vollständige Auflösung von Links zu Dateien im Archiv, so dass diese korrekt dargestellt werden.

Dateistruktur

Eine HTMLC-Archivdatei enthält folgende Elemente:

- Eine primäre HTML-Datei im Stammverzeichnis des Archivs. Beim Rendern eines HTMLC-Archivs verwendet der Viewer die erste Datei mit einer *.html*-Erweiterung im Stammverzeichnis des Archivs als die primäre Datei.
- Sekundäre Dateien, die von der primären Datei referenziert werden, z. B. Cascading Stylesheets, Bilder, Javascript oder andere HTML-Dateien. Sekundäre Dateien können sich in jedem Ordner innerhalb des Archivs befinden.

Für alle Referenzen auf Dateien innerhalb des Archivs sollten relative Pfade verwendet werden.

Erstellung von HTMLC-Dateien

HTMLC-Dateien können bei der Arbeit mit Bericht-Designs, die im BIRT Report Designer für IBM® SPSS® gespeichert sind, in IBM® SPSS® Collaboration and Deployment Services Repository erstellt werden. Benutzerdefinierte HTMLC-Dateien können jedoch auch über ein Programm zur Dateiarchivierung, z. B. über das Java Archive-Tool oder WinZip, erstellt werden. Manuelles Erstellen einer HTMLC-Datei:

1. Erstellen Sie im Dateisystem die Struktur für die Dateien.
2. Erstellen Sie ein Archiv, welches diese Dateien und Ordner enthält, und geben Sie dabei die Dateierweiterung *.htmlc* für die Ausgabedatei an.

Die Dateien im Archiv können manuell oder automatisch erstellt werden. In IBM® SPSS® Statistics können Sie die Ergebnisse einer Analyse z. B. als HTML exportieren. Die resultierenden HTML- und Bilddateien können als HTMLC-Datei archiviert werden. Alternativ können Sie einen HTML-Editor verwenden, um zu archivierende Seiten manuell zu erstellen.

Beispiel für eine benutzerdefinierte HTMLC-Datei

Verwenden Sie für dieses Beispiel die Ordnerstruktur, die in der [“Archivdateien”](#) Abbildung dargestellt ist.

Abbildung 6-1
Archivdateien



Die HTML-Datei *gss.html* referenziert Bilder aus dem Ordner *images* und verwendet Stile aus einem Cascading Stylesheet im Ordner *css*. Unter Verwendung des Java-Archiv-Tools erstellt der folgende Befehl eine HTMLC-Datei namens *custom.HTMLC*, welche die Dateien enthält.

```
jar -cvfM custom.HTMLC gss.htm images css
```

Der Inhalt des resultierenden Archivs wird in der [“HTMLC-Archiv”](#) Abbildung angezeigt.

Abbildung 6-2
HTMLC-Archiv

Name	Typ	Pfad
my_styles.css	Kaskadierendes Stylesheet-Dok...	css\
gss.html	HTML-Dokument	
gss1.jpg	JPEG-Bild	images\
gss2.jpg	JPEG-Bild	images\

Durch Speichern dieses einzelnen Archivs im Repository kann die *gss.html*-Seite in Repository-Clients, wie IBM® SPSS® Collaboration and Deployment Services Deployment Portal oder IBM® SPSS® Collaboration and Deployment Services Deployment Manager, angezeigt werden, wobei ihre referenzierten Grafiken die definierten Stile verwenden.

Anpassungsbeispiel

Die Modellverwaltungsseite von IBM® SPSS® Collaboration and Deployment Services Deployment Portal bietet die Möglichkeit zur kontinuierlichen Überwachung der Leistung der im IBM® SPSS® Collaboration and Deployment Services Repository implementierten Modelle. Diese Modelldateien sind mit Jobs verknüpft, deren Ausführung bei Bedarf angefordert oder geplant werden kann. Die Dateien werden durch Verwendung von IBM® SPSS® Modeler erstellt. Modellbewertungs- und Champion-Challenger-Jobs werden mithilfe von IBM® SPSS® Collaboration and Deployment Services Deployment Manager konfiguriert und ausgeführt, Deployment Portal dient lediglich der Anzeige der Ergebnisse. Die als Bereiche auf der Seite “Modellverwaltung” angezeigten Informationen können folgende Elemente enthalten:

- Liste der Modelle mit der besten und schlechtesten Leistung
- Trends der Modellleistung
- Champion-Modelle
- Liste aller verfügbaren Modelldateien

Die Optionen im Konfigurationsbereich ermöglichen die Definition der Anzeigeparameter sowie das Ein- bzw. Ausblenden einzelner Registerkarten.

Informationen zur Verwendung der Seite “Model Management” finden Sie in der Hilfe zu Deployment Portal.

Die Benutzeroberfläche besteht im Wesentlichen aus einer einzelnen Java-Serverseite (JSP) *MMDMaster.jsp*. Bei den Komponenten, die auf der Seite angezeigt werden, handelt es sich entweder um BIRT Report Designer for IBM® SPSS®-Berichte oder Visualisierungsberichte. Diese Berichte werden mit der IBM® SPSS® Collaboration and Deployment Services Tag Library dargestellt. Die Seite selbst ist über das Framework “Tab Extension” in Deployment Portal integriert.

IBM SPSS Collaboration and Deployment Services Tag Library

Die IBM® SPSS® Collaboration and Deployment Services Tag Library unterstützt die Ausführung der BIRT Report Designer for IBM® SPSS®- und Visualisierungsberichte, die einen Großteil des Inhalts der Seite “Model Management” erzeugen. Die Tag-Bibliothek unterstützt zudem die Interaktivität zwischen Berichten, so dass ein Quellbericht einen Zielbericht aufrufen kann. Der Quellbericht gibt Parameter an den Zielbericht zur Verarbeitung weiter.

Berichtdefinitionen

Die Berichtdefinitionen, die von der Seite “Model Management” verwendet werden, sind in folgendem Verzeichnis in der IBM® SPSS® Collaboration and Deployment Services Repository-Installation gespeichert:

`<Installationsverzeichnis>\components\peb-mmd\reports`

Um die Berichte zu untersuchen, öffnen Sie die Dateien in BIRT Report Designer for IBM® SPSS®. Die Visualisierungsberichte können über die IBM® SPSS® Visualization Designer oder einen Text- oder XML-Editor geöffnet werden.

Die Berichte werden für Referenzzwecke bereitgestellt und sollten nicht direkt bearbeitet werden. Änderungen an den Berichten werden von IBM Corp. nicht unterstützt. Sie können die Berichte jedoch kopieren und die Kopien je nach Bedarf ändern.

Ausführen von BIRT Report Designer for IBM SPSS-Berichten

Auf der Seite “Model Management” werden vier Master-Berichte auf vier Registerkarten bereitgestellt. Diese Registerkarten finden Sie in Abbildung “Registerkarten der Seite “Model Management””.

Abbildung 7-1
Registerkarten der Seite “Model Management”



Jede Registerkarte entspricht einem Master-BIRT Report Designer for IBM® SPSS®-Bericht. Wenn die Registerkarte geladen wird, wird der Master-Bericht, der dieser Registerkarte entspricht, über das Framework der IBM® SPSS® Collaboration and Deployment Services Tag Library ausgeführt.

Für jeden Bericht ist auf der Master-JavaServer-Seite ein Tag vorhanden. Das nachstehende Beispiel zeigt das Tag, das für die Registerkarte “Leistung” verwendet wird.

```
<!-- The tag that represents the report -->
<ibmspss-taglib:repositoryItem
  name="Performance_Tab_Report_Tag"
  inputURI="<<path of the Performance Report>>"
  repositoryCredentialName="localhost"
  activate="ONLOAD"
  location="Performance_Tab_Report_Output"
  outputType="HTML">
<ibmspss-taglib:sourceLinkPrompt
  targetNameParameter="LeastPerformingScenarios"
  parameterValue="<<localized text>>" />

<!--
<<<< Few other parameters which represent localized text>>>>
-->

<!-- This value comes from the User defined prompts -->
<ibmspss-taglib:sourceLinkPrompt
  promptId="Performance_Tab_NumberOfPerformers_Prompt"
  targetNameParameter="NumberOfPerformers"/>
<!-- The value of this parameter is specified in the tag itself -->
<ibmspss-taglib:sourceLinkPrompt
  targetNameParameter="RunsFromDate"
```

```

        parameterValue="<%=scenariosFrom%>"/>
<ibmspss-taglib:sourceLinkPrompt
    targetNameParameter="RunsToDate"
    parameterValue="<%=scenariosTo%>"/>
</ibmspss-taglib:repositoryItem>

```

Das repositoryItem-Tag

Die Berichtsinformationen werden im Tag `repositoryItem` angegeben.

```

<ibmspss-taglib:repositoryItem
    name="Performance_Tab_Report_Tag"
    inputURI="<<path of the Performance Report>>"
    repositoryCredentialName="localhost"
    activate="ONLOAD"
    location="Performance_Tab_Report_Output"
    outputType="HTML">

```

Das Tag `repositoryItem` hat folgende Attribute:

- Das Attribut *name*, das das Tag “`repositoryItem`” eindeutig kennzeichnet. Die öffentliche API `runRepositoryItem` verwendet diesen Namen, um den Bericht darzustellen. Für die Registerkarte “Leistung” ist der Name *Performance_Tab_Report_Tag*.
- Das Attribut *inputURI*, das den Speicherort des Berichts angibt. Für die Seite “Model Management” werden alle Berichte im Dateisystem des Servers im Verzeichnis *peb-mmd* der IBM® SPSS® Collaboration and Deployment Services Repository-Installation abgerufen. Der angegebene URI muss gültig sein.
- Das Attribut *activate* legt fest, wann der Bericht ausgeführt wird. Für die Registerkarte “Leistung” ist der Wert *ONLOAD*, d. h. der Bericht wird ausgeführt, wenn die Seite geladen wird. Der Wert *ONDEMAND* gibt an, dass der Benutzer dafür verantwortlich ist, die Ausführung des Berichts zu starten, indem er die öffentliche API `runReport` aufruft, die in der Tag-Bibliothek “Reporting” bereitgestellt wird. [Für weitere Informationen siehe Thema Javascript-API auf S. 80.](#)
- Das Attribut *location*, das den Speicherort angibt, an dem der Bericht gerendert werden soll. Dieses Attribut entspricht der *ID* des HTML-Elements, die entweder *DIV* oder *IFRAME* lauten kann. Für Model Management verweist der Berichtspeicherort immer auf *DIV*.
- Das Attribut *outputType*, das das Format angibt, in dem der Bericht mithilfe der IBM® SPSS® Collaboration and Deployment Services Tag Library gerendert werden soll. Es muss ein Ausgabeformat verwendet werden, das von der BIRT Report Designer for IBM® SPSS® Report Engine unterstützt wird. Für die Berichte, die in Model Management verwendet werden, ist der Ausgabetyt immer *HTML*.

Das Tag “sourceLinkPrompt”

Das Tag `sourceLinkPrompt` legt die Verknüpfung von Prompts mit dem Bericht fest. Anders ausgedrückt legt dieses Tag fest, wie der Bericht die Prompt-Werte bei der Ausführung abrufft.

Die Prompt-Werte werden für Model Management auf zwei unterschiedliche Arten angegeben. Zum einen können sie über das Attribut *parameterValue* angegeben werden:

```
<ibmspss-taglib:sourceLinkPrompt
  targetNameParameter="RunsFromDate"
  parameterValue="<%=scenariosFrom%>"/>
```

Hier lautet der Name des Prompts *RunsFromDate* (im Bericht definiert). Der Wert für diesen Prompt wird im Attribut *parameterValue* angegeben. Der Wert, der in diesem Attribut weitergegeben wird, wird direkt an den Bericht weitergegeben.

Des Weiteren können Prompt-Werte angegeben werden, indem ein Benutzer-Prompt mit dem Berichtparameter verknüpft wird. Beispiel:

```
<ibmspss-taglib:sourceLinkPrompt
  promptId="Performance_Tab_NumberOfPerformers_Prompt"
  targetNameParameter="NumberOfPerformers"/>

<input type="hidden" id="Performance_Tab_NumberOfPerformers_Prompt"
  name="Performance_Tab_NumberOfPerformers_Prompt"
  value="<%=userProfile.getPerformanceSize()%>"/>
```

Hier verweist das Attribut *promptId* auf die *ID*, die vom verborgenen HTML-Eingabetaag definiert wird. In diesem Fall würde der Wert, der im ausgeblendeten Feld *Performance_Tab_NumberOfPerformers_Prompt* angegeben ist, bei der Ausführung des Berichts als Prompt-Wert für den Berichtparameter *NumberOfPerformers* weitergegeben.

Anmeldeinformationen

Die Berichte, aus denen die Seite "Model Management" besteht, fragen ihre Inhalte in der Datenbank ab, auf der das Repository beruht. Daher benötigen die Berichte eine Datenquelle, die dieser Datenbank entspricht. Diese Datenbank, *MMDDataSource*, wird im Repository erstellt, wenn der Benutzer die Seite "Model Management" erstmals lädt, und immer verwendet, wenn für eines der Tags Zugriff auf die Repository-Datenbank erforderlich ist.

Um auf die Datenquelle *MMDDataSource* zuzugreifen muss der Bericht gültige Anmeldeinformationen angeben. Das *credential*-Tag auf den JavaServer-Seiten ermöglicht die Angabe dieser Anmeldeinformationen.

```
<ibmspss-taglib:credential
  name="MMDDataSource"
  username="<<ein DB-Benutzername>>"
  password="<<Kennwort für den Benutzer>>"/>
```

Die Anmeldeinformationen für diese Datenquelle werden über *Login.jsp* abgerufen, bevor die Seite angezeigt wird, und entsprechen dem Benutzernamen und dem Passwort für die Datenbank, auf der das Repository basiert. Wenn gültige Anmeldeinformationen angegeben werden, werden sie für die Dauer der Sitzung im Cache gespeichert und für die Ausführung der Berichte verwendet. Die Model Management-Berichte werden so definiert, dass der Name der Datenquelle *MMDDataSource* lautet.

Neben den Anmeldeinformationen für die Datenquelle sind für Model Management-Berichte auch Anmeldeinformationen für den Benutzer, der den Bericht ausführt, erforderlich.

```
<ibmspss-taglib:credential
  name="localhost"
  provider="<< eine Provider-ID >>"
  username="<< Name eines CR-Benutzers >>"
  password="<<Kennwort des Benutzers >>" />
```

Diese Anmeldeinformationen haben den Namen *localhost*. Wenn das Repository mehrere Sicherheits-Provider zulässt, muss das Attribut *provider* angegeben werden.

Das Tag `repositoryItem` erfordert gültige Benutzeranmeldedaten für das Repository, die im Attribut `repositoryCredentialName` des Tags angegeben werden. Für Model Management ist der Wert für dieses Attribut *localhost*. Dieser entspricht dem Benutzernamen, dem Provider und dem Passwort des Benutzers, der sich in IBM® SPSS® Collaboration and Deployment Services Deployment Portal angemeldet hat.

Ausführen von Visualisierungsberichten

Die Methodologie der Ausführung von Visualisierungsberichten ist mit der für BIRT Report Designer for IBM® SPSS®-Berichte identisch. Es sind allerdings folgende Unterschiede bei der Verwendung zu berücksichtigen:

- Visualisierungsberichte verwenden den Wert *ONDEMAND* für das Attribut *activate* des `repositoryItem`-Tags und nicht den Wert *ONLOAD*, der in BIRT Report Designer for IBM SPSS-Berichten verwendet wird.
- Parameter, die für Visualisierungsberichte erforderlich sind, werden von den Master-BIRT Report Designer for IBM SPSS-Berichten weitergegeben. [Für weitere Informationen siehe Thema Interaktivität von Visualisierungsberichten auf S. 81.](#)

Javascript-API

Die Tag-Bibliothek verfügt über ein Framework, das mit JavaScript-Methoden erstellt wurde. Diese JavaScript-Methoden bieten sowohl ein solides Validierungs-Framework als auch ein Handle für den Benutzer zur Ausführung von Berichten auf Abruf.

Um die Berichte nach Bedarf auszuführen, stellt die Tag-Bibliothek eine öffentliche API bereit. Diese öffentliche API ist in der Datei `reportTagLibPublicAPI.js` unter `IBMSPSSTagLib.war` verfügbar. Die JavaScript-Datei enthält folgende API:

```
function runRepositoryItem( reportName, linkData, targetId )
```

Für Model Management wird diese Funktion verwendet, um untergeordnete Berichte für den Master-Bericht aufzurufen.

Wenn die Registerkarte "Szenario" angezeigt wird, werden beispielsweise die Szenarioberichtsdaten angezeigt. Wenn der Benutzer auf den Link für ein Szenario im Master-Bericht klickt, wird die JavaScript-Methode `showDetails` aufgerufen. Diese JavaScript-Methode ist in den BIRT Report Designer for IBM® SPSS®-Bericht eingebettet und ruft indirekt die Methode `runRepositoryItem` für die Ausführung von zwei Berichten auf. Ein Bericht ist der BIRT Report Designer for IBM SPSS-Bericht *Szenario-Details* und der andere ist der Visualisierungsbericht *Szenariovergleich*.

Wenn *linkData* im API-Aufruf null ist, wird der Bericht mit den Daten ausgeführt, die auf der JavaServer-Seite verfügbar sind, die über die unterschiedlichen Tags der IBM® SPSS® Collaboration and Deployment Services Tag Library bereitgestellt werden. Kurz vor dem Aufruf von `runRepositoryItem` speichert der Javascript-Code die Parameterwerte im ausgeblendeten HTML-Steuerelement. Das Tag-Bibliothek-Framework ruft diese Werte ab und gibt sie als Parameter an den ausgeführten Bericht weiter.

Die *targetID*-Felder entsprechen den einzelnen DIV-IDs, in denen der Bericht gespeichert werden soll.

Interaktivität von Visualisierungsberichten

Das Diagramm *Leistung/Szenario*, das vom Visualisierungsbericht für die Registerkarte “Champions” erstellt wird, unterstützt Interaktivität. Wenn der Benutzer auf einen Balken im Diagramm klickt, werden die Details des entsprechenden Szenarios in einem angrenzenden Bereich angezeigt. Die Berichte verwenden für diese Funktion die Tags `actionHandler` und `actionParameter`.

Die Verwendung des Tags `actionHandler` ist für Visualisierungsberichte nicht unbedingt erforderlich. Normalerweise würde das Tag `sourceLinkRepositoryItem` einen ebenso guten Dienst für Visualisierungsberichte erweisen. Auf der Seite “Model Management” kann das Visualisierungsdiagramm jedoch mehrmals angezeigt werden. Die Anwendung benötigt eine spezielle Logik, um Detailzeilen zu erweitern und die Zielberichte mit den jeweiligen Ausgabearten auszuführen. Die `actionHandler`-Tags bieten diese zusätzliche Steuerebene.

Der Abschnitt der Seite, die den Visualisierungsbericht *Leistung/Szenario* erstellt:

```
<ibmspss-taglib:repositoryItem
name="Champions_Scenario_Index_Report"
inputURI="ChampionsScenarioIndex.viz"
repositoryCredentialName="localhost"
activate="ONDEMAND"
outputType="HTML"
location="championsTabVisReport">
<ibmspss-taglib:actionHandler event="onclick" function="selectCCScenario">
  <ibmspss-taglib:actionParameter name="filename" />
  <ibmspss-taglib:actionParameter name="filepath" />
  <ibmspss-taglib:actionParameter name="ccid" />
  <ibmspss-taglib:actionParameter name="equivalencekey" />
</ibmspss-taglib:actionHandler>
</ibmspss-taglib:repositoryItem>
```

Das `repositoryItem`-Tag liefert Details zu dem zu erstellenden Balkendiagramm. Das verschachtelte Tag `actionHandler` gibt an, dass die JavaScript-Funktion `selectCCScenario` aufgerufen werden soll, wenn das Ereignis *onClick* für die Balken eintritt. Die `actionParameter`-Tags, die im Tag `actionHandler` verschachtelt sind, geben an, dass *filename*, *filepath*, *ccid* und *equivalencekey* an die Funktion `selectCCScenario` weitergegeben werden.

Jedes dieser Felder wird im XML des Visualisierungsberichts definiert. Die Definition für die Variable *filename* finden Sie unten:

```
<sourceVariable
categorical="true"
```

```
id="filename"
source="delimitedFileSource_430"
sourceName="ct_filename">
```

Dieses Tag gibt an, dass die Spalte, die im Datenset als *ct_filename* definiert ist, als *filename* in diesem Bericht verwendet wird.

Die JavaScript-Funktion `selectCCScenario` erhält die ID des Berichts, in dem das Ereignis eingetreten ist, und ein Array der Parameterwerte. Intern ruft es `runReport` nach abhängigen untergeordneten Berichten auf und gibt das Werte-Array an sie weiter. [Für weitere Informationen siehe Thema Javascript-API auf S. 80.](#)

URL-Fragmente

Die Seite “Model Management” zeigt einige Repository-Artefakte in einem I-FRAME an. Diese Artefakte sind die Ausgaben, die von bestimmten Job-Ausführungen erzeugt werden.

Ein Artefakt wird geladen, indem die Quelle des I-FRAME auf die URL mit folgendem Format gesetzt wird:

```
http://<servername>:<port>/peb/view?id=<artifact resource id>
```

[Für weitere Informationen siehe Thema URL-Parameter in Kapitel 3 auf S. 6.](#)

Framework “Tab Extension”

Die Navigationsregisterkarten von IBM® SPSS® Collaboration and Deployment Services Deployment Portal können über das Framework “Tab Extension” mit benutzerdefinierten Einträgen erweitert werden. Die Funktion “Model Management” verwendet dieses Framework, um einen Einstiegspunkt in die Seite “Model Management” zu integrieren.

Abbildung 7-2
Model Management-Erweiterung



Deployment Portal liest Erweiterungsdateien, die sich in folgendem Verzeichnis befinden:

```
<Installationsverzeichnis>\components\peb\extensions
```

Diese Dateien werden nach allen Instanzen der *peb-extension*-Elemente durchsucht. Diese Elemente werden separat auf der Benutzeroberfläche angezeigt, wenn die Benutzeranmeldeinformationen erforderliche Aktionen umfassen. Benutzerdefinierte Anwendungen müssen folgende Elemente bereitstellen:

- Erweiterungs-XML-Datei oder einen Eintrag in einem vorhandenen Erweiterungs-XML für die Anwendung
- Geeignete Einträge in der lokalisierten Textdatei (.tx)

Die Funktion “Model Management” ist in der Datei *peb-mmd.package* im Verzeichnis *staging* der Repository-Installation enthalten. Das Paket enthält die Datei *mmd_extension.xml* im Verzeichnis *peb/extensions*. Diese XML-Datei steuert das Erscheinungsbild und die Funktionen der Registerkarte “Model Management”.

```
<file-viewer>
  <peb-extension>
    <tab-id>pebMmdTab</tab-id>
    <tab-key>mmd/pebMmdTabTitle</tab-key>
    <tab-url>
      /peb-mmd/controller?actionName=LoginToMMDAction
    </tab-url>
    <tab-icon>/image2?file=somelcon.gif</tab-icon>
    <tab-position>2</tab-position>
    <tab-security>
      <capability>RunReport</capability>
      <capability>ViewModelManagementDashboard</capability>
    </tab-security>
  </peb-extension>
</file-viewer>
```

Folgende Elemente werden in dieser Datei definiert:

- Das Element *tab-id*, welches die eindeutige ID für die Registerkarte ist. In diesem Fall lautet die ID *pebMmdTab*.
- Das Element *tab-key*, das auf den Text verweist, der auf der neuen Registerkarte angezeigt wird. Zu Lokalisierungszwecken isoliert Model Management lokalisierten Text in XML-Dateien mit der Erweiterung *.tx*. Der Schlüssel kennzeichnet das Element in der Lokalisierungsdatei, das den anzuzeigenden Text enthält. In diesem Fall entspricht der Schlüssel *mmd/pebMmdTabTitle* dem Text *Model Management*.
- Das Element *tab-url* das den URL angibt, der aufgerufen wird, wenn der Benutzer auf die Registerkarte klickt. Der URL kann entweder vollständig qualifiziert (mit einem Schrägstrich ‘/’ beginnen) oder relativ zur Anwendung Deployment Portal angegeben werden. Im letzteren Fall wird *peb* als Kontext angenommen. Der Link muss auf einen gültigen URI verweisen, wobei der angegebene URI-Speicherort in der Verantwortung der benutzerdefinierten Anwendung liegt. Für Model Management enthält der Link einen Verweis auf die war-Datei *peb-mmd.war*.
- Das Element *tab-security*, das die Aktionen festlegt, die für den Zugriff auf die Registerkarte erforderlich sind. Wenn der aktuelle Benutzer diese Aktionen nicht ausführt, wird die Registerkarte nicht auf der Header-JSP angezeigt. Für Model Management sind die Aktionen *RunReport* und *ViewModelManagementDashboard* erforderlich.

Erstellen von benutzerdefinierten Datenservicetreibern

Die IBM® SPSS® Collaboration and Deployment Services-Datenservice-API umfasst öffentliche Java-Oberflächen zum Implementieren benutzerdefinierter Treiber, um auf nichtstandardmäßige Datenquellen zugreifen zu können. Ein benutzerdefinierter Treiber kann z. B. erforderlich sein, wenn aus Sicherheitsgründen kein JDBC- oder ODBC-Zugriff auf Datenbankumgebungen für die Produktion zulässig ist. Ein solcher Treiber kann auch für dateibasierte Datenquellen, Daten im Cache oder ältere Datenquellen erforderlich sein, die von JDBC bzw. ODBC nicht unterstützt werden. Diese Funktion wird überwiegend vom Scoring-Service verwendet.

In diesem Anhang finden Sie einen Überblick über die Datenservice-API und Anweisungen zum Erstellen eines benutzerdefinierten Treibers sowie ein Beispiel für einen benutzerdefinierten Treiber, der im Umfang von IBM SPSS Collaboration and Deployment Services enthalten ist.

Datenservice-API

Die API zum Implementieren von benutzerdefinierten Treibern ist im Paket *com.spss.data.service.datasource* als eine Reihe von Oberflächen enthalten:

- *com.spss.data.service.datasource.DataService*
- *com.spss.data.service.datasource.DataServiceSession*
- *com.spss.data.service.datasource.DataServiceRequest*

Die Paketklassen befinden sich in *data-service.jar* unter `<Repository-Installationsverzeichnis>/staging/scoring.package`.

Initialisierung

DataService-Objekte werden zum Initialisieren von mindestens einer Verbindungsinstanz mit den Parametern verwendet, die für die Datenquelle definiert wurden, z. B. Name und Eigenschaften der Datenquelle. Bei der Klasse, die zum Implementieren der *DataService*-Oberfläche verwendet wird, muss der Standardkonstruktor verwendet werden.

Verbindungsinstanzen werden über mehrere Ereignisse im Cache abgelegt. Ressourcenintensive Initialisierungen und das Ablegen von Ressourcen im Cache können durchgeführt werden, da die Anzahl instanziiertter Verbindungen in der Regel klein ist. Für die gleiche Datenquelle können mehrere Instanzen vorliegen. Es empfiehlt sich, Systemressourcen mit der *terminate*-Methode zu veröffentlichen. Eine einzelne Verbindungsinstanz muss in der Lage sein, gleichzeitig mehrere Aufrufe verarbeiten zu können. Bei jedem Aufruf wird ein eigenes *DataServiceRequest*-Objekt erstellt, das für ein einzelnes Abrufen von Daten verwendet werden kann. Dieses Objekt kann nur einmal abgefragt werden. Die Datenserviceverbindung muss die Abfrageergebnisse nicht

im Cache ablegen. Bei einzelnen Ereignissen wird der DataService mit einer bestimmten Mindesthäufigkeit aufgerufen. Beachten Sie, dass zirkuläre Links dazu führen können, dass Tabellen mehrmals abgefragt werden. Die DataServiceverbindung sollte Fehler mittels *DataServiceException* melden. Andere Ausnahmen sind nicht zulässig.

Datenverarbeitung

Das *DataServiceSession*-Objekt wird während der Verarbeitung eines Ereignisses von *DataService* instanziiert und das Objekt kann dazu verwendet werden, um einen ereignisbasierten Status der Sitzung beizubehalten. Das Sitzungsobjekt instanziiert dann mit der *createRequest*-Methode ein *DataServiceRequest*-Objekt für jede Abfrage mit den folgenden Parametern:

- dem Namen der Tabelle;
- den Namen der Attribute (Spalten), für die die Werte zurückgegeben werden sollen;
- der Liste der Schlüsselattributnamen.
- Schlüsseltupeln, die als eine Reihe von Objektlisten angegeben werden. Die Reihenfolge der Objekte in den Listen stimmt mit der Reihenfolge der Attribute überein, die in den Schlüsselattributnamen angegeben sind.

Mit *DataServiceRequest* wird eine Liste der Datensätze zurückgegeben. Jeder Datensatz umfasst eine Liste mit Attributwerten. Die Liste der Attributwerte muss mit der Liste der Attributnamen übereinstimmen, die an die *createRequest*-Methode übergeben wurde. Attributwerte werden als Java-Objektinstanzen zurückgegeben. Für jeden zurückgegebenen Wert müssen die mit Attributen übereinstimmenden Wertesätze, die Bestandteil des Schlüssels sind, mit einem der Schlüsseltupel übereinstimmen. Im Folgenden werden die Zuordnungen zwischen IBM® SPSS® Collaboration and Deployment Services Enterprise View- und Java-Typen aufgeführt:

IBM SPSS Collaboration and Deployment Services Enterprise View-Datentyp	Erforderliche Java-Klasse
Ganzzahl	java.lang.Integer
Lang	java.lang.Long
Double	java.lang.Double
Gleitkomma	java.lang.Float
Zeichenfolge	java.lang.String
Datum	java.sql.Date
Tageszeit	java.sql.Time
Zeitstempel	java.sql.Timestamp
Dezimal	java.math.BigDecimal
Boolesch	java.lang.Boolean

Um ein *Null*-Objekt zurückzugeben, wenn gilt: `value.isNil() = true`, muss das NULL-Objekt zurückgegeben werden.

Erstellen eines benutzerdefinierten Treibers

So erstellen Sie einen benutzerdefinierten Treiber mit der Datenservice-API:

- ▶ Extrahieren Sie *data-server.jar* aus *scoring.package* und stellen Sie sicher, dass sich das Objekt im Klassenpfad befindet.
- ▶ Schreiben Sie den Java-Quellcode, mit dem die Datenserviceoberflächen implementiert werden, und kompilieren Sie die Treiberklassen.
- ▶ Erstellen Sie das Treiberpaket (als JAR-Datei).
- ▶ Stellen Sie das Paket im Verzeichnis `<Repository-Installationsverzeichnis>/components/data-provider` der IBM® SPSS® Collaboration and Deployment Services-Installation bereit. Informationen zu geclusterten Installationen finden Sie im Abschnitt zur Clusterbildung der Installations- und Konfigurationsdokumentation.
- ▶ Starten Sie je nach Anwendungsserver IBM SPSS Collaboration and Deployment Services neu.

Erstellen Sie zum Testen des Treibers eine entsprechende Datenquelle in Deployment Manager, fügen Sie sie zur Enterprise-Ansicht hinzu und erstellen Sie eine Vorschau der Daten.

Beispiel eines benutzerdefinierten Treibers

Die IBM® SPSS® Collaboration and Deployment Services-Installation beinhaltet ein Beispiel eines benutzerdefinierten Treibers. Die Java-Quelldateien für das Beispiel finden Sie unter `<Repository-Installationsverzeichnis>/help/custom_driver`. Dazu gehören:

- *WSDataServiceConnector.java* (implementiert `com.spss.data.service.datasource.DataService`)
- *WSDataServiceRequest.java* (implementiert `com.spss.data.service.datasource.DataServiceRequest`)
- *WSDataServiceSession.java* (implementiert `com.spss.data.service.datasource.DataServiceSession`)

Zu den Beispieldateien gehört auch eine ausführbare *sample.jar*-Datei.

Der Treiber ruft die Daten aus der Textdatei *DATA_1* ab. Dabei wird der Wert des in der Datenservice-Datenquelle angegebenen Schlüssels zugrunde gelegt. Das Dateiformat lautet wie folgt:

```
data_key=1
data_element_0=Dies sind die Daten des ersten Elements
data_element_1=Dies sind die Daten des zweiten Elements
```

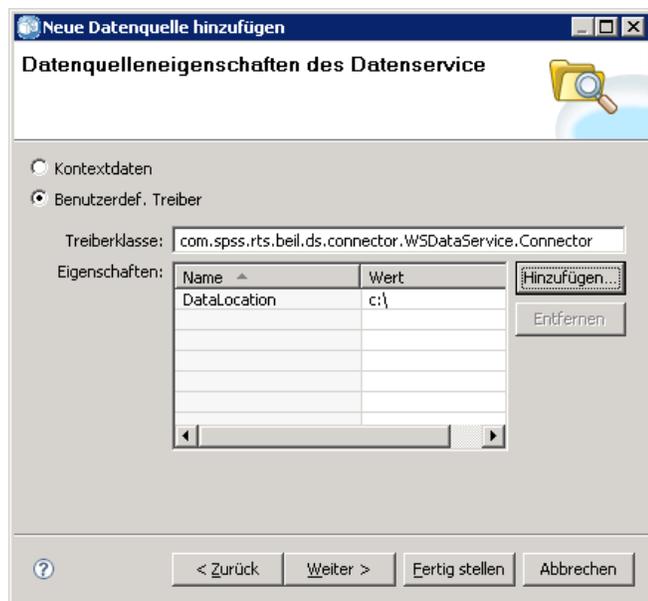
So können Sie die Funktionsfähigkeit des Treibers demonstrieren:

1. Stellen Sie *sample.jar* bereit.

2. Erstellen Sie eine Datenservice-Datenquelle mit folgenden Parametern:
 - **com.spss.rts.beil.ds.connector.WSDataServiceConnector** als Name der Treiberklasse.
 - Treibereigenschaft **DataLocation**, die auf das Verzeichnis verweist, in dem sich die Datendatei *DATA_1* befindet.

Abbildung 8-1

Eigenschaften der Datenservice-Datenquelle unter Verwendung des Beispieldreibers



3. Definieren Sie eine Tabelle für die Datenquelle mit folgenden Spalten:
 - **data_element_0** (Zeichenfolge)
 - **data_element_1** (Zeichenfolge)
 - **data_key** (Zeichenfolge, muss als primärer Schlüssel gekennzeichnet sein)

Abbildung 8-2

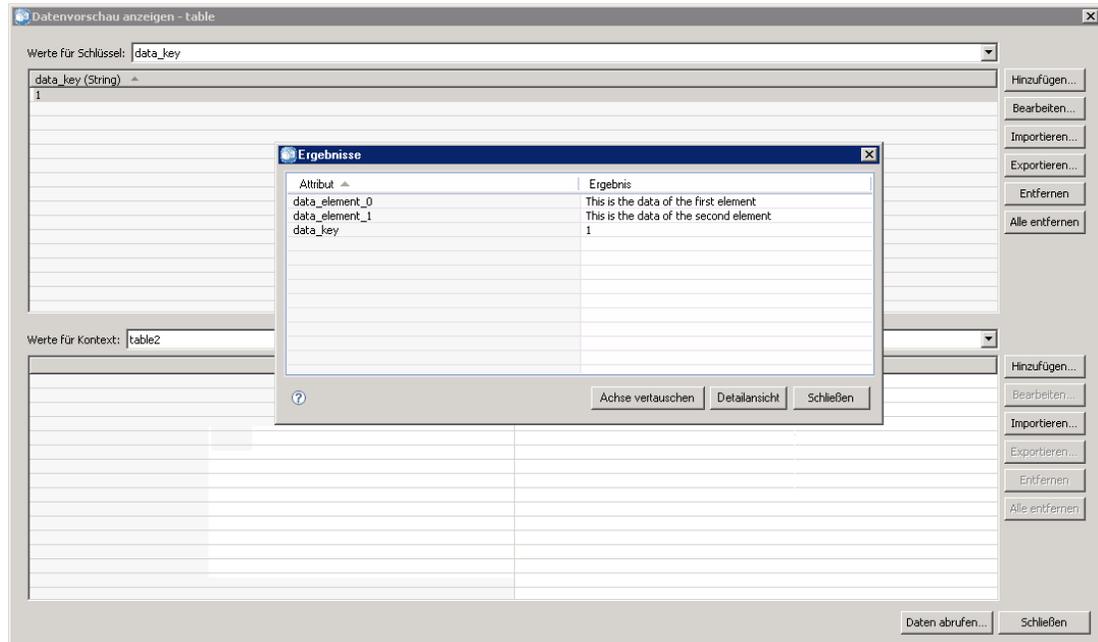
Datenservice-Datenquelle – Tabelleneigenschaften



7. Zeigen Sie eine Vorschau der Daten in der Tabelle an. Geben Sie dabei 1 als Schlüsselwert an. Auf dem Ergebnisbildschirm werden die Inhalte der dem Schlüsselwert entsprechenden Datendatei angezeigt.

Abbildung 8-5

Vorschau der Daten in der Datenquelle für benutzerdefinierte Treiber



Hinweise

Diese Informationen wurden für weltweit angebotene Produkte und Dienstleistungen erarbeitet.

IBM bietet die in diesem Dokument behandelten Produkte, Dienstleistungen oder Merkmale möglicherweise nicht in anderen Ländern an. Informationen zu den in derzeit in Ihrem Land erhältlichen Produkten und Dienstleistungen erhalten Sie bei Ihrem zuständigen IBM-Mitarbeiter vor Ort. Mit etwaigen Verweisen auf Produkte, Programme oder Dienste von IBM soll nicht behauptet oder impliziert werden, dass nur das betreffende Produkt oder Programm bzw. der betreffende Dienst von IBM verwendet werden kann. Stattdessen können alle funktional gleichwertigen Produkte, Programme oder Dienste verwendet werden, die keine geistigen Eigentumsrechte von IBM verletzen. Es obliegt jedoch der Verantwortung des Benutzers, die Funktionsweise von Produkten, Programmen oder Diensten von Drittanbietern zu bewerten und zu überprüfen.

IBM verfügt möglicherweise über Patente oder hat Patentanträge gestellt, die sich auf in diesem Dokument beschriebene Inhalte beziehen. Durch die Bereitstellung dieses Dokuments werden Ihnen keinerlei Lizenzen an diesen Patenten gewährt. Lizenzanfragen können schriftlich an folgende Adresse gesendet werden:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

Bei Lizenzanfragen in Bezug auf DBCS-Daten (Double-Byte Character Set) wenden Sie sich an die für geistiges Eigentum zuständige Abteilung von IBM in Ihrem Land. Schriftliche Anfragen können Sie auch an folgende Adresse senden:

Intellectual Property Licensing, Legal and Intellectual Property Law, IBM Japan Ltd., 1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502 Japan.

Der folgende Absatz gilt nicht für Großbritannien oder andere Länder, in denen derartige Bestimmungen nicht mit dem dort geltenden Recht vereinbar sind. INTERNATIONAL BUSINESS MACHINES ÜBERNIMMT FÜR DIE VORLIEGENDE DOKUMENTATION KEINERLEI GEWÄHRLEISTUNG IRGENDWELCHER ART, WEDER AUSDRÜCKLICH NOCH STILLSCHWEIGEND, EINSCHLIESSLICH (JEDOCH NICHT DARAUF BEGRENZT) DER STILLSCHWEIGENDEN GEWÄHRLEISTUNGEN IN BEZUG AUF DIE NICHTVERLETZUNG VON RECHTEN DRITTER, AUF HANDELSÜBLICHKEIT ODER DIE EIGNUNG FÜR EINEN BESTIMMTEN ZWECK. Einige Staaten lassen bei bestimmten Transaktionen keine Ausschlussklauseln ausdrücklicher oder stillschweigender Gewährleistungen zu, sodass diese Erklärung möglicherweise nicht auf Sie zutrifft.

Diese Informationen können technische Ungenauigkeiten oder typografische Fehler enthalten. An den hierin enthaltenen Informationen werden in regelmäßigen Abständen Änderungen vorgenommen, die in spätere Ausgaben der Publikation eingearbeitet werden. IBM kann jederzeit ohne Vorankündigung Verbesserungen und/oder Veränderungen an den in dieser Publikation beschriebenen Produkten und/oder Programmen vornehmen.

Alle in diesen Ausführungen enthaltenen Verweise auf Websites, die nicht zu IBM gehören, dienen lediglich der Information. Die Nennung bedeutet nicht, dass IBM den Inhalt dieser Websites unterstützt. Das Material auf diesen Websites ist kein Bestandteil des Materials für dieses IBM-Produkt. Sie verwenden diese Websites auf eigene Gefahr.

IBM ist berechtigt, die von Ihnen bereitgestellten Informationen in jeglicher Form zu verwenden bzw. weiterzugeben, die dem Unternehmen geeignet erscheint, ohne dass ihm daraus Verbindlichkeiten Ihnen gegenüber entstehen.

Lizenznehmer dieses Programms, die Informationen dazu benötigen, wie (i) der Austausch von Informationen zwischen unabhängig erstellten Programmen und anderen Programmen und (ii) die gegenseitige Verwendung dieser ausgetauschten Informationen ermöglicht wird, wenden sich an:

IBM Software Group, Attention: Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.

Diese Informationen sind je nach den entsprechenden Geschäftsbedingungen und in manchen Fällen gegen Zahlung einer Gebühr erhältlich.

Das in diesem Dokument beschriebene lizenzierte Programm und sämtliche dafür verfügbaren lizenzierten Materialien werden von IBM gemäß dem IBM-Kundenvertrag, den Internationalen Nutzungsbedingungen für Programmpakete der IBM oder einer anderen zwischen uns getroffenen Vereinbarung bereitgestellt.

Alle in diesem Dokument enthaltenen Leistungsdaten wurden in einer kontrollierten Umgebung ermittelt. Daher können die unter anderen Betriebsumgebungen erzielten Ergebnisse erheblich abweichen. Einige Messungen wurden möglicherweise an Systemen im Entwicklungsstadium vorgenommen und es besteht keine Garantie, dass spätere allgemein verfügbare Systeme dieselben Messwerte aufweisen. Außerdem wurden einige Messwerte möglicherweise mittels Extrapolation geschätzt. Die tatsächlichen Ergebnisse können abweichen. Die Benutzer dieses Dokuments sollten die entsprechenden Daten für ihre jeweilige Umgebung überprüfen.

Informationen zu Nicht-IBM-Produkten stammen von den Herstellern dieser Produkte, ihren veröffentlichten Verlautbarungen oder aus anderen öffentlich verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher die Richtigkeit der Angaben zu Leistung und Kompatibilität oder anderer Behauptungen in Bezug auf Nicht-IBM-Produkte nicht bestätigen. Fragen zu den Fähigkeiten von Nicht-IBM-Produkten sind an die Hersteller dieser Produkte zu richten.

Alle Aussagen in Bezug auf die zukünftige Ausrichtung oder die zukünftigen Vorhaben von IBM können ohne Vorankündigung geändert oder widerrufen werden und stellen lediglich Zielsetzungen dar.

Diese Informationen enthalten Beispiele für Daten und Berichte, die in alltäglichen Betriebsabläufen verwendet werden. Um sie möglichst umfassend darzulegen, enthalten die Beispiele Namen von Einzelpersonen, Unternehmen, Marken und Produkten. Alle diese Namen sind frei erfunden und jegliche Ähnlichkeit mit den von einem tatsächlichen Handelsunternehmen verwendeten Namen und Adressen ist rein zufällig.

Bei der Anzeige dieser digitalen Informationsversion sind die Fotografien und Farbillustrationen möglicherweise nicht sichtbar.

Trademarks

IBM, das IBM-Logo, ibm.com und SPSS sind Marken von IBM Corporation, die in vielen Ländern weltweit eingetragen sind. Eine aktuelle Liste der IBM-Marken finden Sie im Internet unter <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, das Adobe-Logo, PostScript und das PostScript-Logo sind eingetragene Marken oder Marken von Adobe Systems Incorporated in den USA und/oder anderen Ländern.

Intel, das Intel-Logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken von Intel Corporation oder seinen Tochtergesellschaften in den USA und anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken von Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Java und alle Java-basierten Marken und Logos sind Marken von Sun Microsystems, Inc. in den USA und/oder anderen Ländern.

Weitere Produkt- oder Servicenamen können Marken von IBM oder anderen Unternehmen sein.



- actionHandler-Tag, 58, 81
- actionParameter-Tag, 59, 81
- activate-Attribut
 - report-Tag, 78
 - repositoryItem-Tag, 46
- Aktionen, 1
- Anmeldeinformationen, 1
- Anpassen
 - Deployment Portal, 3
- Attribut "Name"
 - actionParameter-Tag, 60
 - credential-Tag, 43
 - repositoryItem-Tag, 45, 78
- Ausgabe
 - für benutzerdefinierte Dialogfelder, 34
 - für Berichte, 32
 - für Jobs, 33
 - für Scoring-Modelle, 34
- Authentifizierung, 4

- Benutzerdefinierte Dialogfelder, 1, 34
- Benutzerdefinierte Dialogfelder in IBM SPSS Statistics, 1
- Benutzerdefinierter Treiber, 84, 86
 - Beispiel, 86
- Berichte, 32
- BIRT-Berichte
 - JSP-Beispiele, 66

- columnName-Attribut
 - sourceLinkVariable-Tag, 58
- contentType-Attribut
 - der page-Aufbauregel, 35
- Cookies, 73
- Credential-Bean, 61
- credential-Tag, 42, 79
- credentialDefinitionName-Attribut
 - credential-Tag, 43

- dataset_table, Parameter
 - in URL-Abfragen, 17
- Datenquelle, 84, 86
- Datenservice, 84
- Datenservice-API, 84, 86
- Datenservice-Datenquelle, 86
- Deployment Portal
 - Anpassen, 3
- Der Parameter "version"
 - in URL-Abfragen, 8

- Echtzeit-DPD, 86
- Einzelanmeldung, 72
- emf-Dateien, 11
- erstellen
 - HTMLC-Dateien, 75
 - event-Attribut
 - actionHandler-Tag, 59
 - Excel-Dateien, 11

 - function-Attribut
 - actionHandler-Tag, 59
 - Funktion "getBookmarkedValues", 40
 - Funktion "retrievePromptValues", 40
 - Funktion "runRepositoryItem", 39

 - getValueJSFunction-Attribut
 - sourceLinkPrompt-Tag, 54

 - height-Attribut
 - repositoryItem-Tag, 49
 - HTMLC-Dateien, 74
 - erstellen, 75
 - Struktur, 74

 - IBM SPSS Statistics-Data File Driver Service, 1
 - IBM SPSS Statistics-Server, 1
 - inputURI-Attribut
 - repositoryItem-Tag, 46

 - Java-Oberflächen, 84, 86
 - Jobs, 33
 - jpeg-Dateien, 11
 - JSP-Beispiele
 - Zugriff, 66
 - JSR 168, 67

 - Klassenlader
 - für benutzerdefinierte Anwendungen, 38
 - Reihenfolge, 38
 - Richtlinie, 38

 - language-Attribut
 - der page-Aufbauregel, 35
 - legal notices, 90
 - linkType-Attribut
 - sourceLinkRepositoryItem-Tag, 57
 - localhost-Anmeldeinformationen, 79
 - location-Attribut
 - outputLocation-Tag, 52
 - report-Tag, 78
 - repositoryItem-Tag, 47
 - Lotus-Dateien, 11

 - MMDDataSource, 79

 - outputId-Attribut
 - outputLocation-Tag, 51

- outputLocation-Tag, 51
- outputType-Attribut
 - report-Tag, 78
 - repositoryItem-Tag, 48

- page-Aufbauregel, 35
- Parameter “allowDownload”
 - in URL-Abfragen, 25
- Parameter “allowPivoting”
 - in URL-Abfragen, 25
- Parameter “allowPrinterFriendly”
 - in URL-Abfragen, 25
- Parameter “batch_type”
 - in URL-Abfragen, 18
- Parameter “dataset”
 - in URL-Abfragen, 16
- Parameter “dataset_label”
 - in URL-Abfragen, 17
- Parameter “dataset.prompt”
 - in URL-Abfragen, 20
- Parameter “dataset_rowlimit”
 - in URL-Abfragen, 18
- Parameter “dataset.search.criteria”
 - in URL-Abfragen, 20
- Parameter “dataset.table”
 - in URL-Abfragen, 20
- Parameter “dataset.uri”
 - in URL-Abfragen, 19
- Parameter “dbcredential_datasourcename”
 - in URL-Abfragen, 13
- Parameter “dbpwd_datasourcename”
 - in URL-Abfragen, 14
- Parameter “dbuser_datasourcename”
 - in URL-Abfragen, 14
- Parameter “format”
 - in URL-Abfragen, 12
- Parameter “fragment”
 - in URL-Abfragen, 12
- Parameter “height”
 - in URL-Abfragen, 15
- Parameter “id”
 - in URL-Abfragen, 7
- Parameter “javascript.name”
 - in URL-Abfragen, 23
- Parameter “javascript.url”
 - in URL-Abfragen, 22
- Parameter “output.filename”
 - in URL-Abfragen, 24
- Parameter “output.format”
 - in URL-Abfragen, 24
- Parameter “outputtype”
 - in URL-Abfragen, 10
- Parameter “partId”
 - in URL-Abfragen, 10
- Parameter “promptstate”
 - in URL-Abfragen, 9
- Parameter “provider”
 - in URL-Abfragen, 9
- Parameter “scoring_configuration”
 - in URL-Abfragen, 18
- Parameter “showLogs”
 - in URL-Abfragen, 25
- Parameter “showOutline”
 - in URL-Abfragen, 24
- Parameter “statistics.server”
 - in URL-Abfragen, 26
- Parameter “statistics.server.credential”
 - in URL-Abfragen, 26
- Parameter “stylesheet.name”
 - in URL-Abfragen, 22
- Parameter “stylesheet.url”
 - in URL-Abfragen, 22
- Parameter “username”
 - in URL-Abfragen, 8
- Parameter “validate.method”
 - in URL-Abfragen, 23
- Parameter “var_variable”
 - in URL-Abfragen, 15
- Parameter “variable.display”
 - in URL-Abfragen, 21
- Parameter “variable.sort”
 - in URL-Abfragen, 21
- Parameter “waitstate”
 - in URL-Abfragen, 10
- Parameter “width”
 - in URL-Abfragen, 14
- parameterName-Attribut
 - repositoryItemPrompt-Tag, 50
- parameterValue-Attribut
 - sourceLinkPrompt-Tag, 54, 79
- partId-Attribut
 - actionHandler-Tag, 59
 - outputLocation-Tag, 52
- password, Parameter
 - in URL-Abfragen, 8
- password-Attribut
 - credential-Tag, 44
- PDF-Dateien, 12
- PevMetaDataBean-Bean, 63
- png-Dateien, 11
- Portal, 67
 - Einzelanmeldung, 72
- Portlet, 67
- Portlet-Pakete, 67
- postscript-Dateien, 11
- PowerPoint-Dateien, 11
- prefix-Attribut
 - der taglib-Aufbauregel, 35
- promptId-Attribut
 - repositoryItemPrompt-Tag, 50
 - sourceLinkPrompt-Tag, 53, 79
- Prompts
 - für benutzerdefinierte Dialogfelder, 34

- für Berichte, 33
- für Jobs, 33
- für Scoring-Modelle, 34
- provider-Attribut
 - credential-Tag, 44
- report-Tag, 51
- ReportBean-Bean, 61
- reportDefinitionURI-Attribut
 - report-Tag, 78
- reportPrompt-Tag, 51
- Repository-Elemente, 32
 - Benutzerdefinierte Dialogfelder, 34
 - Berichte, 32
 - Jobs, 33
 - Scoring-Modelle, 33
- repositoryCredentialName-Attribut
 - repositoryItem-Tag, 47, 80
- repositoryItem-Tag, 44, 78
- repositoryItemName-Attribut
 - repositoryItemPrompt-Tag, 50
- repositoryItemPrompt-Tag, 50
- runRepositoryItem, 80
- Safari-Browser, 73
- Scoring-Modelle, 33
- ScoringBean-Bean, 64
- SearchBean bean, 62
- session-Attribut
 - der page-Aufbauregel, 35
- showNavigationBar-Attribut
 - repositoryItem-Tag, 48
- showTitle-Attribut
 - repositoryItem-Tag, 48
- showToolBar-Attribut
 - repositoryItem-Tag, 48
- sourceLinkPrompt-Tag, 53, 78
- sourceLinkReport-Tag, 57
- sourceLinkRepositoryItem-Tag, 56
- sourceLinkVariable-Tag, 57
- sourceName-Attribut
 - sourceLinkRepositoryItem-Tag, 57
- taglib-Aufbauregel, 35
- targetNameParameter-Attribut
 - sourceLinkPrompt-Tag, 53
 - sourceLinkVariable-Tag, 58
- title-Attribut
 - repositoryItem-Tag, 48
- trademarks, 92
- uri-Attribut
 - der taglib-Aufbauregel, 35
- URL-Parameter
 - Beispiel, 82
- username-Attribut
 - credential-Tag, 44
- useSSO-Attribut
 - credential-Tag, 43
- validateJSFunction-Attribut
 - sourceLinkPrompt-Tag, 54
- Variablenparameter
 - in URL-Abfragen, 15
- Visualisierungsberichte
 - Interaktivität, 81
 - JSP-Beispiele, 66
- war-Datei, 35
- Web Part, 67
- WebSphere, 38
- width-Attribut
 - repositoryItem-Tag, 49
- Word-Dateien, 11