Viewing Documents Using OpenDocument
Content

1  Document History ................................................................. 4

2  Getting started ................................................................. 5
   2.1 About this documentation ............................................. 5
   2.2 Who should use this documentation? ......................... 5
   2.3 About OpenDocument ............................................... 5
   2.4 What’s new in SAP BusinessObjects Business Intelligence platform 4.1 ........................ 6
   2.5 Migrating your links .................................................... 7
      Changes to the default URL path ................................ 7
      Deprecated parameters .............................................. 7

3  OpenDocument syntax ....................................................... 10
   3.1 Basic URL syntax ..................................................... 10
   3.2 URL syntax considerations ........................................ 10

4  Session management ....................................................... 13
   4.1 Serialized sessions .................................................. 13
   4.2 Logon tokens ......................................................... 14
   4.3 User sessions ....................................................... 15

5  Parameter reference ....................................................... 16
   5.1 Session management parameters ................................ 17
      serSes .............................................................. 17
      token ............................................................. 18
   5.2 Document identifier parameters .................................. 20
      iDocID ............................................................. 20
      sDocName ....................................................... 20
      sIDType .......................................................... 21
      sInstance ....................................................... 22
   5.3 Input parameters ...................................................... 22
      lsC[NAME] - complex variables ................................ 22
      lsI[NAME] - index ................................................ 23
      lsM[NAME] - multiple value variables ....................... 23
      lsR[NAME] - range prompts .................................... 24
      lsS[NAME] - single prompt ..................................... 25
      sRefresh ....................................................... 26
      sReportName .................................................. 26
5.4 Output parameters................................................................. 27
NAII......................................................................................... 27
sOutputFormat........................................................................ 28
sViewer.................................................................................. 28
noDocument............................................................................ 29

5.5 Product specific parameters or usage...................................... 29
Analysis, edition for OLAP........................................................ 29
Crystal Reports........................................................................ 35
Web Intelligence......................................................................... 40
eView and Information Spaces.................................................... 44
# 1 Document History

The following table provides an overview of the most important document changes.

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP BusinessObjects Business Intelligence platform 4.1</td>
<td>May, 2013</td>
<td>First release of this document.</td>
</tr>
</tbody>
</table>
| SAP BusinessObjects Business Intelligence platform 4.1 | January, 2015 | • In "Crystal Reports" section, updated the "lsM[NAME] - multiple value variables" sub section  
• In "Web Intelligence" section, updated the "lsM[NAME] - multiple value variables" sub section  
• In "Input parameters" section, updated the "lsM[NAME] - multiple value variables" sub section  
• In "sInstanceId" section, updated the "Description" column |
2 Getting started

2.1 About this documentation

This documentation provides you with information for constructing parameterized URLs with the OpenDocument syntax. OpenDocument URLs link to Business Intelligence (BI) documents in an SAP BusinessObjects Business Intelligence platform system. A parameter reference, including syntax and usage examples, is provided for each OpenDocument URL parameter.

For information about deploying the OpenDocument web application after the installation of the BI platform, see the SAP BusinessObjects Business Intelligence platform Web Application Deployment Guide.

2.2 Who should use this documentation?

This documentation is for anyone creating URLs to BI documents with the OpenDocument syntax. We recommend consulting this guide if you are:

- Providing end users with hyperlinks to a document through email or other direct means.
- Embedding hyperlinks in one document to another.
- Programmatically generating hyperlinks to documents in your custom application.

Familiarity with the management and organization of objects in your BI platform deployment is beneficial.

2.3 About OpenDocument

OpenDocument is one of many deployed web applications within a BI platform installation. It processes incoming URL requests for documents and any other viewable object type in the Central Management Server (CMS), and delivers the correct document to the end user in the appropriate viewer. This allows you to send users direct links to a document and avoid having them navigate through a folder hierarchy, such as in BI launch pad. The OpenDocument syntax and its parameters allow you to construct URLs that link to these documents. For example, consider the following URL:

IDDoc=Aa6GrrM79cRamaOSMgoaK1&sIDType=CUID

i Note
Replace <servername>:<port> with the name and port number of your web server where OpenDocument is deployed.
This URL accesses the object in the CMS with the CUID value of Aa6GrrM79cRAmaOSMGoadKI. If this is a Crystal report, for example, then the report is rendered to the user in a default SAP Crystal Reports viewer. In this example, iDocID is one of many URL parameters. These parameters specify how to access a particular document in the CMS, or determine how to display the document to the user.

You can link to many viewable object types with the OpenDocument syntax. Some examples include:

- Crystal reports
- Web Intelligence documents
- Analysis workspaces
- BI launch pad workspaces
- Dashboards objects (formerly Xcelsius)

Some of the designers for these BI document types provide GUI-based URL builders to help you embed openDocument URLs into your documents. Consult their respective product documentation for information on these features.

2.4 What's new in SAP BusinessObjects Business Intelligence platform 4.1

Expanded support for eView, Analysis for OLAP and Web Intelligence

A new section in this guide has been added to better represent new and expanded support for some OpenDoc parameters for the products as follows:

Analysis, edition for OLAP

- IsC - complex variables
- IsM - multiple value variables
- IsR - interval variables
- IsS - single value variables

Web Intelligence

- IsM - multiple value variables
- IsR - interval variables
- IsS - single value variables
2.5 Migrating your links

2.5.1 Changes to the default URL path

The default URL to the OpenDocument web application bundle has changed in SAP BusinessObjects Business Intelligence platform 4.0. New absolute OpenDocument links need to use the new default URL:

```html
?<parameter1>
&<parameter2>
&...
&<parameterN>
```

If you are migrating reports with existing links from an XI 3.x release platform, resolve the issue by setting up the following redirect in your web server:

- **Redirect**: `../OpenDocument/opendoc/openDocument.jsp`
- **To**: `../BOE/OpenDocument/opendoc/openDocument.jsp`

**Note**

Ensure that all URL request parameters are forwarded correctly by your redirect. Refer to your web server documentation for detailed steps on implementing a redirect.

**Note**

SAP BusinessObjects Business Intelligence platform 4.0 only supports a Java deployment of OpenDocument. The OpenDocument web bundle is part of the `BOE.war` file.

2.5.2 Deprecated parameters

This section lists deprecated and obsolete OpenDocument parameters as of SAP BusinessObjects Business Intelligence platform 4.0. Obsolete parameters are unsupported.

**Note**

Deprecated and obsolete members as of SAP BusinessObjects Enterprise XI 3.1 Service Packs are also listed for reference.
Table 5: Deprecated Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Replace with</th>
</tr>
</thead>
<tbody>
<tr>
<td>sIDType=GUID</td>
<td>Specifies that a GUID is used to specify the viewable document. Use in conjunction with iDocID.</td>
<td>Use sIDType=CUID instead.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Deprecated in SAP BusinessObjects Enterprise XI 3.1 SP3)</td>
<td></td>
</tr>
<tr>
<td>sIDType=RUID</td>
<td>Specifies that a RUID is used to specify the viewable document. Use in conjunction with iDocID.</td>
<td>Use sIDType=CUID instead.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Deprecated in SAP BusinessObjects Enterprise XI 3.1 SP3)</td>
<td></td>
</tr>
<tr>
<td>sKind</td>
<td>Specifies the SI_KIND property of the target Desktop Intelligence document.</td>
<td>Use iDocID instead.</td>
</tr>
<tr>
<td>sPath</td>
<td>The file path of the target document.</td>
<td>Use iDocID instead.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>sPath does not support the use of localized folder names which are available in this release. Legacy documents that use sPath to reference the correct folder name as stored in the CMS will continue to work while under deprecation. But it is recommended that you migrate your links to use the iDocID parameter instead.</td>
<td></td>
</tr>
<tr>
<td>sType</td>
<td>Specifies the file type of the target document.</td>
<td>Use iDocID instead.</td>
</tr>
<tr>
<td>sViewer=actx</td>
<td>Specifies the Crystal Reports ActiveX Viewer.</td>
<td>Use sViewer=html or sViewer=part instead. The ActiveX Viewer is deprecated as of this release.</td>
</tr>
<tr>
<td>sViewer=java</td>
<td>Specifies the Crystal Reports Java Applet Viewer.</td>
<td>Use sViewer=html or sViewer=part instead. The Java Applet Viewer is deprecated as of this release.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Replace with</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>sWindow</td>
<td>Indicates whether the target document will open in the current browser window or whether a new window will be launched.</td>
<td>Use the HTML anchor's <code>target</code> attribute or an equivalent. For example: <code>&lt;a href=&quot;...&quot; target=&quot;_blank&quot;&gt;...&lt;/a&gt;</code></td>
</tr>
</tbody>
</table>

**Note**

Obsolete as of SAP BusinessObjects Enterprise XI 3.1 SP3
3 OpenDocument syntax

3.1 Basic URL syntax

The basic syntax for an OpenDocument URL is as follows:

?<parameter1>
&amp;<parameter2>
&amp;...
&amp;<parameterN>

Replace the <platformSpecific> designation according to your SAP BusinessObjects Enterprise deployment as follows:

- For Java deployments, use openDocument.jsp
- For .NET deployments, use opendocument.aspx

### Note

Variables are denoted with angle brackets. You must substitute the proper value for these variables. For example, you must use the name of your BI platform server where OpenDocument is hosted in place of <servername> and you must use the correct port number in place of <port> to access the OpenDocument web application.

3.2 URL syntax considerations

**Accessing documents**

You must include the `iDocID` or `sDocName` parameter in your OpenDocument URL to specify the document to be viewed. Since there may be multiple documents in the Central Management Server (CMS) with the same name, and documents can be moved or renamed, it is recommended that you use `iDocID` to ensure uniqueness.

**Joining parameters**

Join parameters with the ampersand (`&`). Do not place spaces around the ampersand. For example:

```
sType=wid&sDocName=Sales2003
```

The ampersand is always required between parameters.
Spaces and special characters in parameter values

Because some browsers cannot interpret spaces, the parameters of the link cannot contain spaces or other special characters that require URL encoding. To avoid the misinterpretation of special characters, you can define a URL-encoded string in the source database to replace the special character with an escape sequence. This will allow the database to ignore the special character and correctly interpret the parameter value. Note that certain RDBMS have functions that allow you to replace one special character with another.

By creating an escape sequence for the plus sign (+), you can instruct the database to interpret the plus sign as a space. In this case, a document title Sales Report for 2003 would be specified in the DocName parameter as:
&$DocName=Sales+Report+for+2003&

This syntax prevents the database from misinterpreting the spaces in the title.

In addition, values for serialized sessions (using the serSes parameter) and logon tokens (using the token parameter) must be URL-encoded by your application before being passed to the OpenDocument URL string.

Trailing spaces in parameter values

Trim trailing spaces at the end of parameter values and prompt names. Do not replace them with a plus sign (+). The viewer may not know whether to interpret the plus sign (+) as part of the prompt name or as a space. For example, if the prompt name displays:

Select a City:_

(where _ represents a space), enter the following text in the link:

lsSSelect+a+City:=Paris

where the spaces within the prompt name are replaced with the plus sign, and the trailing space is trimmed off.

Capitalization

All of the OpenDocument parameters and parameter values are case sensitive.

URL length limit

OpenDocument may add characters to your URL when it redirects to the requested document; however, encoded URLs cannot exceed the maximum character limit for the supported browsers. For example, certain versions of Internet Explorer limit the URL length to 2083 characters. Therefore, know the browser character limit to ensure your URL will be within the maximum limit.
Parameter values in links to sub-reports

You cannot pass parameter values to a sub-report of a target Crystal report.

Opening a new window

To force OpenDocument HTML links to open a new browser window, use the HTML anchor's `target` attribute or an equivalent. For example:

```html
```
4 Session management

Normally when using an OpenDocument link to access documents secured in the BI platform, the user will be
prompted for credentials. OpenDocument provides two parameters to avoid having the user prompted for their
username and password information. You can either insert a serialized session or a logon token directly into the
OpenDocument URL. This gives you control over the duration of the access to the document. OpenDocument
URLs can be set to different languages.

4.1 Serialized sessions

Serialized sessions can be used in OpenDocument by inserting the `serSes` parameter into the OpenDocument
URL. This allows users to access files without being prompted for credentials. Creating a serialized sessions does
not use up an additional licence. Serialized sessions expire if the original user’s session times out or logs off.

Example

The following example uses the BI platform Java SDK to pass in a serialized session to the OpenDocument URL.
For more information on the `IEnterpriseSession.getSerializedSession` method, see the SAP
BusinessObjects Business Intelligence platform Java API Reference.

```java
String openDocumentSerSes() throws SDKException, UnsupportedEncodingException {
    IEnterpriseSession sess = CrystalEnterprise.getSessionMgr().logon
        ("username",
            "password",
            "<cms>:<port>",
            "secEnterprise");
    String serSession = sess.getSerializedSession();
    String serSesEncode = URLEncoder.encode(serSession, "UTF-8");
            ?iDocID=Aa6GrrM79cRAmaOSMGoadKI
            &sIDType=CUID
            &serSes=" + serSesEncode
        ;
}
```

Note

- Replace `<server>:<port>` with the server name and port number of your web server.
- Replace `<cms>:<port>` with the Central Management Server (CMS) name and port number.
- You must URL-encode the serialized session.
- Since an OpenDocument URL with a serialized session contains the user session, they must not be
  shared for security reasons.
4.2 Logon tokens

Logon tokens can be used in OpenDocument by inserting the token parameter into the OpenDocument URL. Logon tokens allow users access to files secured in the BI platform without being prompted for credentials, while also giving you control on the duration of the access to the file. Creating a new logon token uses up an additional licence.

Example

Using the BI platform Java SDK

The following example uses the BI platform Java SDK to pass in a logon token to the OpenDocument URL. For more information on the ILogonTokenMgr.createLogonToken method, see the SAP BusinessObjects Business Intelligence platform Java API Reference.

```java
String openDocumentToken() throws SDKException, UnsupportedEncodingException {
    IEnterpriseSession sess = CrystalEnterprise.getSessionMgr().logon
        ("username",
         "password",
         "<cms>:</port>",
         "secEnterprise";
    String token = sess.getLogonTokenMgr().createLogonToken
        ("",
         120,
         100
    );
    String tokenEncode = URLEncoder.encode
        (token,
         "UTF-8"
    );
    sess.logoff();
    return
         ?iDocID=Aa6GrY79cRAmaOSMqadKI
         &sIDType=CUID
         &token=" + tokenEncode);
}
```

Note
- Replace `<server>:<port>` with the server name and port number of your web server.
- Replace `<cms>:<port>` with the Central Management Server (CMS) name and port number.
- The createLogonToken method allows you to specify the machine that can use the token (which can be empty to allow any user to use the token), the number of minutes the token is valid for, and the number of logons that the token can be used for as parameters. Since the newly created logon token consumes an additional session, sess.logoff is called to logoff the original session.
- Since an OpenDocument URL with a logon token contains the user session, they must not be shared for security reasons.
**Example**

**Using the BI platform RESTful Web Services SDK**

The following example passes token fetched using the BI platform RESTful Web Services SDK to the OpenDocument URL. For more information see the *SAP BusinessObjects Business Intelligence platform RESTful Web Service Developer guide*.

```plaintext
?idDocID=<documentID>
&sIDType=CUID
&token=<logonToken>
```

**Note**

- A URL-encoded logon token may contain a large number of characters. Some web browsers may limit the number of characters that are allowed in a URL.
- Replace `<server>:<port>` with the server name and port number of your web server.
- Replace `<documentID>` with the ID of the document to retrieve.
- Replace `<logonToken>` with the URL-encoded logon token value retrieved using the BI platform RESTful Web Services.

4.3 **User sessions**

When OpenDocument is used from BI launch pad or the CMC, it will access the current user session and the user does not need to enter credentials. When a document is viewed using an OpenDocument URL, the user will be prompted for credentials except in the following cases:

- Vintela or Siteminder SSO is configured for the deployed OpenDocument web application.
- The OpenDocument URL uses a `serSes` or `token` parameter.
- The OpenDocument application has an existing user session for that browser session.

If the existing session is different than the session in the `serSes` or `token` parameter, the existing session will be closed and a new session will be created. That is, you can use `serSes` or `token` parameter to over-ride an existing user session. The OpenDocument application will look for an existing user session in the Web application session and in cookies.

**Note**

Only one OpenDocument session can be created from a single browser session.

If the new `serSes` or `token` parameter is incorrect and there is an existing user session, OpenDocument will attempt to open the document using the current user session. If it can’t it will then prompt the user for credentials.
5 Parameter reference

This section provides details about the available OpenDocument parameters, their specific uses, and relevant examples.

Note
The document to which an OpenDocument link points to is referred to as the target document.

Table 7: Session Management Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serSes</td>
<td>Specifies a valid serialized Enterprise session.</td>
</tr>
<tr>
<td>token</td>
<td>Specifies a valid logon token for the current Enterprise session.</td>
</tr>
</tbody>
</table>

Table 8: Document Identifier Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>iDocID</td>
<td>Specifies the unique identifier of the viewable document in the CMS. Use in conjunction with sIDType.</td>
</tr>
<tr>
<td>sDocName</td>
<td>Specifies the name of the viewable document in the CMS.</td>
</tr>
<tr>
<td>sIDType</td>
<td>Specifies the type of object identifier used to specify the viewable document. Use in conjunction with iDocID.</td>
</tr>
<tr>
<td>sInstance</td>
<td>Specifies the scheduled instance of the target document to open. Use in conjunction with sDocName or iDocID.</td>
</tr>
</tbody>
</table>

Table 9: Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsC[NAME] - complex variables</td>
<td>Specifies a contextual prompt for Web Intelligence documents if there is an ambiguity during SQL generation.</td>
</tr>
<tr>
<td>lsI[NAME] - index</td>
<td>Specifies index or key values for a prompt. [NAME] is the text of the prompt.</td>
</tr>
<tr>
<td>lsM[NAME] - multiple value variables</td>
<td>Specifies multiple values for a prompt. [NAME] is the text of the prompt.</td>
</tr>
<tr>
<td>lsR[NAME] - range prompts</td>
<td>Specifies a range of values for a prompt. [NAME] is the text of the prompt.</td>
</tr>
<tr>
<td>lsS[NAME] - single prompt</td>
<td>Specifies a value for a single prompt. [NAME] is the text of the prompt.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sPartContext [page 37]</td>
<td>Specifies the data context of a Crystal report part. Use in conjunction with sReportPart.</td>
</tr>
<tr>
<td>sRefresh [page 26]</td>
<td>Indicates whether a database refresh should be forced when the target document is opened.</td>
</tr>
<tr>
<td>sReportMode [page 38]</td>
<td>Indicates whether the link should open the full target Crystal report or just the report part specified in.</td>
</tr>
<tr>
<td>sReportName [page 26]</td>
<td>Specifies the report to open if the target document contains multiple reports.</td>
</tr>
<tr>
<td>sReportPart [page 39]</td>
<td>Specifies the part of the target Crystal report to open.</td>
</tr>
</tbody>
</table>

### Table 10: Output Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAII [page 27]</td>
<td>Indicates whether to force the display of the prompt selection page for Interactive Analysis Desktop prompts.</td>
</tr>
<tr>
<td>sOutputFormat [page 28]</td>
<td>Specifies the format in which to open the target document.</td>
</tr>
<tr>
<td>sViewer [page 28]</td>
<td>Specifies the selected report viewer.</td>
</tr>
<tr>
<td>noDocument [page 29]</td>
<td>Used with Web Intelligence reports, a value of true automatically forces a report to open in design mode.</td>
</tr>
</tbody>
</table>

### 5.1 Session management parameters

#### 5.1.1 serSes

**Table 11:**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>serSes</td>
<td>Specifies a valid serialized Enterprise session.</td>
<td>A serialized string representing the current Enterprise session.</td>
</tr>
</tbody>
</table>

Contains a serialized session of the current user session. This can be entered into an OpenDocument URL to allow users to access files without being prompted for credentials. Creating a serialized sessions does not use up an additional licence. Serialized sessions expire if the original user’s session times out or logs off.

**Example**

The following example uses the BI platform Java SDK to pass in a serialized session to the OpenDocument URL. For more information on the IEnterpriseSession.getSerializedSession method, see the SAP...
BusinessObjects Business Intelligence platform Java API Reference. You can retrieve a serialized session in a similar fashion using other BI platform SDKs such as .NET and Web Services.

```java
String openDocumentSerSes() throws SDKException, UnsupportedEncodingException {
    IEnterpriseSession sess = CrystalEnterprise.getSessionMgr().logon
        ("username", "password", "<cms>:<port>", "secEnterprise");
    String serSession = sess.getSerializedSession();
    String serSesEncode = URLEncoder.encode(serSession, "UTF-8");
    return
            ?iDocID=Aa6GrrM79cRAmaOSMGoadKI
            &sIDType=CUID
            &serSes=" + serSesEncode
        ;
}
```

**Note**

- Replace `<server>:<port>` with the server name and port number of your web server.
- Replace `<cms>:<port>` with the Central Management Server (CMS) name and port number.
- You must URL-encode the serialized session.
- Since an OpenDocument URL with a serialized session contains the user session, they must not be shared for security reasons.

### 5.1.2 token

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>token</td>
<td>Specifies a valid logon token for the current Enterprise session.</td>
<td>The logon token for the current Enterprise session.</td>
</tr>
</tbody>
</table>

Contains the logon token for the current user. This can be entered into an OpenDocument URL to allow users to access files without being prompted for credentials. Creating a new logon token uses up an additional licence.

**Example**

**Using the BI platform Java SDK**

The following example uses the BI platform Java SDK to pass in a logon token to the OpenDocument URL. For more information on the `ILogonTokenMgr.createLogonToken` method, see the SAP BusinessObjects Business Intelligence platform Java API Reference. You can create logon tokens in a similar fashion using other BI platform SDKs such as .NET and Web Services.

```java
String openDocumentToken() throws SDKException, UnsupportedEncodingException {
    return
            ?iDocID=Aa6GrrM79cRAmaOSMGoadKI
            &sIDType=CUID
            &serSes=" + serSesEncode
        ;
}
```
```java
IEnterpriseSession sess = CrystalEnterprise.getSessionMgr().logon
("username",
"password",
"<cms>:<port>",
"secEnterprise"
);
String token = sess.getLogonTokenMgr().createLogonToken
("", 120, 100);
String tokenEncode = URLEncoder.encode(token, "UTF-8");
return
?iDocID=Aa6GrrM79cRamaOSMgadKI
&sIDType=CUID
&token=" + tokenEncode
);
```

**Note**
- Replace `<server>:<port>` with the server name and port number of your web server.
- Replace `<cms>:<port>` with the Central Management Server (CMS) name and port number.
- The `createLogonToken` method allows you to specify the machine that can use the token (which can be empty to allow any user to use the token), the number of minutes the token is valid for, and the number of logons that the token can be used for as parameters.
- Since an OpenDocument URL with a logon token contains the user session, they must not be shared for security reasons.

**Example**

**Using the BI platform RESTful Web Services SDK**

The following example passes token fetched using the BI platform RESTful Web Services SDK to the OpenDocument URL. For more information see the [SAP BusinessObjects Business Intelligence platform RESTful Web Service Developer guide](http://<server>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=<documentID>
&sIDType=CUID
&token=<logonToken>

**Note**
- A URL-encoded logon token may contain a large number of characters. Some web browsers may limit the number of characters that are allowed in a URL.
- Replace `<server>:<port>` with the server name and port number of your web server.
- Replace `<documentID>` with the ID of the document to retrieve.
- Replace `<logonToken>` with the URL-encoded logon token value retrieved using the BI platform RESTful Web Services.
5.2 Document identifier parameters

5.2.1 iDocID

Table 13:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>iDocID</td>
<td>Specifies the unique identifier of the viewable document in the CMS. Use in conjunction with sIDType.</td>
<td>A numerical identifier associated with the document in the CMS.</td>
</tr>
</tbody>
</table>

You must include the iDocID or sDocName parameter in your OpenDocument URL to specify the document to be viewed. Since there may be multiple documents in the CMS with the same name, it is recommended that you use iDocID to ensure uniqueness.

You can see identifier values for a document within the Central Management Console (CMC) or BI launch pad applications. The properties page for each document contains the document ID and the CUID. You can also obtain the identifier programmatically using the BI platform SDK. For example, in the Java SDK the com.crystaldecisions.sdk.occa.infostore.IInfoObject interface contains getID and getCUID methods which you can pass to an OpenDocument URL.

Note

If you pass in an InfoObject ID rather than a CUID, you do not need to specify the sIDType parameter. However, InfoObject IDs are changed when migrating documents from one CMS to another. It is recommended that the CUID be used, which is preserved during migration.

Example


5.2.2 sDocName

Table 14:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>sDocName</td>
<td>Specifies the name of the viewable document in the CMS.</td>
<td>The title of the document in the CMS.</td>
</tr>
</tbody>
</table>
You must include the `iDocID` or `sDocName` parameter in your OpenDocument URL to specify the document to be viewed. Since there may be multiple documents in the CMS with the same name, and documents can be moved or renamed, it is recommended that you use `iDocID` to ensure uniqueness.

![Note]

`sDocName` does not support the use of localized document names. Legacy documents that use `sDocName` to reference the correct document name as stored in the CMS will continue to work. But it is recommended that you use the `iDocID` parameter instead.

### Example

```plaintext
?sDocName=Sales+in+2003
```

### 5.2.3 `sIDType`

#### Table 15:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
</table>
| `sIDType` | Specifies the type of object identifier used to specify the viewable document. Use in conjunction with `iDocID`. | • InfoObjectID  
• ParentID  
• CUID |

![Note]

If you pass in an InfoObject ID as a value to `iDocID` rather than a CUID, you do not need to specify the `sIDType` parameter. However, InfoObject IDs are changed when migrating documents from one CMS to another. It is recommended that the CUID be used, which is preserved during migration.

### Example

```plaintext
?iDocID=Aa6GrrM79cRAmaOSMGoadKI  
&sIDType=CUID
```
### 5.2.4 sInstance

Table 16:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
</table>
| sInstance  | Specifies the scheduled instance (created during scheduling) of the target document to open. Use in conjunction with sDocName or iDocID. | • User (Latest instance owned by current user)  
• Last (Latest instance of the document)  
• Param (Latest instance of the document with matching parameter values. Crystal reports and Web Intelligence documents only.) |

**Example**

?sDocName=Sales+in+2003
&sInstance=User

### 5.3 Input parameters

#### 5.3.1 lsC[NAME] - complex variables

The lsC parameter specifies the use of different operators when defining complex variable values. Semicolons are used to separate different conditions.

Table 17:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsC[NAME]</td>
<td>Specifies a contextual prompt if there is an ambiguity during SQL generation.</td>
<td>A prompt value that resolves the ambiguity in the SQL generation.</td>
</tr>
</tbody>
</table>

**Example**

iDocID=Aa6GrM79cRAmOaSMGoadKI
&sIDType=CUID
&lsC=Sales
5.3.2  *lsI*[NAME] - *index*

The *lsI* parameter specifies the use of an index or key value.

Table 18:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>lsI</em>[NAME]</td>
<td>Specifies index or key value. This parameter must be associated with one of the parameters <em>lsS</em>[NAME] - single prompt [page 25], <em>lsM</em>[NAME] - multiple value variables [page 23] or <em>lsR</em>[NAME] - range prompts [page 24].</td>
<td>Value can be simple [S], multiple [M] or a range [R] according prompt type [prompt name]=[caption] or, the case of Webi reports [prompt name]=[key/index].</td>
</tr>
</tbody>
</table>

**Example**

?sDoc=IndexTest
&type=wid
&lsMStore=[caption]
&lsIStore=[index]

Using eFashion sample Universe passing a value for the "Store name" object which has been modified to be Index Aware as well as the Index Value for the "Store name" object as follows

?sDoc=IndexTest
&type=wid
&lsMStore=e-Fashion New York Magnolia
&lsIStore=2

**Example**

the *lsI* parameter to provide index values such as a keydate. The parameters are passed using the technical name of the variable as set up in the BEx Query Designer. Note that URL encoding is required.

The following example identifies a data connection [2], the variable's technical name DT_IH, and the date value 20120715.

&lsI[2]DT_IH=20120715

5.3.3  *lsM*[NAME] - *multiple value variables*

The *lsM* parameter allows the use of multiple values in a hierarchy node.
Table 19:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsM[NAME]</td>
<td>Specifies multiple values for a prompt. [NAME] is the text of the prompt.</td>
<td>• Web Intelligence: Use semicolon(;) to separate multiple prompt values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Crystal Reports: Use comma(,) to separate multiple prompt values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• no_value (only for optional parameters)</td>
</tr>
</tbody>
</table>

### Note
You can remove an optional parameter from the prompt by setting it to no_value in the openDocument query string. If you leave an optional parameter out of the openDocument query string, a default parameter value will be applied.

### Example

```
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sRefresh=Y
&lsMSelect+Cities=[Paris],[London]
```

```
&id=ID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sRefresh=Y
&lsMparamStringDR=c,d
&lsMparamNumberDR=3,4
&lsMparamDateTimeDR=[Date(2003,6,3)], [Date(2003,6,4)]
&lsMparamDateTimeDR=[DateTime(2003,6,1,3,1,1)], [DateTime(2003,6,1,4,1,1)]
```

5.3.4  \( lsR[NAME] \) - range prompts

The \( lsR \) parameter allows a range to be specified.

Table 20:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsR[NAME]</td>
<td>Specifies a range of values for a prompt. [NAME] is the text of the prompt.</td>
<td>• A range of values for the prompt. separated by a double period (..).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• no_value (only for optional parameters)</td>
</tr>
</tbody>
</table>
Note
You can remove an optional parameter from the prompt by setting it to `no_value` in the OpenDocument query string. If you leave an optional parameter out of the OpenDocument query string, a default parameter value will be applied.

Example

?iDocID=Aa6GrrM79cRamaOSMGoadKI
&sIDType=CUID
&sRefresh=Y
&lsRTime+Period:=[2000..2004)

?iDocID=Aa6GrrM79cRamaOSMGoadKI
&sIDType=CUID
&sRefresh=Y
&lsRparamStringDR=[h..i]
&lsRparamNumberDR=[7..8]
&lsRparamCurrencyDR=[3..4]
&lsRparamDateTimeDR=[DateTime(2003,6,1,7,1,1)..DateTime(2003,6,1,8,1,1)]
&lsRparamTimeDR=[Time(1,1,7)..Time(1,1,8)]
&lsRparamUnbound1=(..6)
&lsRparamUnbound2=[6..)
&lsRparamStringR=[a..d]
&lsRparamNumberR=[1..3]
&lsRparamCurrencyR=[1..3]
&lsRparamDateR=[Date(2003,6,1)..Date(2003,6,3)]
&lsRparamDateTimeR=[DateTime(2003,6,1,1,1,1)..DateTime(2003,6,1,3,1,1)]
&lsRparamTimeR=[Time(1,1,1)..Time(3,1,1)]

5.3.5 `lsS[NAME]` - single prompt

Table 21:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsS[NAME]</td>
<td>Specifies a value for a single prompt. [NAME] is the text of the prompt.</td>
<td>● A single prompt value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● <code>no_value</code> (only for optional parameters)</td>
</tr>
</tbody>
</table>
### 5.3.6 sRefresh

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>sRefresh</td>
<td>Indicates whether a database refresh should be forced when the target document is opened.</td>
<td>Y, N</td>
</tr>
</tbody>
</table>

Certain documents can contain saved settings to specify that a database refresh must occur when the document is opened in a viewer. These document settings will override `sRefresh=N`.

**Example**

```text
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sRefresh=Y
```

### 5.3.7 sReportName

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>sReportName</td>
<td>Specifies the report to open if the target document contains multiple reports.</td>
<td>The report name for Web Intelligence documents and page name for A-OLAP Intelligence reports.</td>
</tr>
</tbody>
</table>

**Note**

Defaults to the first report if this parameter is not specified.
5.4 Output parameters

5.4.1 NAII

The NAII out parameter is specific to Web Intelligence that allows you to pass a Y or N flag to display the prompt selection page.

Table 24:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAII</td>
<td>Indicates whether to force the display of the prompt selection page.</td>
<td>Y (prompt values that are passed with lsS, lsM, or lsR in the URL are applied and not displayed in the Prompts dialog box)</td>
</tr>
</tbody>
</table>

Note

- NAII= Y raises the Prompts dialog box for any values not specified in the URL. Prompts created with default values are still displayed in the Prompts dialog box.
- If all prompt values are specified in the URL, the prompt window does not appear even if NAII= Y is specified.

Example

This example assumes there are two prompts in the Web Intelligence document: Year and Country. NAII= Y forces the Prompts dialog box to appear and allows the user to specify a value for the Country prompt. The Year prompt is already set to a value of FY1999 in the URL using the lsS parameter and therefore is not prompted for.


?iDocID=Aa6GrrM79cRamaOSMGoadKI&sIDType=CUID
&lsSYear=FY1999
&NAII=Y
&sRefresh=Y
### 5.4.2 sOutputFormat

Table 25:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
</table>
| sOutputFormat | Specifies the format in which to open the target document. | - H (HTML)  
- P (PDF)  
- E (Microsoft Excel (97-2003) - Crystal reports only)  
- W (Rich Text Format (RTF) - Crystal reports only) |

**Note**

Defaults to HTML if this parameter is not specified.

**Example**

?iDocID=Aa6GrrM79cRAmaOSMGoadKI  
&sIDType=CUID  
&sOutputFormat=E

### 5.4.3 sViewer

Table 26:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
</table>
| sViewer | Specifies the selected report viewer. | - html  
- part (Crystal reports only) |

**Example**

?iDocID=Aa6GrrM79cRAmaOSMGoadKI  
&sIDType=CUID  
&sViewer=html
5.4.4  noDocument

Table 27:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>noDocument</td>
<td>A value of true forces a report to open in design mode using the existing report template.</td>
<td>Boolean value: true</td>
</tr>
</tbody>
</table>

**Note**

- noDocument=true automatically forces a Web Intelligence report into design mode.
- Since the existing report template is used, you can prevent overwriting this template by applying the appropriate security.

**Example**


5.5  Product specific parameters or usage

5.5.1  Analysis, edition for OLAP

**Obtaining the base URL of an Analysis document**

In Analysis, obtain the base URL of the document using the Send To > Document Link. To verify the link works, you can copy the value in the Link field to a new web browser window.

**Building an Analysis Open Document URL**

To specify values for an SAP BW variable in the URL, the following elements are added to the end of the URL:

- the appropriate OpenDocument parameters for the type of variable that you want to specify
- the technical names of the variable as defined in the BEx Query Designer
Where to find Technical Names

Technical Names for SAP BW variables are specified in the SAP NetWeaver Business Explorer Query Designer "Properties" pane, which are made visible by turning on the View > Technical Names setting, and viewing the "Variable Sequence" for the selected query. A technical name has no spaces and uses upper case letters, for example Z_SHPDATE, while the display name might read Shipment Date.

The following is an example of adding technical names such as the variable `Z_VAR01` and a date `20120619` value to the end of a URL:

```
&lsSZ_VAR01==20120619
```

URL encoding of special characters

**Note**

Encode the URL if the link fails for some recipients due to the presence of commas or other special characters. For example

- `&lsC[1]Z_VAR06==1;[]5,20;[]10,15` would be encoded to look like
  - `%26amp%3BlsC%5B1%5DZ_VAR06%3D%3D1%3B%5B%5D%2C20%3B%5B%5D%10%2C15`

Multiple data connection setup

Where multiple connections are defined and must be included in the URL, an index number that represents the order in which that connection appears can be used to identify which connection the following variable names and values should be associated with.

The following example shows several OpenDocument tags, showing the data connection index number and the technical names of the variables and their values.

```html
?sIDType=CUID
&iDocID=<ID>
&lsS[1]Z_VAR01=COUNTRY_HIERARCHY_01
&lsS[1]Z_VAR04=EUROPE
&lsS[2]Z_VAR02=20111111
&lsS[2]Z_VAR03=SALES_HIERARCHY_03
&lsM[3]Z_VAR05=1,3,5
&lsC[4]Z_VAR06==1;[]5,20;[]10,15
```

Multiple connections are differentiated from one another by an index number. The sequence in which these connections appear can be viewed in the Analysis Data pane. In the preceding example, the first connection listed is identified as 1, the next connection by a 2 and so on. The fourth [4] connection appears in `&lsC[4]Z_VAR06`. 

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Viewing Documents Using OpenDocument Parameter reference
The first connection \texttt{lsS[1]} includes a single \texttt{COUNTRY\_HIERARCHY\_01} value applied to the \texttt{Z\_VAR01} variable, and a single \texttt{EUROPE} value applied to the \texttt{Z\_VAR04} variable.

The second connection \texttt{lsS[2]} includes a single date \texttt{(2011, 11, 11)} value applied to the \texttt{Z\_VAR02} variable, and a single \texttt{SALES\_HIERARCHY\_03} value applied to the \texttt{Z\_VAR03} variable.

The third connection \texttt{lsS[3]} includes multiple values 1, 2, and 3, which are associated to the \texttt{Z\_VAR05} variable.

The fourth connection \texttt{lsS[4]} includes complex variable values applied to \texttt{Z\_VAR06}, such as 1, and a range from 5 to 20, while values in the range 10 to 15 are excluded.

### OpenDoc syntax errors

Incorrect syntax will generate errors that usually relate to missing variables, operators and unencoded special characters among the few examples.

<table>
<thead>
<tr>
<th>Examples of OpenDocument syntax error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenDoc syntax error. The value &quot;&lt;value&gt;&quot; for variable &lt;variable name&gt; does not contain sufficient information</td>
<td>This error is generated when a complex variable type is incomplete, for example, if a bracket &quot;[&quot; is missing.</td>
</tr>
<tr>
<td>OpenDoc syntax error. The value &quot;&lt;value&gt;&quot; for variable &lt;variable name&gt; does not contain a supported operation</td>
<td>This error is generated when the operator for complex variable type does not match any of the supported operators, for example &quot;]&quot;. If the value is invalid, an error prompt or dialog box will appear in Analysis, edition for OLAP.</td>
</tr>
</tbody>
</table>

### Open Document tags used by Analysis, edition for OLAP

- **\texttt{lsC\{}NAME\}\texttt{\}} - complex variables**

  The following example shows a typical use of the \texttt{lsC} tag to include complex variables. The technical name of the variable is used along with the appropriate parameter in the expected format.
**Example**

**Date variable example**

The following example supplies the technical name of a date variable Z_VAR01, the is-equal-to operator ==, and the numerical date value in year (YYYY), month (MM) and day (DD) format.

```
?sIDType=CUID
&iDocID=AeGNibIUS.1Nmv45dz3jeP4
&lsC[Z_VAR01]==20120619
```

**Example**

**Multiple values including a range and data to exclude**

The following example identifies the first connection [1], and the complex variable name Z_VAR06. The range is from 5 to 20, excluding the range between 10 to 15. Semicolons separate the values.

```
?sIDType=CUID
&iDocID=<ID>
&lsC[1]Z_VAR06==1;[5,20];![](10,15)
```

**Note**

The preceding code snippets show each OpenDocument parameter on its own line to more clearly denote each variable name and value. Be sure to assemble the completed URL as one unbroken string.

The following table represents various operators that can be used with the lsC parameter for Analysis URLs.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>[]</td>
<td>within range</td>
<td>lsC_VAR01=[5,10]</td>
</tr>
<tr>
<td>![]</td>
<td>outside of range</td>
<td>lsC_VAR01=![]5,10</td>
</tr>
<tr>
<td>==</td>
<td>equal</td>
<td>lsC_VAR01==5</td>
</tr>
<tr>
<td>!=</td>
<td>not equal</td>
<td>lsC_VAR01!=5</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than 5</td>
<td>lsC_VAR01&gt;5</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than 5</td>
<td>lsC_VAR01=&lt;5</td>
</tr>
<tr>
<td>&gt;=</td>
<td>equal or greater than</td>
<td>lsC_VAR01&gt;=5</td>
</tr>
<tr>
<td>&lt;=</td>
<td>equal or less than</td>
<td>lsC_VAR01&lt;=5</td>
</tr>
</tbody>
</table>
5.5.1.2 \textit{lsM[NAME]} - multiple value variables

Setting parameters for multiple values with a single data connection

If the target document is an Analysis report you can use the \textit{lsM} parameter to provide multiple values, each separated by commas. Because some operating systems have difficulty with commas in URL strings, URL encoding is required.

Parameters that are passed use the unique technical names as they have been set up in the data sources for the Analysis workspace.

\textbf{Example}

Setting multiple variables for a sales hierarchy

This example shows how multiple variables can be specified. The first data connection is used (so it does not have to be specified after \textit{&lsM}), followed by the variable’s technical name \texttt{VAR\_HN\_S} and values of 1 and 3 separated by commas. For clarity, the example is shown not URL-encoded, and separate openDocument components appear on their own line.

\begin{verbatim}
?siDType=CUID
&iDocID=<ID>
&lsMVAR\_HN\_S=1,3
\end{verbatim}

5.5.1.3 \textit{lsR [NAME]} - interval variables

If the target is an Analysis report, a variable range can be specified as well as the index number of each data connection if there is more than one. The following example demonstrates a data connection and a variable name followed by the range.

(Note that the openDocument preamble is omitted for clarity):

\begin{verbatim}
\end{verbatim}

Where [2] refers to the name of the connection index number (if there are two connections, the number in this example refers to the second connection as it appears in the Data panel in the Analysis client). \texttt{VAR\_20} is the technical name of the field as it appears in the BEx Query Designer (note that the technical name is different from the user friendly name Ship\_date, which cannot be used) and \texttt{2000..2009} is the range, which is the start and end value in years.
5.5.1.4  \texttt{lsS [NAME]} - single value variables

If the target is an Analysis report, you can use the \texttt{lsS} parameter to set a single value for a variable. You can apply the \texttt{lsS} parameter to the following variable types:

- single value
- hierarchy
- hierarchy node
- keydate
- formula
- currency

Refer to the \texttt{lsS[NAME]} - single prompt [page 25] parameter reference for more information.

\textbf{Example}

\textbf{Setting a date parameter}

Here is an example of setting a date variable \texttt{ZCR\_DT} to a value of August 3, 2011 20110803:

\begin{verbatim}
?iDocID=Aa6Grz79cRAma0SMGoaKl
&sIDType=CUID
&lsSZCR\_DT=20110803
\end{verbatim}

\textbf{Example}

\textbf{Setting a currency parameter}

This example sets the value of the currency variable \texttt{Z\_CUR\_MD} to EUR, which is the technical name for the Euro.

\begin{verbatim}
&lsSZ\_CUR\_MD=EUR
\end{verbatim}

\textbf{Example}

\textbf{Setting a hierarchy and hierarchy node}

This example assigns a hierarchy node variable \texttt{Z\_VAR013} to a value of 34 which represents the key for the country of Jamaica, and the next line in the example is hierarchy variable \texttt{Z\_VAR011} which is set to the value \texttt{COUNTRY\_HIERARCHY\_02} which is the key for "Country Hierarchy 2".

\begin{verbatim}
&lsSZ\_VAR013=34
&lsSZ\_VAR011=COUNTRY\_HIERARCHY\_02
\end{verbatim}

\textbf{Example}

\textbf{Setting several different parameter types}

Here is an example of how to set different data connections, using several different single variable types to illustrate how to assemble a longer URL:
5.5.2 Crystal Reports

5.5.2.1 lsM[NAME] - multiple value variables

Crystal reports

If the target is a Crystal report, [NAME] is the parameter name, and each parameter value must be enclosed in square brackets, and comma(,) must be used as a separator to separate multiple entries.

Example

Setting Crystal report parameters

?iDocID=Aa6GrrM79cRSAmOSMGoahKI
&sIDType=CUID
&lsS[1]Z_VAR013=34
&lsS[1]Z_VAR011=COUNTRY_HIERARCHY_02
&lsS[2]Z_CUR_MD=EUR
&lsS[3]DT_IH=20100107
&lsS[4]Z_VAR05=1
&lsS[5]ZCR_DT=20110803

This example opens up a Crystal report with a parameter named SelectState and sets its value to Alberta and Washington.

5.5.2.2 lsR[NAME] - range prompts

Crystal reports

If the target is a Crystal report, [NAME] is the parameter name, and the range must be enclosed in square brackets and/or parentheses (use a square bracket next to a value to include it in the range, and parentheses to exclude it.)
5.5.2.3 ls\{NAME\} - single prompt

Crystal reports

If the target is a Crystal report, \{NAME\} is the parameter name.

Example

Setting a Crystal report parameter

?idDocID=ASsonDFQtV0mHZZJ7JuSo
&sIDType=CUID
&lsMSelectState=California

This example opens up a Crystal report with a parameter named SelectState and sets its value to California.

5.5.2.4 sOutputFormat

Table 28:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
</table>
| sOutputFormat| Specifies the format in which to open the target document. | • H (HTML)
• P (PDF)
• E (Microsoft Excel (97-2003) - Crystal reports only)
• W (Rich Text Format (RTF) - Crystal reports only) |

Note

Defaults to HTML if this parameter is not specified.

Example

?idDocID=Aa6GrMr79cRAma0SMGoaKI
&sIDType=CUID
&sOutputFormat=E
### 5.5.2.5 sPartContext

Table 29:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>sPartContext</td>
<td>Specifies the data context of a report part. Use in conjunction with sReportPart.</td>
<td>The name of the report part data context.</td>
</tr>
</tbody>
</table>

**Note**

Only supported by Crystal reports.

**Note**

Only mandatory if a value is specified for sReportPart.

**Example**

```
?iDocID=Aa6GrrW79cRaMa0EMGoadKI
&sIDType=CUID
&sReportPart=Part1
&sPartContext=0-4-0
```

**Note**

The **sReportPart** and **sPartContext** parameters are only supported with the DHML parts viewer (sViewer=part).

### 5.5.2.6 sRefresh

**Crystal reports**

The **sRefresh** parameter is only supported with the html and part Crystal report viewers, and not the actx and java viewers.
### 5.5.2.7 sReportMode

Table 30:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
</table>
| sReportMode | Indicates whether the link should open the full target Crystal report or just the report part specified in sReportPart. | • Full  
• Part |

**i Note**

- Defaults to **Full** if this parameter is not specified. Only applies if a value is specified for sReportPart.

**Example**

```
 iDocID=Aa6GrM79cRAma0SMGoadKI
 &sIDType=UID
 &sReportPart=Part1
 &sReportMode=Part
```

### 5.5.2.8 sReportName

Table 31:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>sReportName</td>
<td>Specifies the report to open if the target document contains multiple reports.</td>
<td>The report name for Web Intelligence documents and page name for A-OLAP Intelligence reports.</td>
</tr>
</tbody>
</table>

**i Note**

- Defaults to the first report if this parameter is not specified.

**Example**

```
 iDocID=Aa6GrM79cRAma0SMGoadKI
 &sIDType=UID
 &sReportName=First+Report+Tab
```
### 5.5.2.9 sReportPart

Table 32:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>sReportPart</td>
<td>Specifies the part of the target Crystal report to open.</td>
<td>Name of the Crystal report part.</td>
</tr>
</tbody>
</table>

**i Note**

Only supported by Crystal reports.

**Example**

?iDocID=Aa6GrrM79cRAnaO3MSGoadKl
&sIDType=CUID
&sReportPart=Part1

**i Note**

The `sReportPart` and `sPartContext` parameters are only supported with the DHML parts viewer (`sViewer=part`).

### 5.5.2.10 sViewer

Table 33:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
</table>
| sViewer | Specifies the selected report viewer. | • html  
|         |                 | • part (Crystal reports only)     |

**Example**

?iDocID=Aa6GrrM79cRAnaO3MSGoadKl
&sIDType=CUID
&sViewer=html
5.5.3  Web Intelligence

5.5.3.1  Passing BEx prompt variables in an OpenDocument URL

Using technical names

Technical Names for SAP BEx variables, not the the more readable name aliases, must be used in OpenDocument URLs. These technical names can be obtained using the MDX Test editor. A technical name has no spaces and uses upper case letters, for example `0FISCPER`, and the values used within OpenDoc URLs must be enclosed in square brackets, for example `[0FISCPER]`.

The syntax is as follows:

```
http://<domain name>:<port number>/OpenDocument/opendoc/openDocument.jsp
?SIDType=CUID
&iDocID=<23-digit case-sensitive document ID>
&iType=xml
&sRefresh=Y
&sOutputFormat=H
&lsS<technical name of BEx variable>=[<InfoObject technical name>].[<Member / user_input>]
```

A correctly formatted `lsS` parameter would appear as follows:

```
&lsSFiscal+Period+-+User+Input=[0FISCPER].[K42010001]
```

5.5.3.2  `lsI[NAME]` - Index

Web Intelligence documents

When linking to a document that refers to a BEX, unx, or unv using prompts with index as shown in the following example.

```
Example
```

```
?&ls[S/M/R][Prompt identifier]={caption]
&lsI[Prompt identifier]={[key]
```

**Note**

URL encoding is required for `{Prompt identifier}` and `{key}` values. To get a link using Webi, select a cell and right-click, choose Linking/Add Document Link.
5.5.3.3 \texttt{lsM[NAME]} - Multiple value variables

Web Intelligence documents

The character \texttt{?} is a reserved prompt value for Web Intelligence documents in an OpenDocument URL. Setting the prompt value to \texttt{lsM[NAME]=?} in the URL forces the Prompts dialog box to appear for that particular prompt. You must use semicolon(;) as a separator to separate multiple entries.

\textbf{Note}

The following examples show how to use the authorization variable with and without the \texttt{no_value} parameter. In the examples, each OpenDocument variable is shown on its own line for clarity, however the actual URL is one unbroken line.

\textbf{Example}

1. The authorization variable is selected, \texttt{lsM<VAR>=no_value and sRefresh=Y}.

In this example, an error message will be produced that reads as follows:

\begin{verbatim}
BW System XE8 returned state : USER_NOTAUTHORIZED.
Message = WARNING EYE (007): You do not have sufficient authorization
\end{verbatim}

\begin{verbatim}
?iDocID=Aa6GrrM79cRamaOSMGoadKI
&sIDType=CUID
&sType=wid
&sRefresh=Y
&lsMVAR_AUTH_MULT=no_value
\end{verbatim}

\textbf{Example}

2. The authorization variable is selected, \texttt{lsM<VAR>=? and sRefresh=Y}.

In this example, the Prompts dialog box appears, and permissable values may be selected to narrow down what will be displayed:

\begin{verbatim}
?iDocID=Aa6GrrM79cRamaOSMGoadKI
&sIDType=CUID
&sType=wid
&sRefresh=Y
&lsMVAR_AUTH_MULT=?
\end{verbatim}

\textbf{Example}

3. The authorization variable is not selected, \texttt{lsM<VAR>=no_value and sRefresh=N}.

In this example, there is no Prompts dialog box and a report will display with all available information:

\begin{verbatim}
?iDocID=Aa6GrrM79cRamaOSMGoadKI
\end{verbatim}
4. The authorization variable is not selected, \texttt{lsM\textless VAR\textgreater =?} and \texttt{sRefresh=N}.

In this example, the Prompts dialog box appears and all available selection criteria may be selected to narrow down what will be displayed:

```
?iDocID=Aa6GrrM79cRAma0SMGoadKI
&slDType=CUID
&stype=wid
&sRefresh=N
&lsSVAR\_PRODUCT\_OPT\_DEF=\
```

### 5.5.3.4 \texttt{lsS\{NAME\}} - Single prompt

**Web Intelligence documents**

The character `?` is a reserved prompt value for Web Intelligence documents in an openDocument URL. Setting the prompt value to \texttt{lsS\{NAME\}=?} in the URL forces the \textit{Prompts} dialog box to appear for that particular prompt.

### 5.5.3.5 NAII

The \texttt{NAII} out parameter is specific to Web Intelligence that allows you to pass a \texttt{Y} or \texttt{N} flag to display the prompt selection page.

**Table 34:**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAII</td>
<td>Indicates whether to force the display of the prompt selection page.</td>
<td>\texttt{Y} (prompt values that are passed with \texttt{lsS}, \texttt{lsM}, or \texttt{lsR} in the URL are applied and not displayed in the \textit{Prompts} dialog box)</td>
</tr>
</tbody>
</table>

\textit{Note}:

- \texttt{NAII=Y} raises the \textit{Prompts} dialog box for any values not specified in the URL. Prompts created with default values are still displayed in the \textit{Prompts} dialog box.
If all prompt values are specified in the URL, the prompt window does not appear even if NAII=Y is specified.

Example

This example assumes there are two prompts in the Web Intelligence document: Year and Country. NAII=Y forces the Prompts dialog box to appear and allows the user to specify a value for the Country prompt. The Year prompt is already set to a value of FY1999 in the URL using the lsS parameter and therefore is not prompted for.

?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=GUID
&lsSYear=FY1999
&NAII=Y
&sRefresh=Y

5.5.3.6 noDocument

Table 35:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>noDocument</td>
<td>A value of true forces a report to open in design mode using the existing report template.</td>
<td>Boolean value: true</td>
</tr>
</tbody>
</table>

Note

- noDocument=true automatically forces a Web Intelligence report into design mode.
- Since the existing report template is used, you can prevent overwriting this template by applying the appropriate security.

Example

iDocID=6471&noDocument=true
5.5.3.7  sReportName

Table 36:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>sReportName</td>
<td>Specifies the report to open if the target</td>
<td>The report name for Web Intelligence documents and page name for A-OLAP Intelligence reports.</td>
</tr>
<tr>
<td></td>
<td>document contains multiple reports.</td>
<td></td>
</tr>
</tbody>
</table>

i  Note

Defaults to the first report if this parameter is not specified.

Example

?iDocID=Aa6GrrM79cRAmoSMGoadKI
&sIDType=CUID
&sReportName=First+Report+Tab

5.5.4  eView and Information Spaces

5.5.4.1  eView

Obtaining the base URL of an eView document

In the Central Management Console (CMC), view the Folders > Objects List, set the Type column filter to view Explorer View Set documents, and navigate to the appropriate Explorer View Set document. Right-click on the document, select Properties, then copy the 23-character CUID value.

Test that this base URL works by assembling the link in the following format as follows, then paste the link into a web browser address bar.

?sIDType=CUID
&iDocID=Ac4WR2LgKmPtpEc_npw_Nk
&mode=album

i  Note

The finished URL must not contain line returns. Line breaks are used in the example code snippets to more clearly show and separate the parameters.
Building an eView Open Document URL

Filtering the view requires adding parameters using the syntax listed in Customizing Information Spaces with dynamic URL parameters:

Using the previous base URL example and adding a filter of the Region to show only Europe and EEMEA using the parameter fans (the parameter for a sorted list of facets and facets values) using the separator characters _|_ is as follows:

```
?siTypeID=CUID
&iDocID=Ac4WR2LgKMPqEc_npw_Nk
&fans=Region_|_Europe_|_EEME
```

URL encoding of special characters

**Note**

Encode the URL if the link fails for some recipients due to the presence of spaces, commas or other special characters. For example

- `&fans=Region_|_Europe_|_EEME` would be encoded to look like
- `%26amp%3Bfans%3DRegion_%3A_Europe_%5D_EEME`

5.5.4.2 Customizing Information Spaces with dynamic URL parameters

SAP BusinessObjects Crystal Reports and Interactive Analysis tools support the insertion of dynamic hyperlinks in a query that return an HTML page for a dimension. You can use this feature to insert a URL using an SAP BusinessObjects Explorer specific URL syntax specific to open and customize an Information Space that corresponds to a dimension in a Crystal Report or Interactive Analysis query.

Explorer may be referenced from any URL-supported location such as an email link, web page or address bar from a web browser. An example of such a link would include the CUID of an Explorer workspace such as "isid=ASFuWlg_wBpQq7MrYiTYY_g" and preselected facets and facets values such as "fan=FacetNameA_:FacetValueA1". Refer to the Crystal reports and Interactive Analysis user guides for information on how to implement the URL. The syntax for the URL is as follows:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Example</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>Url for retrieving the parameters.</td>
<td>url=dummy.xml</td>
<td>Needs a valid url with a well formed xml</td>
</tr>
<tr>
<td>isid</td>
<td>Information Space ID (CUID).</td>
<td>isid=ASFuWlg_wBpQq7MrYiTYY_g</td>
<td>A valid Information Space CUID</td>
</tr>
<tr>
<td>Syntax</td>
<td>Description</td>
<td>Example</td>
<td>Values</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>isna</td>
<td>Information Space Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| fans   | A sorted list of facets and facets values | fan=FacetNameA_:_:FacetValueA1 | A list of facets and facets values with separators:  
- `_:_:` is the separator for facet and its values  
- `_:_:` is the separator for facet values  
- `_:_` is the separator for facets |
| mens   | A sorted list of measures | mens=Revenue;mens=Revenue_:_:Quantity | A list of measures:  
- `_:_:` is the separator for measures |
| secr   | A search criteria | secr=revenue | A search criteria (string) |
| cht    | Chart type | cht=bvg | \- bvg horizontal bar  
- bv vertical bar  
- mr multiradar  
- r radar  
- sf surface  
- bvgd vertical bar dual axis  
- tg tag cloud  
- p pie  
- bhs horizontal stacked bar  
- bvs vertical stacked bar  
- mp multipie  
- tm treemap  
- kxy XY chart  
- s bubble  
- lc line  
- lcd line dual axis |
| chdi   | Chart analysis dimension | chdi=Year | A valid dimension (facet) |
| chso   | Chart ordered by | chso=Revenue  
- chso=Revenue_:_:asc | Specify the dimension or measure that determines the ordering. It accepts an optional parameter that gives the sort type (for example ascending.) |
| chts   | Chart threshold | chts=12 | Specify the chart threshold. |

Restrictions: When missing, application default = 12.
<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Example</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>chot</td>
<td>Chart display “others”</td>
<td>chot=false</td>
<td>When “false” is specified, it will hide “others”</td>
</tr>
<tr>
<td></td>
<td>Restrictions: When missing, application default = YES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cui</td>
<td>Control User Interface</td>
<td>cui=htb:__hball</td>
<td>Specify the user interface “quick customization”:</td>
</tr>
<tr>
<td></td>
<td>Restrictions: When missing, application default applies.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Example

**Customizing Information Spaces with dynamic URLs**

- Selecting 2 measures: `store_cost` and `unit_sales` and creating two filters on `store_type` and `store_number`:

  ```
  +Grocery_:;_store_number_:;_3_:;_11_:;_15
  ```

- Set the chart to ‘horizontal bar’, select 2 measures, set the analysis dimension, display only 5 values and hide ‘others’ value:

  ```
  ```

- An URL that specifies the user interface style:

  ```
  http://vs0112:50001/explorer/index.jsp?isna=cube+7&cui=htb
  ```
Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

Accessibility

The information contained in the SAP documentation represents SAP’s current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of wilful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or wilful misconduct. All links are categorized for transparency (see: http://help.sap.com/disclaimer).

© 2015 SAP SE or an SAP affiliate company. All rights reserved.