IBM SPSS Collaboration and Deployment Services 4.2 Customization Reference

IBM ®

Preface

IBM® SPSS® Collaboration and Deployment Services is an enterprise-level application that enables widespread use and deployment of predictive analytics. IBM SPSS Collaboration and Deployment Services provides centralized, secure, and auditable storage of analytical assets, advanced capabilities for management and control of predictive analytic processes, as well as sophisticated mechanisms of delivering the results of analytical processing to the end users. The benefits of IBM SPSS Collaboration and Deployment Services include safeguarding the value of analytical assets, ensuring compliance with regulatory requirements, improving the productivity of analysts, and minimizing the IT costs of managing analytics.

Technical Support

The services of SPSS Inc. Technical Support are available to registered customers of SPSS Inc.. Customers may contact Technical Support for assistance in using SPSS Inc. products or for installation help for one of the supported hardware environments. To reach Technical Support, see the SPSS Inc. Web site at *http://www.spss.com*, or contact your local office, listed on the SPSS Inc. Web site at *http://www.spss.com/worldwide*. Be prepared to identify yourself, your organization, and the serial number of your system.

Tell us your thoughts

Your comments are important. Please let us know about your experiences with SPSS Inc. products. Please send e-mail to *suggest@us.ibm.com*, or write to SPSS Inc., Attn: Director of Product Planning, 233 South Wacker Drive, 11th Floor, Chicago IL 60606-6412.

Contents

5 Portal integration   60

6 HTML archive   66

7 Customization example   68

8   Creating Custom Data Service Drivers   76

Appendix

A   Notices   83

Index   85

Customization overview

IBM® SPSS® Collaboration and Deployment Services Deployment Portal serves as a thin-client interface into the IBM® SPSS® Collaboration and Deployment Services Repository, allowing any user with a browser and valid credentials to work with content stored within the repository. However, the default appearance and functionality may not be optimal for all users. For example, you may want to modify the appearance of the browser interface to better match a corporate standard. Alternatively, you may wish to create your own interface to repository content.

IBM® SPSS® Collaboration and Deployment Services offers a variety of techniques for customizing the interaction with content stored in the repository, including the following:

- Modifying package components, such as images and stylesheets, to control the Deployment Portal appearance. For more information, see *IBM SPSS Collaboration and Deployment Services Deployment Portal customization*.

- Referencing repository content directly using uniform resource locators (URL) parameters. For more information, see *URL parameters*.

- Creating custom web pages based on information obtained from reports and queries stored in the repository using Java Server Page tags. For more information, see *IBM SPSS Collaboration and Deployment Services Tag Library*.

- Embedding repository content, such as reports, on portal pages. For more information, see *Portal integration*.

- Performing batch processing of repository content using Python scripting. For more information, see the IBM® SPSS® Collaboration and Deployment Services - Essentials for Python documentation

Prerequisites

For proper processing of custom dialogs, the following requirements must be satisfied:

- A remote execution server for IBM® SPSS® Statistics must be set up in IBM® SPSS® Collaboration and Deployment Services Deployment Manager and then designated as the default server for executing custom dialog syntax using the browser-based IBM® SPSS® Collaboration and Deployment Services Deployment Manager. It is also possible to configure individual custom dialogs to use a specific SPSS Statistics server different from the system default.

- The user must be assigned the *Run Custom Dialogs* action to be able to execute custom dialogs.

- SPSS Statistics save file access is enabled by the SPSS Statistics Data File Driver Service, which must be installed, started, and then designated as the driver for SPSS Statistics data using the browser-based Deployment Manager. The software is available as a download to SPSS Inc. customers.

Important! The SPSS Statistics Data File Driver Service must run on a host with the same operating system type as the repository host. For example, it is impossible to use a repository running on a Linux server in conjunction with the Data File Driver Service running on a Windows server. For information about IBM® SPSS® Collaboration and Deployment Services system configuration and actions, see the administrator's documentation.

IBM SPSS Collaboration and Deployment Services Deployment Portal customization

Administrators can customize certain elements of the IBM® SPSS® Collaboration and Deployment Services Deployment Portal user interface by modifying various files in a repository package (*peb-webcontent.package*) and redeploying the package with the IBM® SPSS® Collaboration and Deployment Services Package Manager. Experience with stylesheets (*.css*) is recommended.

In addition, the system can be configured to use a custom authentication mechanism, eliminating the need to manually enter credentials when accessing the system using a browser.

Customizing the user interface

To customize the IBM® SPSS® Collaboration and Deployment Services Deployment Portal user interface:

1. In the *staging* directory of your repository installation, copy the file *peb-webcontent.package* and store it in a separate directory. Use this file as a back up if you want to revert to the default user interface in the future.

2. In the *staging* directory of your repository installation, open *peb-webcontent.package* with a file compression utility and extract its contents to a temporary directory. The extracted content will contain an *images* directory and a *css* directory that contain most of the graphics and style sheets used.

3. In the extracted *images* directory, modify or replace any of the graphics as desired. To modify the appearance of pages, change the contents of the *spssStyles.css* file using a simple text editor. To assist in changing the appearance of these elements, consider the following suggestions:

   ■ To determine which image or style to modify, load the web page in a browser and use the *View Source* functionality. The HTML source will identify the images and styles used.

   ■ Some images may be specified by rules in the *css* file.

   ■ To suppress the product name in the banner, specify a font-size of 0pt.

   ■ To suppress images, replace them with transparent images.

4. In the extracted *config* directory, open the *UIConfig.xml* file in a text editor. Modify settings to suppress certain elements of the user interface as desired. When finished, save and close the file. Common elements to suppress include the following:

   - `Footer`: The bottom footer bar containing the "Powered by" logo (suppressed by default). Change value from `false` to `true` to display.

   - `FileHeader`: The gray file information bar containing the file name and date/time last modified, displayed when viewing a document.Change value from `true` to `false` to suppress.

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

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

5. Using a file compression utility, compress all folders and customized files that were extracted previously. Save the file as *peb-webcontent.package*. Be sure to use a *.package* file extension, not *.zip*.

6. Use IBM® SPSS® Collaboration and Deployment Services Package Manager to install the customized *peb-webcontent.package* file you saved. For instructions, see the IBM SPSS Collaboration and Deployment Services Package Manager documentation in the IBM® SPSS® Collaboration and Deployment Services Repository Installation and Configuration Guide. When finished, log out and close the utility.

7. Restart the repository server.

8. Wait several minutes and open Deployment Portal (*http://<hostname>:<port>/peb*) to verify your changes.

Authentication customization

If single sign-on is configured and enabled at your site, manual entry of security credentials is not necessary for accessing IBM® SPSS® Collaboration and Deployment Services Deployment Portal. However, if single sign-on is not enabled, the system can still be customized to avoid manual credential entry.

The architecture includes a Java interface named AuthenticationCriteriaInterface. This interface includes the following methods:

```
public String getUserName()
public String getPassword()
public String getProvider()
```

To customize authentication, you must first create a Java class that implements this interface, such as *com.spss.AuthenticationCriteriaImpl*. Your class must provide the username, password, and provider authentication information. These values may be supplied by a variety of sources, such as a previously authenticated intranet site or portal. The class should be placed in the classpath for the application server.

Next, configure the system to use your class. In the browser-based IBM® SPSS® Collaboration and Deployment Services Deployment Manager, modify the following Deployment Portal configuration settings:

- *Configured Authentication Criteria Class.* Supply the name of your custom class.
- *Use Configured Authentication Criteria.* Select this option to enable the use of your class.

After configuring you authentication class, users should be able to enter the Deployment Portal without entering credentials, provided that the credentials supplied by the custom class are valid. In addition, URL references to repository items will not need to include credential information.

URL parameters

You can access IBM® SPSS® Collaboration and Deployment Services Deployment Portal reports and other repository objects using direct URLs (Uniform Resource Locators). With URLs, you can directly share reporting information in different ways such as embedding reporting into your external web sites and applications. This reference document lists various URL parameters and contains some tips for building and using Deployment Portal URL query strings. For assistance, contact Technical Support.

The URL parameters outlined in this document are unrelated to the URLs available in IBM® SPSS® Collaboration and Deployment Services Deployment Manager.

URL base path

The base path for all requests is:

http://*<hostname>*:*<port>*/peb/view/*<content repository path>*

or

http://*<hostname>*:*<port>*/peb/view?id=*<object-id>*

where:

- *<hostname>* is the name or IP address of the repository server
- *<port>* is the port number on which to connect to the repository
- *<content repository path>* is the resource path of the repository object on which to act
- *<object-id>* is the resource ID of the repository object on which to act

Examples

```
http://yourserver:8080/peb/view/sample/employee.dbq
http://yourserver:8080/peb/view?id=0a58c3461e885d240000010f4cc607188375
```

Query string

The base path for the URL reference can be followed by a query string containing parameters that provide additional processing information. The query string begins with a question mark and contains parameter/value pairs separated by ampersands (&). Note that if a repository item is referenced by its resource identifier, the question mark initiating the query string is already present for the id parameter and should not be repeated for any other parameters.

At a minimum, a URL must contain the content repository path in the base path or the `id` parameter. Other parameters are optional. Unless otherwise stated, parameters and their values are case sensitive. Some parameters, such as `username` and `password`, are used in virtually all URL queries, while the use of other parameters may depend on the type of item being referenced in the query. Note that the system can be configured to use a custom authentication mechanism to eliminate the need to supply security credential parameters in the query string. For more information, see the topic Authentication customization in Chapter 2 on p. 4.

Reserved characters like `&` and excluded US-ASCII characters like `#` should be URL encoded before being specifying as a parameter value in the query string. However, characters in the reserved set are not reserved in all contexts. In general, a character is reserved if the semantics of the URI changes if the character is replaced with its escaped US-ASCII encoding. Hence some characters (like `?`, `=`, and `:`) are not reserved in the parameter values, but characters like `&` and `#` are, and hence need to be URL encoded.

For example, the `&` character should be URL encoded as `%26`. Thus, the following URL:

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

should be specified as

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

The following sections describe each parameter.

Common parameters

Common parameters are used in virtually all URL references, or are used across multiple types of repository items.

The id parameter

The `id` parameter specifies the repository identifier for the item on which to act.

Syntax

```
id=<identifier>
```

The value of *<identifier>* corresponds to the repository object identifier.

Examples

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

The version parameter

The `version` parameter specifies the version of the repository object on which to act using the version marker or the version label. Special characters, such as spaces, must be escaped. Omit this parameter to display the LATEST version.

Syntax

```
version=m.<version marker>
version=l.<label>
```

The value of *<version marker>* corresponds to the version of the repository object. Alternatively, the value of *<label>* designates the version label of the repository object.

Examples

```
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
```

The username parameter

The `username` parameter specifies the user with which to log in to the system.

Syntax

```
username=<user_ID>
```

```
username=<user_ID>
```

The value of *<user_ID>* is the user identifier of the person logging in to the repository server.

Example

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

The password parameter

The `password` parameter specifies the password with which to log in to the system.

Syntax

```
password=<password>
```

The value of *<password>* specifies the password of the person logging in to the repository server.

Example

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

The provider parameter

The `provider` parameter specifies the security provider against which to validate credentials. A value for `provider` must be specified if the `username` and `password` parameters are used.

Syntax

```
provider=<provider>
```

The value of *<provider>* specifies the security provider. Valid values include:

- *Native* for the built-in provider
- *AD/<domain>* for Active Directory, where *<domain>* corresponds to the DNS namespace
- *ADL/<domain>* for Active Directory with local override, where *<domain>* corresponds to the DNS namespace
- *iSeries* for IBM i
- *devldapOpenLDAP* for OpenLDAP

Example

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

The promptstate parameter

The `promptstate` parameter specifies whether to suppress the runtime prompt dialog for prompted variable values that are not specified in the query string.

Syntax

promptstate=*<x>*

A value of *1* for *<x>* suppresses the runtime prompt dialog, using the specified default variable value for any prompted variables that are not specified. A value of *2* displays the runtime prompt dialog for any prompted variables that are not specified. Alternately, you can omit this parameter to allow the prompt dialog to be displayed.

Example

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

The waitstate parameter

The `waitstate` parameter specifies whether to suppress the Wait screen while a report is running.

Syntax

waitstate=*<x>*

A value of *1* for *<x>* suppresses the Wait screen. Omit this parameter to display the Wait screen.

Example

```
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
```

The partId parameter

The `partId` parameter identifies a specific part of the repository object being referenced. For HTMLC files, this parameter can reference a specific file within the archive. For IBM® SPSS® Statistics output files (*.spw*), the parameter corresponds to the index as shown in the outline for the file. For example, to get the first part, specify `partId=0`.

Syntax

partId=*<reference_id>*

The value of *<reference_id>* is one of the following values:

- The relative path and name of a file within an HTMLC file
- The index of the desired output within an *.spw* file

Example

```
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
```

The outputtype parameter

The `outputtype` parameter specifies the file type of the result set.

Syntax

outputtype=*<file_type>*

The value of *<file_type>* corresponds to one of the values in the following table.

Table 3-1
Output types

Report Type	Value	Returns
IBM® ShowCase®	*html*	HTML
	pdf	Portable Document Format
	wk4	Lotus 1-2-3
	text	text
	csv	comma separated values
	biff8	Microsoft Excel
	xlsx	Microsoft Excel 2007 XML format
	rptdocument	BIRT Report Designer for IBM® SPSS® report document

Report Type	Value	Returns
BIRT Report Designer for IBM SPSS	*HTML*	HTML
	Excel 97/2003	Microsoft Excel
	PowerPoint	Microsoft PowerPoint
	Word Document	Microsoft Word
	PDF	Portable Document Format
	PDF - Fit to Page Width	Portable Document Format using width magnification
	PDF - Page Break Pagination Only	Portable Document Format using page break pagination
	PDF - Fit to Whole Page	Portable Document Format using page magnification
	PostScript	PostScript format
	PostScript - Fit to Page Width	PostScript format using width magnification
	PostScript - Page Break Pagination Only	PostScript format using page break pagination
	PostScript - Fit to Whole Page	PostScript format using page magnification
	BIRT RPT Document	BIRT Report Designer for IBM SPSS report document
	ask	a prompt for the user at runtime to specify an output format
Visualization	*png*	Portable Network Graphics format
	emf	Enhanced Metafile format
	jpeg	JPEG
	html	HTML. This is a valid output format for visualization reports only when the output is a table. If HTML is specified as the format for a visualization report that does not produce a table, the output is converted to a PNG image.
	pdf	PDF
	ask	a prompt for the user at runtime to specify an output format
Custom dialogs	*SPW*	IBM® SPSS® Statistics web output viewer
	HTML	HTML

Example

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

The format parameter

The `format` parameter specifies whether to return the original file stored to the repository, rather than running the file.

Syntax

```
format=raw
```

The value of *raw* returns the original file. For example, for IBM® ShowCase® Query definitions, using the `format` parameter will download the original *.dbq* file instead of running the query dynamically.

Example

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

The fragment parameter

The `fragment` parameter specifies whether to display the IBM® SPSS® Collaboration and Deployment Services Deployment Portal user interface elements (i.e., header, footer, Content Repository tree) with the report results.

Syntax

```
fragment=true
```

The value of *true* suppresses the Deployment Portal interface elements. Omit this parameter to display the interface.

Example

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

Parameters for variables

For non-report repository items that use variables, such as jobs, the value for a variable can be specified by including the variable name and value in the URL query string. For custom dialogs, jobs, and scoring, variable value prompts will appear for all variables or no variables, depending on the value of the parameter.

For report items, the variable name must be preceded by the `var_` prefix. For more information, see the topic

Syntax

```
<variable>=<value>
```

The value of *<variable>* corresponds to the name of the variable to satisfy. The value of *<value>* is the entry to use to satisfy the specified report variable

Example

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

Report parameters

Report parameters are used in references to reports stored within the IBM® SPSS® Collaboration and Deployment Services Repository. The reports may be visualization reports, BIRT Report Designer for IBM® SPSS® reports, or IBM® ShowCase® reports.

The dbcredential_datasourcename parameter

The `dbcredential_datasourcename` parameter specifies the credential with which to log on to the data source. This is used if the data source user ID differs from the IBM® SPSS® Collaboration and Deployment Services Deployment Portal user ID.

Syntax

dbcredential_*<datasourcename>*=*<credential id>*

The value of *<datasourcename>* is the name of the given data source. The value of *<credential id>* is the identifier of the credential object to be used for connecting to the data source.

Example

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

The dbuser_datasourcename parameter

The `dbuser_datasourcename` parameter specifies the user identifier with which to log on to the data source. This is used if the user identifier for the data source differs from the user identifier for the IBM® SPSS® Collaboration and Deployment Services Deployment Portal.

Syntax

dbuser_*<datasourcename>*=*<user_ID>*

The value of *<datasourcename>* is the name of the given data source. The value of *<user_ID>* is the user identifier of the person connecting to the data source.

Example

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

The dbpwd_datasourcename parameter

The `dbpwd_datasourcename` parameter specifies the password with which to log on to the data source. This is used if the data source user identifier differs from the IBM® SPSS® Collaboration and Deployment Services Deployment Portal user identifer.

Syntax

dbpwd_*<datasourcename>*=*<password>*

The value of *<datasourcename>* is the name of the given data source. The value of *<password>* is the password of the person connecting to the data source.

Example

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

Note

If the `dbcredential_datasourcename` parameter has been specified, that parameter will be considered for connecting to the data source before the `dbuser_datasourcename` and `dbpwd_datasourcename` parameters.

The width parameter

The `width` parameter specifies width of the resulting image or graph. This parameter is used specifically with visualization reports.

For reports containing height and width specifications, both `height` and `width` parameters must be provided. If either parameter is missing, the graph would be rendered with its default height and width.

Syntax

width=*<x>*

The value of *<x>* specifies the integer value for the width in pixels.

Example

```
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
```

The height parameter

The `height` parameter specifies height of the resulting image or graph. This parameter is used specifically with visualization reports.

For reports containing height and width specifications, both `height` and `width` parameters must be provided. If either parameter is missing, the graph would be rendered with its default height and width.

Syntax

height=*<x>*

The value of *<x>* specifies the integer value for height in pixels.

Example

```
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
```

The var_variable parameter

The `var_variable` parameter specifies the value to use to satisfy the specified report variable.

Syntax

var_*<variable>*=*<value>*

The value of *<variable>* is the name of the variable to satisfy. To locate the variable name, in IBM® ShowCase® Query or IBM® ShowCase® Report Writer, from the Query menu, select Variables. A list of variable names for the current report is displayed.

The value of *<value>* is the entry to use to satisfy the specified report variable

Example

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

Notes

- For reports, specifying a variable value on the URL will suppress the runtime prompt for that variable.
- To specify a single variable value (=), use the syntax `var_Lastname=Curtis`
- To specify multiple variable values (`IN`), use the syntax `var_Lastname=Curtis&var_Lastname=McLind`
- To specify a range of variable values (`BETWEEN`), use the syntax `var_Dateship=3-1-2007&var_Dateship=3-31-2007`
- To specify values for multiple variables, use the syntax `var_Lastname=Curtis&var_Dateship=3-1-2007&var_Dateship=3-31-2007`

Scoring parameters

Scoring parameters are used when referencing scoring configurations to generate scores.

The dataset parameter

The `dataset` parameter specifies the location of a SQL data provider definition that will be used for batch scoring. The value of this parameter will be a relative path within the IBM® SPSS® Collaboration and Deployment Services Repository.

Syntax

dataset=*<dpd_location>*

The value of *<dpd_location>* is the repository path to the data provider definition.

Example

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

The dataset_label parameter

The dataset_label parameter allows the user to specify the appropriate version of the dataset. The specified dataset version must be compatible with the data provider defined in the scoring configuration. If not specified, the *LATEST* version is used.

Syntax

dataset_label=*<myLabel>*

The value of *<myLabel>* is the label for the desired dataset version.

Example

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

The dataset_rowlimit parameter

The user may limit the amount of data processed from the dataset for batch scoring. This will help prevent long running processes. The dataset_rowlimit specifies the number of rows of data that will be extracted from the dataset.

Syntax

dataset_rowlimit=*<x>*

The value of *<x>* denotes the number of dataset rows to be extracted.

Example

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

The scoring_configuration parameter

The scoring_configuration parameter specifies the scoring configuration used by the scoring engine to score the specified model.

Syntax

scoring_configuration=*<configName>*

The value of *<configName>* is the name of scoring configuration to use for scoring. The specified configuration must be able to process a scoring request. A reference to a suspended configuration will be unable to produce scores.

Example

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

The batch_type parameter

The `batch_type` parameter specifies which scoring input prompts should be displayed. If the parameter specifies *dataset*, the scoring interface will generate the input prompts for the dataset and label. If the `batch_type` is not specified and parameter inputs are not defined, the interface based on scoring parameters is used.

Syntax

batch_type=*<inputPrompt>*

The value of *<inputPrompt>* indicates the source for the input prompts. Currently, the only supported source is *dataset*. Omit this parameter to prompt the user for input values based on parameters.

Example

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

Custom dialog parameters

Custom dialog parameters are used when referencing custom dialog (*.spd*) files.

Note: This functionality requires IBM® SPSS® Statistics adapters in the IBM® SPSS® Collaboration and Deployment Services environment. For more information, see the SPSS Statistics installation documentation.

The dataset.uri parameter

The URI of the dataset to be use by the custom dialog. For DPDs and *.sav* files in the IBM® SPSS® Collaboration and Deployment Services Repository, the URI can be specified as a repository path or the resource ID. When the URI references a file on the file system, the path to the file must be valid from the IBM® SPSS® Statistics Data File Driver server that is used to retrieve the variable metadata. It must also be a valid path on the SPSS Statistics Server that will execute the syntax. If a repository dataset object is used, the version of the object can be appended to the URI either as a version maker or a label.

Syntax

dataset.uri=*<myURI>*

The value of *<myURI>* denotes the URI for the dataset.

Example

```
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:/Program%20Files/SPSSInc/Samples/accidents.sav
```

The dataset.table parameter

For IBM® SPSS® Collaboration and Deployment Services Enterprise View data sources, the table to be used by the custom dialog. If no name is specified, the user will be prompted to select from the list of tables available in the DPD.

Syntax

dataset.table=*<myTable>*

The value of *<myTable>* identifies the table to use.

Example

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

The dataset.prompt parameter

Indicates that the user will be forced to select a dataset for the custom dialogs. Otherwise, the dataset selected for the first dialog opened by the user that contains matching search criteria during a session will be used for any subsequent custom dialogs that are not configured to use a specific dataset.

Syntax

dataset.prompt=*<indicator>*

The value of *<indicator>* is either *true* or *false*.

Example

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

The dataset.search.criteria parameter

Search criteria to be used for generating a list of data sets at run time. The entire search string must be entered on a single line. Multiple conditions may be combined using parenthesis and and/or logic.

Search criteria

$$repository/title_field_name=<Object name>

$$search/mimetype=<Object MIME type>

$$repository/version_created_by_field=<Created by user stamp>

$$repository/version_created_date_field=<Version created date>

$$repository/description_field_name=<Object description>

$$repository/object_last_modified_by=<Created by user stamp>

Syntax

dataset.search.criteria=*<myCriteria>*

The value of *<myCriteria>* corresponds to the search expression.

Example

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

# locates all SAV files
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'

# locates all files that match the keyword SPECIAL_DATASET
http://yourserver:8080/peb/view/myDialog.spd
?dataset.search.criteria='$$repository/keyword_field_name%3D%3DSPECIAL_DATASET'
```

The variable.display parameter

The `variable.display` parameter indicates whether or not to show variable names or labels.

Syntax

variable.display=*<type>*

The value of *<type>* is either *names* to show variable names or *labels* to show variable labels.

Example

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

The variable.sort parameter

The `variable.sort` parameter specifies the sort criterion used for ordering variables.

Syntax

variable.sort=*<myCriteria>*

The value of *<myCriteria>* is one of the following:

- *none* to do no additional sorting beyond the original order in the data
- *alphanumeric* for an alphanumeric sort of field names or labels, whichever is displayed
- *measurement* to sort by the field measurement levels

Example

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

The stylesheet.url parameter

If you are using a CSS style sheet stored in the repository, the repository URL of the style sheet.

Syntax

stylesheet.url=*<myURL>*

The value of *<myURL>* is the URL for the style sheet.

Example

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

The stylesheet.name parameter

If you are using a CSS style sheet embedded in the custom dialog file, the name of the style sheet. The style sheet file can be added to the custom dialog file using compressed archive software, such as WinZip.

Syntax

stylesheet.name=*<myStyles>*

The value of *<myStyles>* specifies the name of the style sheet.

Example

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

The javascript.url parameter

If you are using a JavaScript stored in the repository, the repository URL of the script file.

Syntax

javascript.url=*<myURL>*

The value of *<myURL>* is the URL for the JavaScript file.

Example

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

The javascript.name parameter

If you are using a JavaScript sheet embedded in the custom dialog file, the name of the script file.

Syntax

javascript.name=*<myFile>*

The value of *<myFile>* is the name of the JavaScript file.

Example

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

The validate.method parameter

A validation method from the specified JavaScript file to call before a page is submitted. The form that is being submitted should be the only parameter for the method. Upon evaluating the form input, the method should return a Boolean value. The method should return true if everything is valid and false if the submit should be cancelled.

Syntax

validate.method=*<myMethod>*

The value of *<myMethod>* is the name of the method in the JavaScript file to use for validation.

Example

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

The output.format parameter

The format of the output to create. Default format is IBM® SPSS® Statistics Web Output viewer format (*.spw*. In some cases, it may be appropriate to create HTML instead. The output format is case sensitive.

This parameter specifies the same information as the `outputtype` parameter, but is honored only for custom dialogs.

Syntax

output.format=*<myFormat>*

The value of *<myFormat>* is the format for the output. Valid values include the following:

- *SPW* for SPSS Statistics web output viewer
- *HTML* for HTML output

Example

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

The output.filename parameter

The name of the output file. If not specified, the output file will be generated with the same name as the custom dialog file name but without the *.spw* extension.

Syntax

output.filename=*<myFile>*

The value of *<myFile>* is the name for the output file.

Example

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

The showOutline parameter

Indicates whether the outline should be displayed. Default is `true`.

Syntax

showOutline=*<indicator>*

The value of *<indicator>* is either *true* or *false*.

Example

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

The allowPivoting parameter

Indicates whether table manipulation should be allowed. When the option is disabled, the user will not be allowed to pivot, flip, or change layers, save views or open data in a new window. Default is true.

Syntax

allowPivoting=*<indicator>*

The value of *<indicator>* is either *true* or *false*.

Example

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

The allowPrinterFriendly parameter

Indicates whether the printer friendly display can be opened for a particular table. Default is true.

Syntax

allowPrinterFriendly=*<indicator>*

The value of *<indicator>* is either *true* or *false*.

Example

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

The allowDownload parameter

Indicates whether the data can be downloaded to a local data file. Default is true.

Syntax

allowDownload=*<indicator>*

The value of *<indicator>* is either *true* or *false*.

Example

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

The showLogs parameter

Indicates whether log entries should be shown in the output. Default is true.

Syntax

showLogs=*<indicator>*

The value of *<indicator>* is either *true* or *false*.

Example

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

The statistics.server parameter

IBM® SPSS® Statistics server used to execute the syntax of the custom dialog. The value may be a URI or a name that references a server defined in IBM® SPSS® Collaboration and Deployment Services. If you have multiple servers, this value can specify the URI or name of a server cluster.

Syntax

statistics.server=*<serverIdentifier>*

The value of *<serverIdentifier>* identifies the server to use for execution.

Example

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

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

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

The statistics.server.credential parameter

The credential that should be used to connect to the IBM® SPSS® Statistics server when executing syntax. The value may be a URI or a name that references a credential for IBM® SPSS® Collaboration and Deployment Services.

Syntax

statistics.server.credential=*<myCredential>*

The value of *<myCredential>* identifies the credential under which execution occurs.

Example

```
http://yourserver:8080/peb/view/myDialog.spd?statistics.server=localStatisticsServer
&statistics.server.credential=spsscr:///?id=0a30063bc975ede40000011cafb8deda8327.

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

HTML techniques

Use an HTML editor

Many HTML editors can simplify the creation of URL query strings and insert the proper delimiters between parameters.

Use HTML forms to submit requests

IBM® SPSS® Collaboration and Deployment Services Deployment Portal requests can be submitted from HTML forms included on a web page. For example, a form can be used to allow a user to:

■ Select from a list of available reports

■ Select an output file type

■ Specify prompted variables prior to submitting the report request

■ Supply an ID and password prior to running a report

The following example references a custom dialog file in the action for a form.

```
<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>
```

Use the IBM SPSS Collaboration and Deployment Services Repository to store custom web pages containing relative paths

The repository can be used as a central location for storing all files for a custom web site. Relative or absolute paths can be used within the custom web site to link to items such as *.css* style sheets, images, Deployment Portal reporting objects, and JavaScript.

For example, you might store a folder called *MyWebPage* in the repository containing a custom web page called *MyWebPage.htm* and resources such as images, stylesheets, and JavaScript files. *MyWebPage.htm* can contain **relative** references to the resources such as the following:

```
<img src="MyLogo.gif?fragment=true">
<script language="javascript" src="MyJS.js?fragment=true">
</script>
```

```
<LINK REL="StyleSheet" HREF="MyStyles.css?fragment=true"
TYPE="text/css" MEDIA="screen" />
```

Note that for such relative references to work properly, the web page needs to be accessed using the parameter `fragment=true` in the URL. For example:

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

If you want to store the resources for your web site in a different repository location from where your web page is stored, they can be referenced from your web page (for example, *MyWebPage.htm*) using **absolute** paths as follows:

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

Or, they can be referenced by using the full host name and port in the path:

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

IBM SPSS Collaboration and Deployment Services Tag Library

A JavaServer Pages (JSP) tag library is provided with IBM® SPSS® Collaboration and Deployment Services for administrators and advanced users who want to create relationships between repository items and create custom Web pages (*.jsp* pages) containing items that can feed values to one another. The tag library provides the following basic functionality:

Authentication: You can set the user, password, and security provider and share across any items or prompts defined on the page. Authentication is required to access the items in the IBM® SPSS® Collaboration and Deployment Services Repository and for data source authentication.

Items: You can specify the definition of items, including the target "container" (`<div>` or `<iframe>` element). The items will run using a POST request for IFRAME targets and using AJAX (Asynchronous JavaScript and XML) for DIV targets.

Prompts: You can use prompts to dynamically adjust the parameters used to run items. The prompt location is only restricted to a location on the current page. Prompts can either be user defined or a selected parameter from an existing item definition.

Linking Relationships: You can define relationships between the following items:
- Source report items and target report, job, scoring, or custom dialog items
- A list of prompts and a target item. Both the activation location (DIV or IFRAME) and the timing (ONDEMAND, ONLOAD, or NONE) are supported.

The tag library framework is made up of the following main parts:
- Public JavaScript API.
- Custom tags and their interactions with each other.
- Tag library beans for data set retrieval. For more information, see the topic Tag library beans on p. 53.

This document describes each tag function available in the JSP tag library and includes usage examples. After reading this document, we recommend reviewing the sample *.jsp* files shipped with the tag library before creating your own custom pages. For more information, see the topic JavaServer pages samples on p. 58.

***Upgrading to IBM SPSS Collaboration and Deployment Services Tag Library***

Note that previous versions of IBM SPSS Collaboration and Deployment Services used a *.tld* file named *reporting-taglib.tld* or *pasw-taglib.tld*. Any existing JSP pages using either of those names should be updated to reference *IBMSPSSTaglib.tld*.

JavaServer pages architecture

The "JSP Architecture" figure illustrates the architecture underlying the use the tag library. The application server hosting the IBM® SPSS® Collaboration and Deployment Services Repository includes a servlet engine that transforms the information contained in the library tags into input for web services included in IBM® SPSS® Collaboration and Deployment Services.

Figure 4-1
JSP Architecture
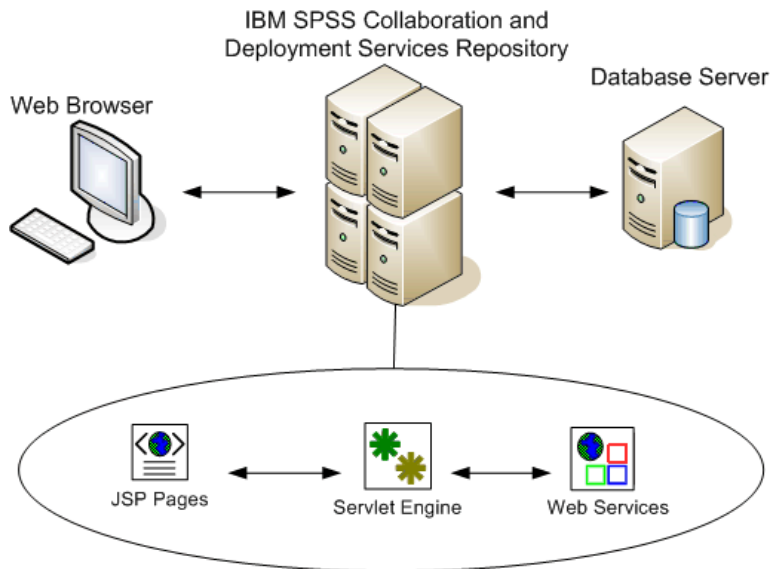
In general, the process of running items using the tag library is as follows:

▶ The JSP developer uses custom tags to define credentials, prompts, items, and item relationships in a *.jsp* file and stores the file on the application server hosting IBM SPSS Collaboration and Deployment Services Repository.

▶ When a client accesses the *.jsp* page, the server evaluates the tags and generates XML data islands or HTML elements as appropriate, which are used by the JavaScript components of the framework to identify and manage relationships between items and prompts.

▶ Validations are carried out in each tag handler and appropriate error messages are displayed on the page so the user is aware of any errors at each stage of page creation.

▶ A servlet provides support for running items and processing and returning the output.

▶ The web service associated with the item type is invoked to run the item and perform various validations.

Supported items

A variety of repository items can be referenced in JSP pages using the IBM® SPSS® Collaboration and Deployment Services Tag Library. When processing the page, the MIME type of the item determines how the item gets processed. Valid items include the following:

- Reports
- Jobs
- Scoring models
- Custom interface definitions

Reports

For a report, the repository item must reference one of the following types of files:

- BIRT Report Designer for IBM® SPSS® report design (**.rptdesign*)
- IBM® ShowCase® report (**.dbq* or **.rpt*)
- Visualization definition (**.viz*)

The following properties should be considered when working with report items:

Output. A report item typically generates a single output. Visualization reports, however, generate an imagemap in addition to the visualization. The output for the item can be delivered in a variety of formats that depend on the report type. Available formats include the following:

- HyperText Markup Language (**.html*)
- Portable document format (**.pdf*)
- Report document (**.rptdocument*)
- HTML Complete (**.htmlc*)
- MIME HTML (**.mht*)
- Microsoft Word document (**.doc*)
- Microsoft PowerPoint (**.ppt*)
- Portable Network Graphic (**.png*)
- Enhanced Metafile (**.emf*)
- Joint Photographic Experts Group (**.jpeg*)

Prompts. When processed, the item will prompt for values for any variables defined in the report.

Location restrictions. Output of the **.rptdocument* type can only be displayed in an IFRAME.

Item linking. Report items can be used as sources for subsequent items or as targets of other items.

Supported tags. Report items do not support the `outputLocation` tag. All other tags in the tag library are supported.

The item may include additional information controlling the output display, such as the window title or the presence of a toolbar.

Jobs

For a job, the repository item must reference a job in IBM® SPSS® Collaboration and Deployment Services, which has a MIME type of *application/x-vnd.spss-prms-job*. The following properties should be considered when working with job items:

Output. A job item can generate any number of outputs of varying types. The output produced depends on the steps contained within the job.

Prompts. When processed, the item will prompt for values for any job parameters defined for the job.

Location restrictions. Output from the individual steps within the job must be explicitly defined.

Item linking. Job items can be used as targets of other items but not as sources.

Supported tags. Job items do not support the `actionHandler` tag. All other tags in the tag library are supported.

Scoring models

For a scoring model, the repository item must reference a file configured for scoring. Valid types of files include:

- Scenario (**.scn*)
- IBM® SPSS® Modeler stream (**.str*)
- Predictive Model Markup Language (PMML)
- Real Time predictive application definition

The following properties should be considered when working with scoring items:

Output. A scoring item produces HTML output.

Prompts. When processed, the item can prompt for values for parameters, a data file, a data provider definition, and a model name.

Item linking. Scoring items can be used as targets of other items but not as sources.

Supported tags. Scoring items do not support the `outputLocation` and `actionHandler` tags. All other tags in the tag library are supported.

Custom dialogs

Note: This functionality requires IBM® SPSS® Statistics adapters in the IBM® SPSS® Collaboration and Deployment Services environment. For more information, see the SPSS Statistics installation documentation.

For a custom web interface, the repository item must reference a dialog definition (**.spd*). The following properties should be considered when working with custom dialog items:

Output. A custom dialog item generates one of the following:

■ A single output file (**.spw*) that must be targeted to a frame or window

■ HTML that can be targeted to a frame/window or a DIV

Prompts. When processed, the item will prompt for values for any prompts defined in the dialog definition. The item can also prompt for data sets. However, any help for prompts defined in the *.spd* file is not used. The application should include its own help references.

Location restrictions. The output can be viewed in a frame, DIV, or a new window.

Item linking. Dialog items can be used as targets of other items but not as sources.

Supported tags. Dialog items do not support the `actionHandler` tag. All other tags in the tag library are supported.

The web deployment properties described for use in a URL referencing a custom dialog item can be specified in the tag library either as properties nested in the `repositoryItem` tag or using the `sourceLinkPrompt` tag. For more information, see the topic Custom dialog parameters in Chapter 3 on p. 17.

The `dataset.uri` and `dataset.table` properties should always be defined, with the latter applying to data provider definitions only. In contrast, the `javascript.url`, `javascript.name`, `stylesheet.url`, and `stylesheet.name` properties are all ignored. Values for those properties should be defined within the JSP itself.

Building an application

Each JSP page in a custom application must define some standard directives to allow the tag library to be used and referenced properly. The first, the `page` directive, sets properties for the entire page itself. These properties include:

■ The *language* attribute defining the scripting language used by the page

■ The *contentType* attribute specifying the MIME type and character set used for responses to clients

■ The *session* attribute indicating whether or not the tag library stores information on the session

The second directive, `taglib`, indicates which tags will be used by the JSP page. Properties defined for this directive include:

■ The *uri* attribute specifying the proper path to *IBMSPSSTaglib.tld*

■ The *prefix* attribute defining a scope for the tags

Note that previous versions of IBM® SPSS® Collaboration and Deployment Services used a *.tld* file named *reporting-taglib.tld* or *pasw-taglib.tld*. Any existing JSP pages using either of those names should be updated to reference *IBMSPSSTaglib.tld*.

The following sample uses the `page` directive to define the content type as text/html using the UTF-8 character set, the scripting language as Java, and use of the session object as true. The `taglib` directive identifies the location of the reporting *.tld* file and specifies a prefix of *r* for all tags defined within.

```
<%@ 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>
```

To put your application into production you should plan on creating a web application archive (*.war*) file containing the *.jsp* files and deploy it as a separate web application on your application server. This is the preferred method.

For example, the structure of expanded sample reporting tag library application archive (*IBMSPSSTaglib.tld.war*) included in the default installation of IBM SPSS Collaboration and Deployment Services is as follows:

```
IBMSPSSTaglib
│   index.html
│   setup.html
├──js
│   <JavaScript files> │
├──jsp
│   <Java Server Page files>
├──META-INF
│   MANIFEST.MF
│├──WEB-INF
│   │   web.xml
│   │   weblogic.xml
│   ├──lib
│   │   <Java archive files>
│   └──tlds
│       IBMSPSSTaglib.tld
│       reporting-taglib.tld
```

```
|└─xsl
```
<Extensible Stylesheet Language files>

Note that the TLD (Tag Library descriptions) file and libraries (.*jar* files) are included in the deployed .*war* file. The TLD file is also referenced in the application descriptor file (*web.xml*):

```
<?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 responsible for fulfilling all requests from
      reporting taglibs
    </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>

<!–End : Taglib Node –>

<!–start : Security-Constraint Node –>
<!–End : Security-Constraint Node –>
</web-app>
```

The application descriptor also specifies that *ReportingTaglibServlet* servlet is mapped to the */taglib* and */reportingTagLib* URL patterns, and either URL would call the servlet. The servlet Java class is *com.spss.report.taglib.servlet.ReportingTaglibServlet*. Optional single sign-on functionality is enabled by a servlet filter *Common Authentication Filter* which uses *com.spss.report.taglib.filter.AuthFilter* class and is mapped to the servlet by URL. The filter is initialized with encoding and SSO adaptor class parameters.

For more information about *.war* files, see online resources such as *http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/WebComponents3.html*. Also see your application server's documentation for additional information and requirements.

Important! Application *.war* files that are not deployed by IBM SPSS Collaboration and Deployment Services installation scripts or IBM® SPSS® Collaboration and Deployment Services Package Manager, such as tag library or custom applications, may need to have class loader order modified. For example, for reporting and scoring tag library applications on WebSphere, class loader order must be set to *Classes loaded with application class loader first* and *.war* class loader policy to *Single class loader for application*.

Implementation details

Users creating custom *.jsp* pages should be aware of the following information:

■ Each time the server stops, any *.jsp* that are placed directly in the *tmp/deploy* directory files are lost. To preserve your *.jsp* files, save backup copies to a local drive and copy them back to the server after each restart. Production applications should by packaged in .war files. For more information, see the topic Building an application on p. 31.

■ Internet Explorer 6.0 and Mozilla Firefox™ 1.5 are supported. Firefox has the following restrictions:
   ■ For reports containing images and/or charts, the *.mht* output format combines all the images/charts and HTML in a single file compatible with Internet Explorer only.
   ■ IBM® ShowCase® Report Writer reports (*.rpt*) use special technology for charts. This technology is only compatible with Internet Explorer.

■ The server pre-validates all tags to the extent possible and writes error messages to the HTTP response as they are identified. This provides the JSP developer assistance in resolving problems as a page is being created. For example, the following items are validated: verify

all required credentials are defined; verify credentials; verify report parameters exist; verify column names exists for a report object; verify the linkage between items is logically sound.

■ The tags require a repository server.

■ The tag library supports links between prompts and items, between row clicks and target items, between reports and drill-down reports in the same frame, and between prompts/table rows and target items opened in separate windows.

■ All linkage behavior is hidden from the user and is defined using `sourceLinkRepositoryItem` and/or `sourceLinkPrompt` JSP tags. The user is not required to understand any technology beyond JSP tag usage.

■ All target items must be predefined with parameters to receive the passed parameters.

■ For custom dialogs, the standard CSS defines default styles that are included automatically at the point the `repositoryItem` tag is used. To override those styles, include a custom style sheet after the `repositoryItem` tag. For example:

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

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

Public JavaScript API

The framework provides JavaScript functions for processing repository items, retrieving bookmarked report values, and retrieving cascading prompt values.

The runRepositoryItem function

The public JavaScript function provided by the framework for running an item is `runRepositoryItem`. It allows the developer to run an item by connecting this JavaScript to an event handler, and activate an item when using prompts. However, when running an item directly using this function, the normal prompt validation is bypassed. It is the application's responsibility to validate any parameters before invoking function.

The function accepts the following three arguments.

■ A string corresponding to the name of the item to execute. The name must have been defined using the `name` attribute of the `repositoryItem` tag.

■ An Array of data values to use as parameter values when running the item. The array has the following structure:

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

```
var linkedData = new Array(thisVar);
```

- An optional parameter specifying a target location for the item output. This follows the same rules as the `location` attribute of the `repositoryItem` tag. It may be the ID of a DIV, the name of an IFRAME or FRAME, *NEW* or *windowName*. If omitted, the default location from the `repositoryItem` is used.

The data value array many be specified in one of the following ways:

- As null (or omitted). In this case, any necessary values are retrieved using any `sourceLinkPrompt` tags defined for the item.
- As the link data from actionHandler. The `actionHandler` tags define functions to call and the parameter values which are captured as part of the event. Those parameters can be passed directly to the `runRepositoryItem` API.
- As user defined. The JavaScript calling the `runRepositoryItem` API can define the values of the array above as necessary. The *columnName* is the name of the column defined in the `sourceLinkVariable` tag. The value is the value to pass as the parameter.

The getBookmarkedValues function

The `getBookmarkedValues` function retrieves the values of cells that have been bookmarked in a BIRT Report Designer for IBM® SPSS® report. The id attribute of the cell should be set to the bookmark value. This function can be used for linking complex BIRT Report Designer for IBM SPSS reports involving crosstabs and nested tables.

The `getBookmarkedValues` function accepts the following arguments:

- A parent node in the DOM of the report that the function needs to traverse to get the cell values matching the items specified in the bookmarks array
- An array of bookmarks defined in the report whose values are needed. For example, ["bookid1","bookid2"].

The function returns an array of data values to use as parameter values when running the report. The array has the following structure:

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

The *columnName* is the name of the bookmark. The *value* is the value of the specified cell that is bookmarked.

The retrievePromptValues function

The `retrievePromptValues` function should be called when using parameters with custom controls, and supports both cascading and non-cascading prompts. Call this function in the body `onLoad` handler to load the initial values of the prompt (or the parent prompt in the case of a cascading prompt). Call this function in the `onChange` handler of the control used to define the cascading parameter. The function will make calls to the server and get the prompt values to fill the parameter controls with updated values depending on the parent parameter value selected.

The `retrievePromptValues` function accepts the following arguments:

■ A string denoting the name of the report containing the definitions of the cascading parameters. The name must have been defined as the `name` attribute of a `repositoryItem` tag.

■ A string corresponding to the name of the parameter in the report. For cascading parameters, this string is the name of the cascading parameter group. The cascading group must be present in the report.

■ A user-defined function that accepts an array of value and display text for the new options. The array can be null, in which case the function should clear the control. This function will be called by `retrieveCascadingPromptValues` to populate the parameter controls with new values.

```
function callback(options) {
  // logic to clear the control
  // logic to add value and display text to control
  for(var i = 0; i <options.length; i++) {
    control.value = options[i].value;
    Display Text for control = options[i].displayText;
  }
}
```

■ An array of the selected preceding values present in the cascading group. This array is only needed for cascading parameters and should be omitted for a non-cascading parameter. The parameters must be in sequential order. To get the list of the parent cascading parameter, specify preceding values:

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

The preceding values array has the following structure. For example, to get the list of cities in MN:

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

IBM SPSS Collaboration and Deployment Services Tag Library tag reference

The various tags included in the IBM® SPSS® Collaboration and Deployment Services Tag Library are dependent on each other and, for validation purposes, need to know that the references are correctly met. The tags must also be defined in the correct sequence. The following sections describe each available tag in detail.

This tag library depends upon JSP 1.1.

The credential tag

The `credential` tag defines both a data source login credential and a repository login credential. The credential is referenced by name for all items and/or prompts defined on the page. It should be defined prior to any tags that may reference the credential. In normal use, it would be the first tag referenced in the JSP.

The `credential` tag can contain `properties` elements. For, example, in the case of J. D. Edwards (JDE) enabled data sources, the `credential` looks like the following:

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

Tag nesting

None

Expected output

None. This tag provides authentication information. The tag does not produce output, but caches the credentials using the `name` attribute as a key for later use with a report or prompt tag.

Table 4-1
Attributes for the credential tag

Name	Required	Description
name	true	Either an internal name for the IBM® SPSS® Collaboration and Deployment Services Repository credential or the name of a data source used in a repository item object. This is used to link items and prompts to this credential and to satisfy any data source logins required by referenced items. ■ For repository credentials, this must match the name provided on the `repositoryCreden-tialName` attribute of the `repositoryItem` tag. ■ For database credentials, the name must match the data source name as it is referenced by the item using that data source. This name is used to store the credential values in a session variable. All credentials must have a unique name.

Name	Required	Description
useSSO	false	Indicates if Single Sign On credential for Kerberos should be used. If this attribute is set to true, then the username, password, and provider attributes must not be specified. When using SSO, the Authentication Filter must be configured in the web.xml file. For more information, see the topic Building an application on p. 31.
credentialDefinitionName	false	The name of a credential defined as a resource in the repository. If this value is specified, the `username`, `password`, and `provider` attributes do not need to be defined as the credential resource includes this information.
provider	false	For repository credentials, this is the optional security provider name. Valid values include: ■ *Native* for the built-in provider ■ *AD/<domain>* for Active Directory, where *<domain>* corresponds to the DNS namespace ■ *ADL/<domain>* for Active Directory with local override, where *<domain>* corresponds to the DNS namespace ■ *iSeries* for IBM i ■ *devldapOpenLDAP* for OpenLDAP If not specified, the built-in repository security provider is used. This attribute is ignored for database credentials.
username	false	The user name to use for authentication.
password	false	The password for the specified username. The password is used internally by the tag library. It is NOT written to the JSP result.

Sample usage

The following sample specifies three credentials. This first is for accessing the IBM SPSS Collaboration and Deployment Services Repository with a specified username and password. The value of *Native* for *provider* indicates that the username/password pair for validation is defined in

the native local security provider. The second credential employs single sign-on for the IBM SPSS Collaboration and Deployment Services Repository using the user's previously authenticated credentials. The third credential is for a data source named *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'/>
```

The repositoryItem tag

The `repositoryItem` tag is the main tag for for defining repository item definitions that will be used by the application. The `repositoryItem` tag may reference reports (IBM® ShowCase® reports, BIRT Report Designer for IBM® SPSS® reports or visualization reports), jobs, scoring items, or SPD files. The repository items may be run directly, used to provide prompts, or ran programatically.

Any `sourceLinkPrompt` and `sourceLinkRepositoryItem` tags should be nested within the `repositoryItem` tag as follows:

- Use a nested `sourceLinkRepositoryItem` tag if this item will be run when the user clicks on a different item.

- Use `sourceLinkPrompt` when the parameter values will come from prompts defined on the page or defined directly in the item.

You may optionally specify aditional properties that are specific to a type of repository item. The property names must be in lower case for them to work in the Firefox browser. These property values will be passed to the URL to run the repository item. The properties are specified as a nested XML block.

Tag nesting

This tag may include one `sourceLinkRepositoryItem` and/or multiple `sourceLinkPrompt` and `outputLocation` tags.

Table 4-2
Attributes for the repositoryItem tag

Name	Required	Description
name	true	Defines a unique name for the item. The name can then be referenced by other tags or via the `runRepositoryItem()` JavaScript API.

Name	Required	Description
inputURI	true	The item definition to be used to render the report output. This value must specify a URI that may be used to locate the item definition. The following URI schemes are supported: ■ **file**: References a specific file on the application server and/or a network file location ■ **spsscr**: References a file in the IBM® SPSS® Collaboration and Deployment Services Repository. This scheme allows files to be referenced by identifier or hierarchical path within the repository. Specific version markers can be specified. If no version or label is specified, the latest version is used. ■ **scoring**: References a model configuration from the repository. Scoring configurations are referenced by name from the tag libraries. If a scoring configuration is renamed, the tag library reference must also be modified.
activate	true	Specifies when the item will be activated. Options include: ■ **ONDEMAND**: Runs the item when activated by a row click on a source report. ■ **ONLOAD**: Runs the item when the page initially loads. ■ **NONE**: Item does not run automatically. In this case, the item is used to provide prompts or prompt values. Regardless of the activate setting, any report may be run programatically by using the public JavaScript `runRepositoryItem()` API.

Name	Required	Description
location	false	The destination for the output resulting from running the item. The usage varies slightly depending on what the target type is. ■ For DIV targets, the location should specify the ID of the DIV tag where the output is to be placed. ■ For IFRAME targets, the location must specify the name of the frame. ■ To open the output in a new window, specify a location of `*NEW`. ■ To direct the output to a named window, use an asterisk (`*`) followed by the window name. For example, `*MYWINDOW` will open a new window called *MYWINDOW* and will reuse that window on each activation of the link.
repositoryCredentialName	true	The name of the credential that should be used when accessing the item from the repository and running the item via the reporting service. The credential should have been previously defined using the `credential` tag.
outputType	false	The type of output to generate. The supported output types vary by item type. Normally this will be either HTML or PNG but other options include: **BIRT Report Designer for IBM SPSS Reports:** HTML, PDF, RPTDocument, PowerPoint, Word Document, HTMLC **Visualization Reports:** PNG, EMF, JPEG, HTML **IBM® ShowCase® Query:** HTML, PDF, CSV, WK4, Excel 97/2003, xlsx **IBM® ShowCase® Report Writer:** HTML, PDF, MHT If not specified, the output will default to HTML (or PNG for visualization reports). When a BIRT Report Designer for IBM SPSS or ShowCase report is used as a link source, the `outputType`

Name	Required	Description
		will be ignored and HTML generated since the other output types don't support linking. To display reports using the viewer, specify a type of *RPTDocument*. For this type, the target must be an iframe.
showTitle	false	Specifies whether the title bar of the BIRT Viewer is to be shown. Specify the value as either true or false. This setting only applies when the outputType is RPTDocument which displays in the Report Viewer. Default is true.
title	false	Specifies the title for BIRT Viewer. This setting only applies when the outputType is RPTDocument which displays in the Report Viewer. If not specified, the default title is displayed.
showToolBar	false	Specifies whether the ToolBar of the BIRT Viewer is to be shown. Specify the value as either true or false. This setting only applies when the outputType is RPTDocument which displays in the Report Viewer. Default is true.
showNavigationBar	false	Specifies whether the navigation bar of the BIRT Viewer is to be shown. Specify the value as either true or false. This setting only applies when the outputType is RPTDocument which displays in the Report Viewer. Default is true.
width	false	This is the width of the output image. The width must be greater than 0 and specified in conjunction with the height. If not specified, the default width and height are used.
height	false	This is the height to use when output is an image. The width must also be specified or the setting will have no effect. The value must be greater than 0.

Sample usage

The following sample defines an item named *AllCountries* for a BIRT Report Designer for IBM SPSS report stored in the IBM SPSS Collaboration and Deployment Services Repository.

```
<r:repositoryItem name="AllCountries"
  inputURI="spsscr:///SampleReports/BIRT/CountrySales.rptdesign"
  repositoryCredentialName="repositoryCredential"
```

```
  outputType="HTML" width="400" height="300"
  activate="ONLOAD" location="ReportDIV">
</r:repositoryItem>
```

To prompt for parameter values for an item, include a *sourceLinkPrompt* tag. The following sample retrieves a value for the parameter *ShipCountry* using the JavaScript function *getValue*.

```
<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>
```

To run a second item in response to a user action, include a *sourceLinkRepositoryItem* tag. The following sample runs the visualization report *CityDetails* in response to an action in the source report *AllCountries*.

```
<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>
```

The repositoryItemPrompt tag

The `repositoryItemPrompt` tag generates the HTML for a prompt variable that is defined in the referenced item. The item that the prompt is referencing must be defined using the `repositoryItem` tag before this tag can be used. Use this when you want prompt controls such as those used in IBM® SPSS® Collaboration and Deployment Services Deployment Portal to be used in your application.

This tag generates the HTML prompt controls in the location corresponding to where the tag is used. The tag must be associated with a particular parameter of an item to be useful. The association with parameters is done using the `sourceLinkPrompt` tag, where the `promptID` of the `sourceLinkPrompt` must match the `promptID` of this tag.

Tag nesting

None

Expected output

An HTML element that allows the user to select and/or type personal values depending on the `promptType`, which is selected as `parameterName`. The `repositoryItemPrompt` tag supports all parameters Deployment Portal supports. As a result, all types of prompts are supported and the appropriate HTML element is generated.

Table 4-3
Attributes for the repositoryItemPrompt tag

Name	Required	Description
promptId	false	A unique identifier that can be referenced from the `promptId` attribute of the `sourceLinkPrompt` tag.
repositoryItemName	true	A reference to the name of the item as defined in the `name` attribute of the `repositoryItem` tag.
parameterName	false	Name of the prompt variable as defined in the item.

Sample usage

The following sample prompts for a value for the *EmployeeID* parameter in the *Employees* report.

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

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

The report tag

This tag is deprecated. Use the `repositoryItem` tag instead

The reportPrompt tag

This tag is deprecated. Use the `repositoryItemPrompt` tag instead

The outputLocation tag

This tag associates the generated output that exists in the repository with the location on the page where the output is displayed. When the item runs, the output is retrieved from the repository and displayed at the specified target location on the page.

This tag must always be nested within a `repositoryItem` tag.

Tag nesting

None

Table 4-4
Attributes for the outputLocation tag

Name	Required	Description
outputId	false	This is the path to the output that exists in the repository. For custom dialogs, this attribute should be omitted. The output from running the syntax is automatically detected.
location	true	This attribute specifies where the output should be placed on the page. ■ For DIV targets, the location should specify the ID of the DIV tag where the output is to be placed. ■ For IFRAME targets, the location must specify the name of the frame. ■ To open the report output in a new window, specify a location of *NEW. ■ To direct the output to a named window, use an asterisk (*) followed by the window name. For example, *MYREPORTS will open a new window called *MYREPORTS* and will reuse that window on each activation of the link. HTML outputs may target a DIV. All other outputs should target an IFRAME or a window.
partId	false	This is used to identify the specific part or item of the SPW archive output. **Note:** This functionality requires IBM® SPSS® Statistics adapters in the IBM® SPSS® Collaboration and Deployment Services environment. For more information, see the SPSS Statistics installation documentation.

Sample usage

The following sample specifies an output location for a chart stored in the IBM® SPSS® Collaboration and Deployment Services Repository using the *ChartFRAME* IFRAME tag.

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

If the attribute values depend on parameter values, use the *sourceLinkPrompt* tag to define matches for the parameters. If a match is found, it is substituted for the parameter. For example, the following sample defines two *outputLocation* tags with filenames that depend on parameters.

```
<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>
```

For *JobParam1*, a value of *Jan* is substituted in the name, resulting in *output_chart_Jan.png* appearing at *ChartFRAME*.

For *JobParam2*, the value associated with the html control for the parameter is substituted in the name. If that value is *Illinois*, the file *output_tab_Illinois.html* appears at *ReportDIV*.

The sourceLinkPrompt tag

The `sourceLinkPrompt` tag associates the item parameters with the prompts providing their values. These could be user defined HTML elements, javaScript functions, prompts created using the `repositoryItemPrompt` tag, or directly specified values.

The `sourceLinkPrompt` tag must always be nested within a `repositoryItem` tag. When the item runs, the parameter values are retrieved using the `sourceLinkPrompts`.

Tag nesting

None

Validations performed

None

Expected output

None

Table 4-5
Attributes for the sourceLinkPrompt tag

Name	Required	Description
targetNameParameter	true	Name of the report parameter as it is defined in the report definition.

Name	Required	Description
promptId	false	The `promptId` could be the ID of a `reportPrompt` tag or the name of an HTML control. When a prompt value is needed, the `reportPrompt` or the HTML control will be used to determine the prompt value. Either `promptId`, `parameterValue` or `getValueJSFunction` should be specified.
parameterValue	false	Specifies a value for the parameter instead of prompting for one. This should be specified when the application knows the parameter value when the JSP is being processed. In that case, the value can be specified directly using this attribute. If `parameterValue` is specified, then `promptId` and `getValueJSFunction` should not be used.
getValueJSFunction	false	Identifies a function to call to retrieve the prompt value(s). The function should return either a single value or return an array of values. This attribute should include the function name, parentheses and any parameters as necessary. For example, for a function called *MyGetValues* that takes one parameter, set the attribute to `MyGetValues('myPromp-tID')`.
validateJSFunction	false	Identifies a function to call to to provide validation of the prompt. The function should return `true` if the prompts are valid. This attribute should include the function name, parentheses and any parameters as necessary. For example, for a function called *MyValidate* that takes one parameter, set the attribute to `MyValidate('myPromptID')`.

Sample usage

The following sample prompts for two parameter values using `reportPrompt` tags. The `sourceLinkPrompt` tags for the *CountrySales* report use the identifiers for those prompts to supply their values to the report.

```
<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>
```

The sourceLinkRepositoryItem tag

The `sourceLinkRepositoryItem` tag identifies the source item and variables used to satisfy the item's defined parameters. Using this mechanism, when the source item is clicked, the parent item runs using the parameters defined within the nested `sourceLinkVariable` tags.

This tag must always be nested within a `repositoryItem` tag. It should contain one or more nested `sourceLinkVariable` tags.

Tag nesting

The `sourceLinkRepositoryItem` tag contains one or more `sourceLinkVariable` tags that identify the source column and the target parameter names.

Expected output

None

Table 4-6
Attributes for the sourceLinkRepositoryItem tag

Name	Required	Description
sourceName	true	Name of the `repositoryItem` that will serve as the source of the relationship
linkType	false	Determines what action on the source report will trigger the running of the current report. Currently there is only one supported `linkType`, *row*. For this type, when a row in the source report is clicked, the target report runs. In future releases, additional linkTypes may be added.

Sample usage

The following sample identifies *CityDetails* as the report to run in response to a user action in the *AllCountries* report.

```
<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>
```

The sourceLinkReport tag

This tag is deprecated. Use the `sourceLinkRepositoryItem` tag instead

The sourceLinkVariable tag

The `sourceLinkVariable` tag defines the mapping between the variable or column to use in the source item and the parameter as defined in the target item. This tag must always be nested under a `sourceLinkRepositoryItem` tag.

Tag nesting

None

Validations performed

None

Expected output

None

Table 4-7
Attributes for the sourceLinkVariable tag

Name	Required	Description
columnName	true	For IBM® ShowCase® reports, this attribute specifies the name of the column in the source report. For Visualization reports, this attribute contains the id of the `sourceVariable` or `derivedVariable` element of the Visualization specification. Currently only categorical variables are supported.
targetNameParameter	true	Name of the parameter in the target query

Sample usage

The following sample maps the *ShipCountry* variable in the *AllCountries* report to the *ShipCountry* parameter in the *CityDetails* report.

```
<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>
```

The actionHandler tag

Defines the action handlers that should be applied to the item. When action handlers are defined, the automatic linking setup using `sourceLinkRepositoryItem` no longer applies. The application builder is responsible for running any target items using the `runRepositoryItem` public Java script API.

Tag nesting

Any data values that need to be passed as parameters to the JavaScript function should be defined using nested `actionParameter` tags.

Table 4-8
Attributes for the actionHandler tag

Name	Required	Description
event	true	The event name. Valid events include: ■ *onclick* ■ *onmouseover* ■ *onmouseout*

Name	Required	Description
function	true	The name of the Java Script function to call when the event occurs. This should be the function name only, without () or any parameters.
partId	false	This is used to identify the specific part of the report that the actions should apply to.

Sample usage

The following `repositoryItem` tag defines three action handlers, one for each type of event that could occur. Each handler calls a unique JavaScript function that defines the subsequent processing.

```
<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>
```

The actionParameter tag

There should be an `actionParameter` for each data value from the item that needs to be passed to the `actionHandler` JavaScript function. This tag must be nested within the `actionHandler` tag.

Tag nesting

None

Table 4-9
Attributes for the actionParameter tag

Name	Required	Description
name	true	Name of the column or variable that defines which value from the report results should be passed to the function. ■ For visualization reports, the name is the id attribute of the sourceVariable or derivedVariable element. Currently only categorical variables are supported. ■ For IBM® ShowCase® reports, this would be the column name.

Sample usage

The following sample defines an `actionParameter` named *ShipCountry* that gets passed to the JavaScript function *myOnClick* when the user clicks the report.

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

Tag library beans

The framework includes tag library beans that can be used together for a variety of purposes. For example, the beans can be used to retrieve a data set that can then be used to build custom HTML controls.

In order to use the beans, you must first declare references to them in the JSP. This is done through the import attribute of the page directive.

```
<%@ 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"
%>
```

The code samples for beans use the JavaServer Pages Standard Tag Library (JSTL) which should be included using the taglib directive.

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

For more information on JSTL, refer to the Sun documentation (*http://java.sun.com/products/jsp/jstl/*).

Credential bean

The Credential bean defines the credentials that will be used by other beans. The code sample below creates two credentials and stores them in a HashMap. In the sample below, the *localhost* credential provides the logon information for the IBM® SPSS® Collaboration and Deployment Services Repository. The *ps4008* credential is for a SQL Server data source called ps4008 that is referenced by the report definition.

```
<%
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

The ReportBean is used to retrieve the data for a data set that is defined in a report definition. The code below uses the previously created credentialMap to retrieve a data set. Visualization reports do not support this function.

```
<%–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>
```

The properties used in this code are the following:

- The *reportDefinitionURI* property specifying the location of the report
- The *repositoryCredentialName* property identifying the host
- The *port* property defining the port
- The *dataSetName* property specifying the name of the data set as defined in the report definition for BIRT Report Designer for IBM® SPSS® reports. This does not apply to IBM® ShowCase® reports and should be omitted.
- The *credentialMap* property defining a reference to a HashMap containing the credentials to use

The ReportBean can then be run to return the data set. The data can be used to generate a list control as shown in the code below.

```
<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

The SearchBean bean provides a query mechanism for locating content in the repository that meet specified criteria. For example, the bean can retrieve a list of data provider definition and IBM® SPSS® Statistics data file (*.sav*) sources in the IBM® SPSS® Collaboration and Deployment Services Repository that match a specified search criterion. The code below defines bean properties to query for all data provider definition and SPSS Statistics data sources using the MIME types associated with those sources.

```
<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>
```

The properties used in this code are as follows:

- The *request* property defining an `HttpServletRequest` object.

- The *credentialName* property specifying the credential needed to connect to repository. In this case, the value corresponds to the credential *AuthenticationCredential* defined using the `credential` tag.

- The *searchQuery* property specifying a string denoting the search criterion. The structure of this string matches the syntax used for the dataset.search.criteria parameter for URL strings.

The SearchBean can then be run to return the matching data sources. The code below presents the name, modification date, version label, and author metadata for the data sources in a table.

```
<Table border="0" height="100%" width="100%" cellpadding="0" cellspacing="0">
  <tr>
    <td align="center" bgcolor="#EEEEEE"  style="padding-top:5px; padding-bottom:5px;">
      Data Source
    </td>
```

```
              <td align="center" bgcolor="#EEEEEE"  style="padding-top:5px; padding-bottom:5px;">
                Modified Date
              </td>
              <td align="center" bgcolor="#EEEEEE"  style="padding-top:5px; padding-bottom:5px;">
                Version Label
              </td>
              <td align="center" bgcolor="#EEEEEE"  style="padding-top:5px; padding-bottom:5px;">
                Author
              </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

The PevMetaDataBean bean retrieves variable metadata from data provider definition and
IBM® SPSS® Statistics data file (*.sav*) sources. The code below defines properties for the bean
to query a *.sav* file.

```
<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>
```

The properties used in this code are as follows:

- The *request* property defining an HttpServletRequest object.

- The *dataseturi* property specifying the URI for the data file or data provider definition containing the variables.

- The *credentialName* property defining the credential needed to connect to repository. In this case, the value corresponds to the credential *AuthenticationCredential* defined using the `credential` tag.

The PevMetaDataBean can then be run to return the metadata for the variables in the dataset. The code below presents the metadata in a table.

```
<Table border="0" height="100%" width="100%" cellpadding="0" cellspacing="0">
  <tr>
    <td align="center" bgcolor="#EEEEEE" style="padding-top:5px;
    padding-bottom:5px;">
    Variable Name
    </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

The ScoringBean bean retrieves a list of scoring configurations for a specified model that are able to respond to a scoring request. The `getScoringConfigurations` method of the bean accepts the following parameters:

- The *credential* parameter specifying the credentials for accessing the IBM® SPSS® Collaboration and Deployment Services Repository defined using the `Credential` bean.

- The *modelLocationUri* parameter identifying the URI for a model in the IBM SPSS Collaboration and Deployment Services Repository.

Alternatively, instead of supplying a `Credential` bean item, the following two parameters can be used for specifying credentials:

- The *request* parameter specifying an `HttpServletRequest` object.

- The *credentialName* parameter defining the credential needed to connect to the IBM SPSS Collaboration and Deployment Services Repository defined using the `credential` tag.

The following code retrieves the scoring configurations for the model *KMeans.xml* that can respond to a scoring request using a credential defined using the `credential` tag:

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

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

The array returned by the bean can be used to populate a form from which a user can select a scoring configuration to use for subsequent scoring.

```
<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>
  Select Scoring Configuration:
  <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 samples

IBM® SPSS® Collaboration and Deployment Services includes a variety of JSP samples illustrating the use of the tag library. The samples are grouped into the following categories:

■ **Reporting.** Using BIRT Report Designer for IBM® SPSS® and visualization reports interactively, including running a second report in response to a selection. To access these samples, go to:

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

■   **Scoring.** Generating scores for a predictive model configured for scoring. To access these samples, go to:

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

■   **SPSS Statistics syntax.** Generating and executing IBM® SPSS® Statistics syntax, as well as working with the resulting output. To access these samples, go to:

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

Note: This functionality requires SPSS Statistics adapters in the IBM SPSS Collaboration and Deployment Services environment. For more information, see the SPSS Statistics installation documentation.

If the URL for a set of samples fails to return an introduction page, the package or war file containing the samples may not be deployed to the IBM® SPSS® Collaboration and Deployment Services Repository server. Use the Package Manager tool to deploy the desired package or deploy the war file in accordance with the documentation for your application server.

On the introduction page for the samples, click View Source for any sample to examine its source code. To explore their functionality, you can run the samples from the page by clicking Run. However, successful execution requires the following:

■   Sample resources in a specific folder structure in the IBM SPSS Collaboration and Deployment Services Repository

■   Valid credentials for accessing the resources referenced in the samples

Instructions for configuring the environment for successful sample execution are available from the introduction page for the samples.

Portal integration

The IBM® SPSS® Collaboration and Deployment Services Web services architecture provides the ability to integrate it with portal servers. This enables delivery of highly customized content through pluggable user interface components that utilize Web services to produce fragments of markup code that are aggregated into a portal page. Typically, a portal page is displayed as a collection of non-overlapping windows, where each window displays a segment of content. Some examples of portal applications are e-mail, weather reports, discussion forums, and news. Similarly, IBM SPSS Collaboration and Deployment Services portals can be used to deliver customized content, such as output of reports and analytical processing, charts, diagrams, etc.

The repository supports portal integration based on JSR 168 standard. JSR 168 proposed by Java Community Process group (*http://jcp.org*), enables interpretability for portlets between different Web portals. This specification defines a set of APIs for interaction between the portlet container and the portlet, addressing the areas of personalization, presentation and security. Implementation of JSR 168 include IBM Web Portal from WebSphere, Oracle Application Server Portal 10g, BEA WebLogic Portal, Vignette Portal, Sun Portal Server, and JBoss.

The repository also supports portal integration with Microsoft SharePoint server using Web Parts.

Officially supported portal environments include:

- 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

The repository may also be integrated with other portal environments based on JSR 168 and J2SE 5.0.

IBM® SPSS® Collaboration and Deployment Services Portlet and IBM® SPSS® Collaboration and Deployment Services Web Part can be used to deliver repository content to portal users. The architecture also enables creation of custom JSR 168-compliant portlets and SharePoint Web Parts that use IBM SPSS Collaboration and Deployment Services Web services.

Installation

IBM® SPSS® Collaboration and Deployment Services portal components are distributed on installation Disk 2 in */PORTLET* as *IBMSPSSPortlet.war* (portlet) and *IBMSPSSWebPart.wsp* (Web Part).

IBM SPSS Collaboration and Deployment Services Portlet installation

▶ The procedure for installing *IBMSPSSPortlet.war* varies depending on the portal server type. Refer to portal server vendor documentation for details.

IBM SPSS Collaboration and Deployment Services Web Part installation

SharePoint Web Part installation prerequisites include:
- Microsoft SharePoint 2007
- Microsoft Web Service Enhancement 2.0 (WSE 2.0 SP3)

To install IBM® SPSS® Collaboration and Deployment Services Web Part:

1. Copy *IBMSPSSWebPart.wsp* from the repository installation disk to a predefined location on the SharePoint host, for example, *c:\temp*.

2. From the */bin* directory of the SharePoint server installation run the following commands:

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

3. Use SharePoint administration utilities to add the Web Part to the Web Part gallery and to subsequently deploy it. For more information, see Microsoft SharePoint documentation.

Once the component has been installed, it must be configured to access a specific resource in the repository. Component preferences must also be set up.

Configuration

After the portal component has been installed and the portal page layout has been completed, you will be prompted to configure the component to access a repository resource. The general procedure for configuring portal access involves defining the repository server, specifying repository credentials, selecting the resource to be delivered to the portal, and if necessary, specifying data source credentials and default prompt values. You can also configure components' appearance and behavior by setting the preferences.

Configuring IBM SPSS Collaboration and Deployment Services Portlet

Open the portlet configuration page. The page may open differently depending on the portal server type.

1. Specify the repository host and port and whether the server requires a secure connection.

Figure 5-1
Portlet configuration: specifying the repository

2. Specify the repository user credentials and security provider for login authentication.

Figure 5-2
Portlet configuration: specifying repository credentials

3. Select the repository resource to be delivered to the portal. Make sure the correct resource version is specified.

Figure 5-3
Portlet configuration: selecting the resource

4. If necessary, specify the credentials for the data source referenced by the resource; for example if a report uses a database, database credentials must be provided. Note that depending on the resource, it may be necessary to specify credentials for multiple data sources

Figure 5-4
Portlet configuration: specifying data source credentials

IBM® SPSS® Collaboration and Deployment Services Portlet - Configuration	? Cancel
Enter Deployment Server DataSource Login	

Data Source: bank_db

User ID: [ ]

Password: [ ]

Previous   Next

5. If the resource includes prompts (for example, a report may allow for a dynamic selection of values), specify the default prompt settings.

Figure 5-5
Portlet configuration: setting default prompt values

IBM® SPSS® Collaboration and Deployment Services Portlet - Configuration	? Cancel

Select Parameter Values
Prompt Group:

Bi-Weekly ▾   [ ]

Previous   Next

6. Verify that configuration information is correct. To start over, click Refresh.

Figure 5-6
Portlet configuration: confirmation page

IBM® SPSS® Collaboration and Deployment Services Portlet - Configuration	? Cancel
Confirm Configuration Input	

Deployment Server: CDSServer
Port: 8080
Protocol: http
Repository User: Native/admin
Report URI: spsscr:///Reports/churn_analysis.rptdesign
Data Source: bank_db
Data Source User: CDSUser
Parameter Value Assignment: DateRange

Refresh   Next

7. Click Next to proceed to viewing the resource.

Figure 5-7
Portlet configuration: completion message

IBM® SPSS® Collaboration and Deployment Services Portlet - Configuration	? Cancel

Configuration complete.
You may now return to View mode to validate your work.

Next

Portlet settings can been edited after the initial configuration: for example, it can be pointed to a different repository resource if necessary.

▶ Certain aspects of the appearance and behavior of the portlet are set through its preferences. The following preferences are available:

Preference	Description
expiration-cache	The expiration period for the portlet cache, i.e., the time in seconds after which the portlet output expires. -1 indicates that the output never expires. The default value is 600.
log-messages	Specifies whether portlet messages will be appended to the portal server log file. The default value is NO.
reenter-dsLogin	Specifies whether the user must provide the data source credentials for the portlet instance every time she logs into the portal. The default value is NO.
reenter-parameter	Specifies whether the user must reenter the prompt values for the portlet instance every time she logs into the portal. The default value is NO.
refresh-parameter	Specifies whether the user can enter different parameter values and re-display the content based on those values. The default value is NO.
use-single-sign-on	Specifies whether the portlet will be used with single sign-on. The default value is NO.
validate-input-parameter	Enables user input parameter validation in order to protect against cross-site scripting attacks. The default value is YES.
window-height	The height of the portlet window (pixels). The default value is 750.
window-title	Descriptive name for the portlet instance.
window-width	The width of the portlet window (percent). The default value is 100%.

Preferences are set with portal server administration facilities and the way they are accessed will differ depending on the server type.

Configuring IBM SPSS Collaboration and Deployment Services Web Part

Web Part configuration involves the same basic steps as the portlet configuration: setting up access to the repository resource and configuration option. Note that the number of displayed items in the repository tree (when you select the resource) is controlled by an additional configuration option.

Single sign-on

IBM® SPSS® Collaboration and Deployment Services allows single sign-on access, and special configuration of the portal server may be required to enable it for the portlet or Web part. The procedures for enabling single sign-on will be different depending on the portal server. For example, JBoss portal configuration is as follows:

▶ *<JBoss installation folder>/bin/run.bat* file must be modified to include Java arguments for Kerberos-based single sign-on as in the example below.

```
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
```

▶ The following section must be added to *<JBoss installation folder>/server/default/conf/login-config.xml*.

```
<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>
```

Known issues

■ When the portlet is used with JBoss portal, the repository tree view may not expand. In order to correct the problem, modify the *<JBoss installation folder>/bin/run.bat* (*run.sh* on UNIX) to increase the new generation and permanent generation size by adding the following arguments to JAVA_OPTS:

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

■ Cookie settings in the Safari browser may prevent some repository artifacts from being displayed in the portlet without first prompting for credentials. The browser cookie policy should be set to *Always* instead of *Only from sites I visit* to avoid repeated requests for credentials.

HTML archive

An HTML report typically involves a number of HTML files displaying a variety of referenced images using style sheets to control the appearance of the output. Due to the number of files involved, managing and sharing this output can be a challenge. If one file is missing or incorrectly referenced, the pages do not display correctly.

The HTML Archive, or HTMLC, format addresses the issue of managing numerous intra-linked files by placing all associated HTML artifacts into a single, cross-browser archive file. The IBM® SPSS® Collaboration and Deployment Services Repository includes a viewer enabling a variety of client applications to display the contents of the archive. When accessing an HTMLC file stored in the repository, relative cross-references within the archive are silently replaced with full paths that reference the archive file. This allows links to files within the archive to resolve completely and display correctly.

File structure

An HTMLC archive file contains the following:

■ A primary HTML file at the root of the archive. When rendering an HTMLC archive, the viewer uses the first file with an *.html* extension at the archive root as the primary file.

■ Secondary files referenced by the primary file, such as cascading style sheets, images, javascript, or other HTML files. Secondary files can exist in any folder within the archive.

All references to files within the archive should use relative paths.

Creating HTMLC files

HTMLC files can be created in BIRT Report Designer for IBM® SPSS® when working with report designs stored in the IBM® SPSS® Collaboration and Deployment Services Repository. However, custom HTMLC files can also be created using a file archiver such as the Java Archive tool or WinZip. To manually create an HTMLC file:

1. Create the structure for the files in the file system.

2. Create an archive containing those files and folders, specifying an extension of *.htmlc* for the output file.

The files in the archive may be created manually or automatically. In IBM® SPSS® Statistics, for example, you can export the results of an analysis as HTML. The resulting HTML and image files can be archived as an HTMLC file. Alternatively, you can use an HTML editor to manually create pages to be archived.

Custom HTMLC file example

For this example, consider the folder structure shown in the "Archive Files" figure.

Figure 6-1
Archive Files

The HTML file *gss.html* references images contained in the *images* folder and uses styles contained in a cascading stylesheet in the *css* folder. Using the Java Archive tool, the following command creates an HTMLC file named *custom.HTMLC* containing the files.

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

The contents of the resulting archive appear in "HTMLC Archive" figure.

Figure 6-2
HTMLC Archive

Storing this single archive in the repository allows the *gss.html* page to be displayed in repository clients, such as the IBM® SPSS® Collaboration and Deployment Services Deployment Portal or IBM® SPSS® Collaboration and Deployment Services Deployment Manager, with its referenced graphics using the defined styles.

Customization example

The Model Management page of IBM® SPSS® Collaboration and Deployment Services Deployment Portal provides the ability to monitor the ongoing performance of models deployed to IBM® SPSS® Collaboration and Deployment Services Repository as scenario files. These scenarios are associated with jobs that can be executed on demand or scheduled. Scenario files are created with the IBM® SPSS® Modeler application, and they use streams for underlying analytical processing. Model evaluation and champion challenger jobs are set up and executed using IBM® SPSS® Collaboration and Deployment Services Deployment Manager, and Deployment Portal is used only to view the results. The information displayed as tabs on the Model Management page can include a listing of the best and worst performing models, trends of model performance, champion models, and a listing of all deployed scenarios. The options on the Configuration tab can be used to specify display parameters and show or hide individual tabs.

For information on using the Model Management page, see the Deployment Portal help system.

The user interface mainly consists of a single Java server page (JSP), *MMDMaster.jsp*. The interface components rendered on the page are either reports from BIRT Report Designer for IBM® SPSS® or Visualization reports. These reports are rendered using the IBM® SPSS® Collaboration and Deployment Services Tag Library. The page itself is integrated into Deployment Portal using the Tab Extension framework.

IBM SPSS Collaboration and Deployment Services Tag Library

The IBM® SPSS® Collaboration and Deployment Services Tag Library provides support for running the BIRT Report Designer for IBM® SPSS® and Visualization reports that generate the bulk of the content on the Model Management page. The tag library also supports interactivity between reports, allowing a source report to invoke a target report. The source report passes parameters to the target report for processing.

Report definitions

The Report definitions used by the Model Management page are stored in the following directory within the IBM® SPSS® Collaboration and Deployment Services Repository installation:

<installation-directory>\components\peb-mmd\reports

To examine the reports, open the files in BIRT Report Designer for IBM® SPSS®. The visualization reports can be opened using the IBM® SPSS® Visualization Designer, or a text or XML editor.

The reports are provided for reference purposes, and should not be directly modified. Any modification of the reports will not be supported by SPSS Inc.. However, you may copy the reports and modify the copies as desired.

Running BIRT Report Designer for IBM SPSS reports

The Model Management page involves four master reports that are displayed in four tabs. These tabs are displayed in the "Model Management tabs" figure.

Figure 7-1
Model Management tabs
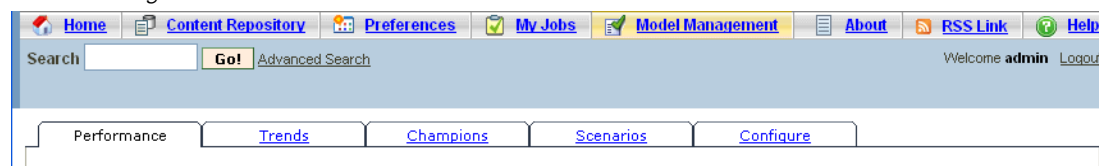
Each of the tabs corresponds to a master BIRT Report Designer for IBM® SPSS® report. When the tab loads, the master report associated with the tab runs using the IBM® SPSS® Collaboration and Deployment Services Tag Library framework.

On the master JavaServer Page, there is one tag for each report. The example below shows the tag used for the Performance Tab.

```
<!--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>
```

The repositoryItem tag

The report information is specified in the repositoryItem tag.

```
<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">
```

The repositoryItem tag has following attributes:

- The *name* attribute which uniquely identifies the repositoryItem tag. The runRepositoryItem public API uses this name to render the report. For the Performance tab, the name is *Performance_Tab_Report_Tag*.

- The *inputURI* attribute which specifies the location of the report. For the Model Management page, all of the reports are picked up from the server's file system in the *peb-mmd* directory of the IBM® SPSS® Collaboration and Deployment Services Repository installation. The URI specified must be a valid URI.

- The *activate* attribute determines when the report is run. For the Performance tab, the value is *ONLOAD*, indicating the report will run when the page loads. A value of *ONDEMAND* indicates that the user is responsible to initiate running of the report by calling the runReport public API provided by the Reporting tag library. For more information, see the topic Javascript API on p. 72.

- The *location* attribute which specifies the location in which the report is to be rendered. This attribute corresponds to the *id* of the HTML element, which can be either a DIV or an IFRAME. For Model Management, the report location always points to a DIV.

- The *outputType* attribute which specifies the format in which the report is to be rendered using the IBM® SPSS® Collaboration and Deployment Services Tag Library. The output format specified must be one which is supported by the BIRT Report Designer for IBM® SPSS® Report Engine. For the reports used in Model Management, the output type is always *HTML*.

The sourceLinkPrompt tag

The sourceLinkPrompt tag specifies the linking of prompts to the report. In other words, this tag specifies how the report acquires the prompt values while running.

There are two ways in which the prompt values are specified for Model Management. The first method is using the *parameterValue* attribute, such as:

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

Here, the name of the prompt is *RunsFromDate*, which is defined in the report. The value for this prompt is specified in the *parameterValue* attribute. The value passed in this attribute is directly passed to the report.

The second method of specifying prompt values is by linking a user prompt to the report parameter. For example:

```
<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()%>"/>
```

Here the *promptId* attribute points to the *id* defined by the hidden HTML input tag. In this case, the value specified in the hidden field *Performance_Tab_NumberOfPerformers_Prompt* would be passed as the prompt value for the report parameter *NumberOfPerformers* while running the report.

Credentials

The reports comprising the Model Management page query the database underlying the repository for their content. Consequently, the reports need a data source corresponding to that database. This data source, *MMDDataSource*, is created within the repository when the user initially loads the Model Management page and is used whenever access to the repository database is required by any of the tags.

To access the *MMDDataSource* data source, the reports must specify valid credentials. The credential tag within the JavaServer pages allows the definition of these credentials.

```
<ibmspss-taglib:credential
    name="MMDDataSource"
    username="<<some db user name>>"
    password="<< password for the user>>"/>
```

The credentials for this data source are collected via *Login.jsp* before displaying the page and correspond to the username and password for the database underlying the repository. When valid credentials are obtained, the credentials are cached for the duration of the session and are used to run the reports. The Model Management reports are defined such that the name of the datasource is *MMDDataSource*.

In addition to the data source credentials, the Model Management reports also require credentials for the user executing the report.

```
<ibmspss-taglib:credential
    name="localhost"
    provider="<< some provider id >>"
    username="<< name of some CR user >>"
    password="<<password of the user >>" />
```

These credentials have the name *localhost*. Given that the repository may be configured to allow multiple security providers, the *provider* attribute is required.

The repositoryItem tag requires valid repository user credentials, which are specified in the *repositoryCredentialName* attribute of the tag. For Model Management, the value for this attribute is *localhost*, which corresponds to the user name, provider, and password of the user who has logged into IBM® SPSS® Collaboration and Deployment Services Deployment Portal.

Running visualization reports

The methodology to run visualization reports is identical to that used for BIRT Report Designer for IBM® SPSS® reports. However, the following usage differences should be considered:

- Visualization reports use a value of *ONDEMAND* for the *activate* attribute of the repositoryItem tag instead of the *ONLOAD* value used by BIRT Report Designer for IBM SPSS reports.

- Parameters required for the visualization reports are passed by the master BIRT Report Designer for IBM SPSS reports. For more information, see the topic Visualization report interactivity on p. 72.

Javascript API

The tag library has a framework built using JavaScript methods. These JavaScript methods provide both a sound validation framework and a handle to the user to run the reports on demand.

In order to run the reports on demand, the tag library provides a public API. This public API is available in the *reportTagLibPublicAPI.js* file within the *IBMSPSSTagLib.war*. The JavaScript file contains the following API:

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

For Model Management, this function is used to invoke the child reports for the master report.

For example, when the Scenario tab is visible, the Scenario report data is displayed. When the user clicks on the link for a scenario in the master report, the JavaScript method showDetails is called. This JavaScript method is embedded within the BIRT Report Designer for IBM® SPSS® report and indirectly calls the runRepositoryItem method to run two reports. One report is the *Scenario details* BIRT Report Designer for IBM SPSS report and the other is the *Scenario Comparison* visualization report.

If the *linkData* in the API call is null, the report runs with the data available within the JavaServer page supplied using the various IBM® SPSS® Collaboration and Deployment Services Tag Library tags. Just before calling runRepositoryItem, the Javascript code stores the parameter values to the html hidden control. The tag library framework picks up these values and passes them as parameters to the report being run.

The *targetID* fields correspond to the individual DIV ids where the report is to be rendered.

Visualization report interactivity

The *Performance vs. Scenario* graph generated by the visualization report for the Champions Tab supports interactivity. Whenever the user clicks on a bar in the graph, the details of the corresponding scenario are displayed in an adjacent area. The reports use the actionHandler and actionParameter tags to achieve this functionality.

Using the actionHandler tag is not necessarily required for visualization reports. Typically, the sourceLinkRepositoryItem tag would work just as well for visualization reports. However, in the case of the Model Management page, the visualization chart can occur multiple times on the page. The application needs special logic to be able to expand detail rows and to run the target reports with specific output locations. The actionHandler tags offer that additional level of control.

The section of the page that renders the *Performance vs. Scenario* Visualization report follows:

```
<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>
```

The repositoryItem tag gives details about the bar chart to be rendered. The nested actionHandler tag indicates that the JavaScript function selectCCScenario should be called whenever the *onClick* event occurs for the bars. The actionParameter tags nested within the actionHandler tag indicate that *filename*, *filepath*, *ccid*, and *equivalencekey* will be passed to the selectCCScerario function.

Each of these fields is defined within the visualization report XML. The definition for the *filename* variable is shown below:

```
<sourceVariable
categorical="true"
  id="filename"
  source="delimitedFileSource_430"
  sourceName="ct_filename">
```

This tag indicates that the column defined as *ct_filename* within the dataset will be used as *filename* by this report.

The JavaScript function selectCCScerario receives the id of the report on which the event occurred and an array of the parameter values. Internally, it calls runReport for dependent child reports and passes them the value array. For more information, see the topic Javascript API on p. 72.

URL fragments

The Model Management page displays some repository artifacts in an I-FRAME. These artifacts are the outputs generated by certain job runs.

An artifact is loaded by setting the source of the I-FRAME to the URL having the following format:

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

For more information, see the topic URL parameters in Chapter 3 on p. 6.

Tab extension framework

The navigation tabs of IBM® SPSS® Collaboration and Deployment Services Deployment Portal can be expanded to include custom entries using the Tab Extension framework. The Model Management functionality uses this framework to add an entry point into the Model Management page.

Figure 7-2
Model Management extension
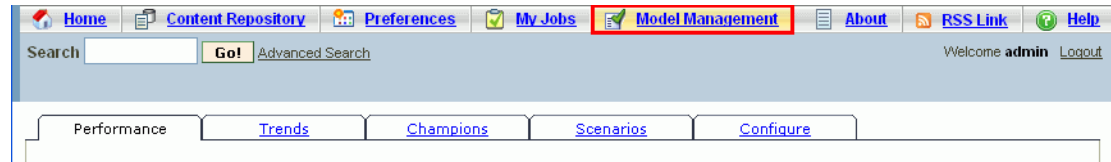
Deployment Portal reads extension files present in the following directory:

<installation-directory>\components\peb\extensions

These files are scanned to find all instances of the *peb-extension* elements. These elements will be individually rendered in the interface, provided the user credentials include any required actions. Any custom application must provide:

■   Extension XML file or an entry in an existing extension XML for the application

■   Appropriate entries in the localized text (*.tx*) file

The Model Management functionality is contained within the *peb-mmd.package* file in the *staging* directory of the repository installation. The package includes the file *mmd_extension.xml* in the *peb/extensions* directory. This XML file controls the appearance and functionality of the Model Management tab.

```
<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=someIcon.gif</tab-icon>
    <tab-position>2</tab-position>
    <tab-security>
      <capability>RunReport</capability>
      <capability>ViewModelManagementDashboard</capability>
    </tab-security>
  </peb-extension>
</file-viewer>
```

Elements defined within this file include:

■   The *tab-id* element which is the unique id for the tab. In this case it is *pebMmdTab*.

- The *tab-key* element which references the text appearing on the new tab. For localization purposes, Model Management isolates any localized text in XML files having *.tx* extensions. The key identifies the element in the localization file containing the text to be displayed. In this case, the *mmd/pebMmdTabTitle* key corresponds to the text *Model Management*.

- The *tab-url* element which specifies the URL invoked when the user clicks the tab. The URL can be either fully qualified (starting with a slash '/' character) or relative to the Deployment Portal application. In the latter case, the context is assumed to be *peb*. The link must point to a valid URI, with the URI location specified being the responsibility of the custom application. For Model Management, the link includes a reference to the war file *peb-mmd.war*.

- The *tab-security* element which identifies the actions required to access the tab. If the current user does not have these actions, the tab will not be displayed in the header JSP. Model Management requires the *RunReport* and *ViewModelManagementDashboard* actions.

Creating Custom Data Service Drivers

The IBM® SPSS® Collaboration and Deployment Services data service API provides public Java interfaces for implementing custom drivers to access non-standard data sources. For example, a custom driver may be necessary in cases when JDBC or ODBC access to a production database environment is not allowed because of security considerations. It may also be necessary for file-based data sources, cached data, or legacy data sources for which no JDBC or ODBC support exists. This functionality is used primarily by the scoring service.

This appendix provides and overview of the data service API, instructions for creating a custom driver, and a custom driver example included with IBM SPSS Collaboration and Deployment Services distribution.

Data service API

The API for implementing a custom driver is provided in *com.spss.data.service.datasource* package as a set of interfaces:

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

The package classes can be found in *data-service.jar* in *<repository installation directory>/staging/scoring.package*.

Initialization

DataService object is used to initialize one or more connector instances with the parameters that have been defined for the data source, such as the data source name and properties. The class implementing the *DataService* interface must use the default constructor.

Instances of the connector are cached over multiple events. It is acceptable to perform resource-intensive initialization and cache resources because the number of instantiated connectors will usually be low. There can be multiple instances for the same data source. It is recommended that system resources be release using the *terminate* method. A single instance of a connector must be able to handle multiple concurrent invocations. Each invocation creates its own *DataServiceRequest* object, which can be used for single data retrieval. This object can be queried only once. There is no need for the data service connector to cache the query results. The data service is invoked a minimum number of times for a single event. Note that circular links can

cause a table to be queried more than once. The data service connector should report errors using the *DataServiceException*. It is not allowed to throw any other exceptions.

Data handling

DataServiceSession object is instantiated by *DataService* during the handling of an event and the object can be used to maintain per-event session state. The session object then instantiates a *DataServiceRequest* object for each query using *createRequest* method with the following parameters:

- The name of the table.

- The names of the attributes (columns) for which the values should be returned

- The list of key-attribute names.

- Key tuples specified as a set of lists of objects. The order of objects in the lists matches the order of the attributes specified in the key attribute names.

A *DataServiceRequest* returns a list of records. Each record contains a list of attribute values. The list of attribute values must correspond to the list of attribute names that was passed to the *createRequest* method. Attribute values are returned as Java object instances. For each record returned, the set of values corresponding to attributes that are part of the key must match one of the key tuples. The following are the mappings between IBM® SPSS® Collaboration and Deployment Services Enterprise View data types and Java types:

IBM SPSS Collaboration and Deployment Services Enterprise View Data Type	Required Java Class
Integer	java.lang.Integer
Long	java.lang.Long
Double	java.lang.Double
Float	java.lang.Float
String	java.lang.String
Date	java.sql.Date
Daytime	java.sql.Time
Timestamp	java.sql.Timestamp
Decimal	java.math.BigDecimal
Boolean	java.lang.Boolean

To return a *Nil* object where value.isNil() = true, the null object must be returned.

Creating a custom driver

To create a custom driver using the data service API:

▶ Extract *data-server.jar* from *scoring.package* and make sure it is in the classpath.

▶ Write the Java source code implementing the data service interfaces and compile the driver classes.

▶ Create the driver package (as a JAR file).

▶ Deploy the package into *<repository installation directory>/components/data-provider* directory of IBM® SPSS® Collaboration and Deployment Services installation. For clustered installation, refer to the clustering section of the installation and configuration documentation.

▶ Depending on the application server, restart IBM SPSS Collaboration and Deployment Services.

To test the driver, create a corresponding data source in Deployment Manager, add it to the Enterprise View, and preview the data.

Custom driver example

IBM® SPSS® Collaboration and Deployment Services installation includes an example of custom driver. The Java source files for the example can be found in *<repository installation directory>/help/custom_driver*. They include:

■ *WSDataServiceConnector.java* (implements *com.spss.data.service.datasource.DataService*)

■ *WSDataServiceRequest.java* (implements *com.spss.data.service.datasource.DataServiceRequest*)

■ *WSDataServiceSession.java* (implements *com.spss.data.service.datasource.DataServiceSession*)

The example files also include *sample.jar* executable.

The driver retrieves the data from a text file *DATA_1* based on the value of the key specified in the data service data source. The file format is as follows:

```
data_key=1
data_element_0=This is the data of the first element
data_element_1=This is the data of the second element
```

To demonstrate the functionality of the driver:

1. Deploy *sample.jar*.

2. Create a data service data source with the following parameters:

■ **com.spss.rts.beil.ds.connector.WSDataServiceConnector** as the driver class name.

■ **DataLocation** driver property, pointing to the directory where the *DATA_1* data file is located.

Figure 8-1
Properties of the data service data source using the example driver
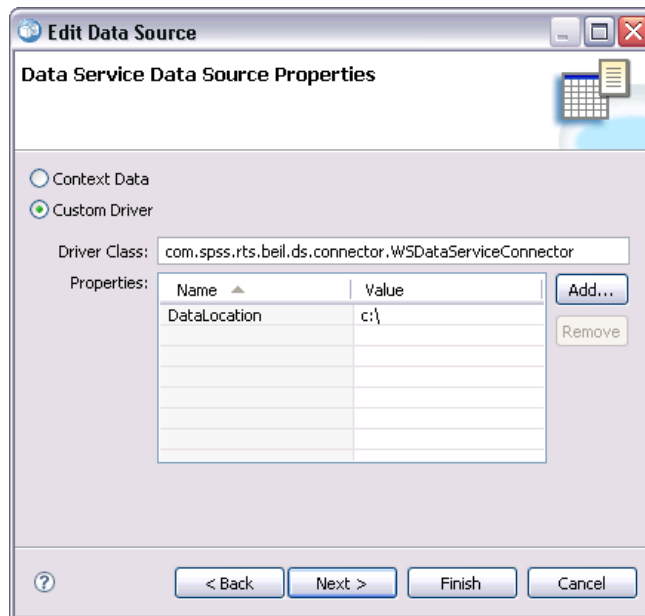
3. Define a table for the data source with the following columns.

- **data_element_0** (String)

- **data_element_1** (String)

- **data_key** (String, must be designated as the primary key)

Figure 8-2
Data service data source table properties
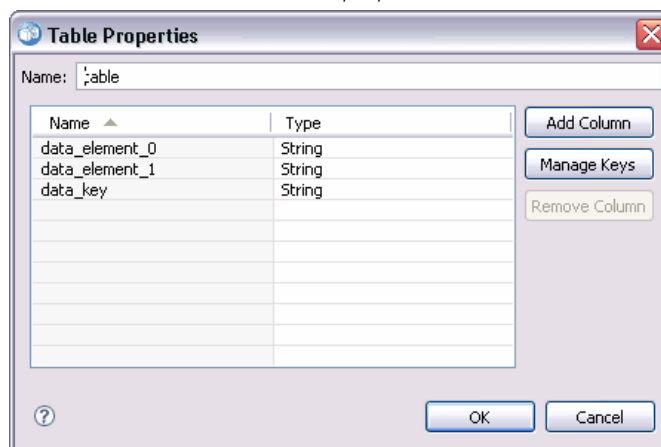
Figure 8-3
Data service data source table key
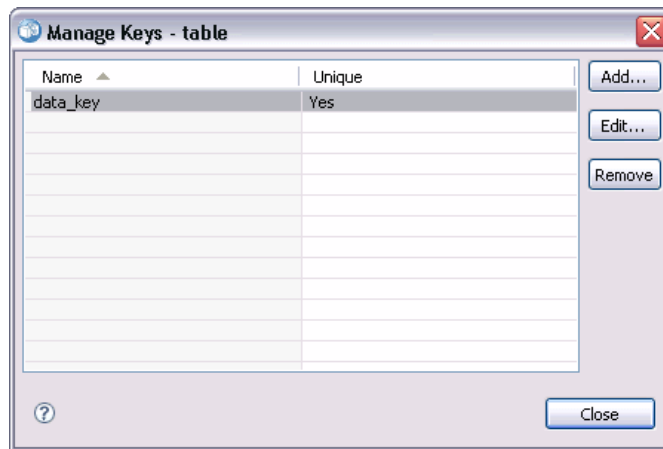
4. Create the *DATA_1* data file as shown and make sure it is in the location specified by the *DataLocation* property of the driver.

5. Create a Data Provider Definition-Real Time.

6. Using the Data Sets tab of the Data Provider Definition-Real Time editor, add the table defined in your data service data source to Data Provider Definition.

Figure 8-4
Custom driver data source table used in Data Provider Definition-Real Time

7. Preview the data in the table, specifying **1** as the key value. The results screen will display the contents of the data file corresponding to the key value.

Figure 8-5
Previewing the data in the custom driver data source
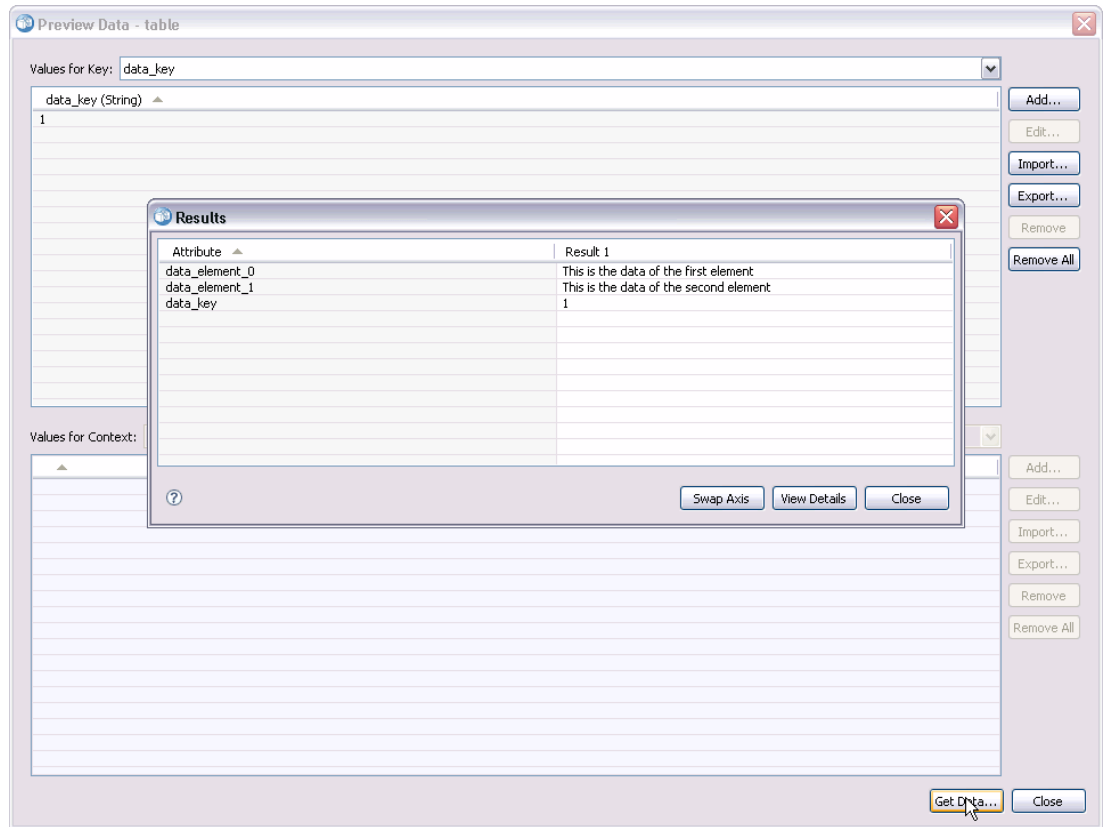
Notices

Licensed Materials – Property of SPSS Inc., an IBM Company. © Copyright SPSS Inc. 2004, 2010..

Patent No. 7,023,453

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: SPSS INC., AN IBM COMPANY, PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. SPSS Inc. may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-SPSS and non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this SPSS Inc. product and use of those Web sites is at your own risk.

When you send information to IBM or SPSS, you grant IBM and SPSS a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Information concerning non-SPSS products was obtained from the suppliers of those products, their published announcements or other publicly available sources. SPSS has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-SPSS products. Questions on the capabilities of non-SPSS products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to SPSS Inc., for the purposes of developing,

using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. SPSS Inc., therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. SPSS Inc. shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks of IBM Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at *http://www.ibm.com/legal/copytrade.shmtl*.

SPSS is a trademark SPSS and Showcase are trademarks of SPSS Inc., an IBM Company, registered in many jurisdictions worldwide.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other product and service names might be trademarks of IBM, SPSS, or other companies.

Adobe product screenshot(s) reprinted with permission from Adobe Systems Incorporated.

Microsoft product screenshot(s) reprinted with permission from Microsoft Corporation.

Index

actionHandler tag, 51, 72
actionParameter tag, 52, 72
actions, 1
activate attribute
   report tag, 70
   repositoryItem tag, 41
allowDownload parameter
   in URL queries, 23
allowPivoting parameter
   in URL queries, 23
allowPrinterFriendly parameter
   in URL queries, 23
authentication, 4

batch_type parameter
   in URL queries, 17
BIRT reports
   JSP samples, 58

class loader
   for custom applications, 34
   order, 34
   policy, 34
columnName attribute
   sourceLinkVariable tag, 51
contentType attribute
   of page directive, 31
cookies, 65
creating
   HTMLC files, 67
Credential bean, 54
credential tag, 38, 71
credentialDefinitionName attribute
   credential tag, 39
credentials, 1
custom dialogs, 1, 31
custom driver, 76–77
   example, 78
customizing
   Deployment Portal, 3

data service, 76
data service API, 76–77
data service data source, 78
data source, 76–77
dataset parameter
   in URL queries, 15
dataset_label parameter
   in URL queries, 16
dataset.prompt parameter
   in URL queries, 18

dataset_rowlimit parameter
   in URL queries, 16
dataset.search.criteria parameter
   in URL queries, 19
dataset.table parameter
   in URL queries, 18
dataset.uri parameter
   in URL queries, 17
dbcredential_datasourcename parameter
   in URL queries, 13
dbpwd_datasourcename parameter
   in URL queries, 13
dbuser_datasourcename parameter
   in URL queries, 13
Deployment Portal
   customizing, 3

emf files, 11
event attribute
   actionHandler tag, 51
Excel files, 11

format parameter
   in URL queries, 11
fragment parameter
   in URL queries, 12
function attribute
   actionHandler tag, 52

getBookmarkedValues function, 36
getValueJSFunction attribute
   sourceLinkPrompt tag, 48

height attribute
   repositoryItem tag, 43
height parameter
   in URL queries, 14
HTMLC files, 66
   creating, 67
   structure, 66

IBM SPSS Statistics custom dialogs, 1
IBM SPSS Statistics Data File Driver Service, 1
IBM SPSS Statistics server, 1
id parameter
   in URL queries, 7
inputURI attribute
   repositoryItem tag, 41

Java interfaces, 76–77

Index