

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	27

5.4	Output parameters.	27
	NAII.	27
	sOutputFormat.	28
	sViewer.	29
	noDocument.	29
5.5	Product specific parameters or usage.	30
	Analysis, edition for OLAP.	30
	Crystal Reports.	36
	Web Intelligence.	40
	eView and Information Spaces.	45

1 Document History

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

Version	Date	Description
SAP BusinessObjects Business Intelligence platform 4.1	May, 2013	First release of this document.
SAP BusinessObjects Business Intelligence platform 4.1 SP05	October, 2014	<ul style="list-style-type: none">• Updated the "IsM[NAME] - multiple value variables" section of "Crystal Reports"• Updated the "IsM[NAME] - multiple value variables" section of "Web Intelligence"• Updated the "IsM[NAME] - multiple value variables" section of "Input parameters"• Updated the table's "Description" of "sInstance" section
SAP BusinessObjects Business Intelligence platform 4.1 SP06	May, 2015	Added a new note in the "User sessions" section of "Session management"

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:

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp?  
iDocID=Aa6GrrM79cRAmaOSMGoadKI&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:

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?<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

Parameter	Description	Replace with
sIDType=GUID <i>i</i> Note Deprecated in SAP BusinessObjects Enterprise XI 3.1 SP3)	Specifies that a GUID is used to specify the viewable document. Use in conjunction with iDocID.	Use sIDType=CUID instead.
sIDType=RUID <i>i</i> Note Deprecated in SAP BusinessObjects Enterprise XI 3.1 SP3)	Specifies that a RUID is used to specify the viewable document. Use in conjunction with iDocID.	Use sIDType=CUID instead.
sKind	Specifies the SI_KIND property of the target Desktop Intelligence document.	Use iDocID instead.
sPath	The file path of the target document.	Use iDocID instead. <i>i</i> Note 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.
sType	Specifies the file type of the target document.	Use iDocID instead.
sViewer=actx	Specifies the Crystal Reports ActiveX Viewer.	Use sViewer=html or sViewer=part instead. The ActiveX Viewer is deprecated as of this release.
sViewer=java	Specifies the Crystal Reports Java Applet Viewer.	Use sViewer=html or sViewer=part instead. The Java Applet Viewer is deprecated as of this release.

Table 6: Obsolete Parameters

Parameter	Description	Replace with
<p>sWindow</p> <div> <p>i Note</p> <p>Obsolete as of SAP BusinessObjects Enterprise XI 3.1 SP3</p> </div>	Indicates whether the target document will open in the current browser window or whether a new window will be launched.	Use the HTML anchor's <code>target</code> attribute or an equivalent. For example: <code>...</code>

3 OpenDocument syntax

3.1 Basic URL syntax

The basic syntax for an OpenDocument URL is as follows:

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/<platformSpecific>  
?<parameter1>  
&<parameter2>  
&...  
&<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`

i 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:

```
&sDocName=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:

```
<a href="http://<servername>:<port>/BOE/OpenDocument/opendoc/<platformSpecific>?  
      iDocID=Aa6GrrM79cRAmaOSMGoadKI  
      &sIDType=CUID"  
      target="_blank">hyperlink text</a>
```

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*.

```
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
    ( "http://<server>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
      ?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*.

```
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
    ( "http://<server>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
      ?iDocID=Aa6GrrM79cRAmaOSMGoadKI
      &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*.

```
http://<server>:<port>/BOE/OpenDocument/opensdoc/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.

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 a `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.
- The User session is released once the user closes the browser.

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

Parameter	Description
serSes [page 17]	Specifies a valid serialized Enterprise session.
token [page 18]	Specifies a valid logon token for the current Enterprise session.

Table 8: Document Identifier Parameters

Parameter	Description
iDocID [page 20]	Specifies the unique identifier of the viewable document in the CMS. Use in conjunction with <code>sIDType</code> .
sDocName [page 20]	Specifies the name of the viewable document in the CMS.
sIDType [page 21]	Specifies the type of object identifier used to specify the viewable document. Use in conjunction with <code>iDocID</code> .
sInstance [page 22]	Specifies the scheduled instance of the target document to open. Use in conjunction with <code>sDocName</code> or <code>iDocID</code> .

Table 9: Input Parameters

Parameter	Description
IsC[NAME] - complex variables [page 22]	Specifies a contextual prompt for Web Intelligence documents if there is an ambiguity during SQL generation.
IsI[NAME] - index [page 23]	Specifies index or key values for a prompt. <code>[NAME]</code> is the text of the prompt.
IsM[NAME] - multiple value variables [page 23]	Specifies multiple values for a prompt. <code>[NAME]</code> is the text of the prompt.
IsR[NAME] - range prompts [page 24]	Specifies a range of values for a prompt. <code>[NAME]</code> is the text of the prompt.
IsS[NAME] - single prompt [page 25]	Specifies a value for a single prompt. <code>[NAME]</code> is the text of the prompt.

Parameter	Description
sPartContext [page 37]	Specifies the data context of a Crystal report part. Use in conjunction with <code>sReportPart</code> .
sRefresh [page 26]	Indicates whether a database refresh should be forced when the target document is opened.
sReportMode [page 38]	Indicates whether the link should open the full target Crystal report or just the report part specified in.
sReportName [page 27]	Specifies the report to open if the target document contains multiple reports.
sReportPart [page 39]	Specifies the part of the target Crystal report to open.

Table 10: Output Parameters

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

5.1 Session management parameters

5.1.1 `serSes`

Table 11:

Syntax	Description	Values
<code>serSes</code>	Specifies a valid serialized Enterprise session.	A serialized string representing the current Enterprise session.

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.

```
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
    ( "http://<server>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
      ?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 12:

Syntax	Description	Values
token	Specifies a valid logon token for the current Enterprise session.	The logon token for the current Enterprise session.

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.

```
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"
    );
    return
    ( "http://<server>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
      ?iDocID=Aa6GrrM79cRAmaOSMGoadKI
      &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:

Syntax	Description	Values
<code>iDocID</code>	Specifies the unique identifier of the viewable document in the CMS. Use in conjunction with <code>sIDType</code> .	A numerical identifier associated with the document in the CMS.

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

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp  
?iDocID=2010
```

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp  
?iDocID=Aa6GrrM79cRAmaOSMGoadKI  
&sIDType=CUID
```

5.2.2 `sDocName`

Table 14:

Syntax	Description	Values
<code>sDocName</code>	Specifies the name of the viewable document in the CMS.	The title of the document in the CMS.

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

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?sDocName=Sales+in+2003
```

5.2.3 `sIDType`

Table 15:

Syntax	Description	Values
<code>sIDType</code>	Specifies the type of object identifier used to specify the viewable document. Use in conjunction with <code>iDocID</code> .	<ul style="list-style-type: none">• <code>InfoObjectID</code>• <code>ParentID</code>• <code>CUID</code>

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

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
```

5.2.4 sInstance

Table 16:

Syntax	Description	Values
sInstance	Specifies the scheduled instance(created during scheduling) of the target document to open. Use in conjunction with sDocName or iDocID.	<ul style="list-style-type: none">• 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

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp  
?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:

Syntax	Description	Values
lsc[NAME]	Specifies a contextual prompt if there is an ambiguity during SQL generation.	A prompt value that resolves the ambiguity in the SQL generation.

Example

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp?  
iDocID=Aa6GrrM79cRAmaOSMGoadKI  
&sIDType=CUID  
&lsc=Sales
```

5.3.2 `lsI[NAME]` - index

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

Table 18:

Syntax	Description	Values
<code>lsI[NAME]</code>	Specifies index or key value. This parameter must be associated with one of the parameters lsS[NAME] - single prompt [page 25] , lsM[NAME] - multiple value variables [page 23] or lsR[NAME] - range prompts [page 24] .	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].

Example

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?sDoc=IndexTest
&sType=wid
&lsMStore=[caption]
&lsIStore=[index]
```

Example

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

```
http://localhost:8080/OpenDocument/opendoc/openDocument.jsp
?sDoc=IndexTest
&sType=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:

Syntax	Description	Values
lsM[NAME]	Specifies multiple values for a prompt. [NAME] is the text of the prompt.	<ul style="list-style-type: none"> Web Intelligence: Use semicolon(;) to separate multiple prompt values Crystal Reports: Use comma(,) to separate multiple prompts values no_value (only for optional parameters)

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

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp?
iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sRefresh=Y
&lsMSelect+Cities=[Paris],[London]
```

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp?
iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sRefresh=Y
&lsMparamStringDR=c,d
&lsMparamNumberDR=3,4
&lsMparamDateDR=[Date(2003,6,3)],[Date(2003,6,4)]
&lsMparamDateTimeDR=[DateTime(2003,6,1,3,1,1)],[DateTime(2003,6,1,4,1,1)]
```

Related Information

[lsM\[NAME\] - multiple value variables \[page 36\]](#)

[lsM\[NAME\] - Multiple value variables \[page 41\]](#)

5.3.4 lsR[NAME] - range prompts

The `lsR` parameter allows a range to be specified.

Table 20:

Syntax	Description	Values
lsR [NAME]	Specifies a range of values for a prompt. [NAME] is the text of the prompt.	<ul style="list-style-type: none"> A range of values for the prompt, separated by a double period (..). no_value (only for optional parameters)

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

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sRefresh=Y
&lsRTime+Period:=[2000..2004)
```

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sRefresh=Y
&lsRparamStringDR=[h..i]
&lsRparamNumberDR=[7..8]
&lsRparamCurrencyDR=[3..4]
&lsRparamDateDR=[Date(2003,6,7)..Date(2003,6,8)]
&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:

Syntax	Description	Values
lsS [NAME]	Specifies a value for a single prompt. [NAME] is the text of the prompt.	<ul style="list-style-type: none"> A single prompt value. no_value (only for optional parameters)

Note

You can remove an optional parameter from the prompt by setting it to `no_value` in the OpenDocument URL. If you leave an optional parameter out of the OpenDocument URL, a default parameter value will be applied.

Example

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sRefresh=Y
&lsSparamString=h
&lsSparamNumber=1
&lsSparamCurrency=121
&lsSparamDate=Date(2003,6,11)
&lsSparamDateTime=DateTime(2003,6,11,14,38,37)
&lsSparamBoolean=false
&lsSparamTime=Time(12,39,2) &lsSparamStringDR=a
&lsSparamDateDR=Date(2003,6,1)
```

5.3.6 `sRefresh`

Table 22:

Syntax	Description	Values
<code>sRefresh</code>	Indicates whether a database refresh should be forced when the target document is opened.	<ul style="list-style-type: none">• Y• N

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

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sRefresh=Y
```

5.3.7 sReportName

Table 23:

Syntax	Description	Values
sReportName	Specifies the report to open if the target document contains multiple reports.	The report name for Web Intelligence documents and page name for A-OLAP Intelligence reports.

Note

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

Example


```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sReportName=First+Report+Tab
```

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:

Syntax	Description	Values
NAII	Indicates whether to force the display of the prompt selection page.  Note Only supported by Web Intelligence documents.	<ul style="list-style-type: none">Y (prompt values that are passed with l sS, l sM, or l sR in the URL are applied and not displayed in the <i>Prompts</i> dialog box)

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 `lsYear` parameter and therefore is not prompted for.

```
http://<servername>:<port>/BOE/OpenDocument/opensoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&lsYear=FY1999
&NAII=Y
&sRefresh=Y
```

5.4.2 `sOutputFormat`

Table 25:

Syntax	Description	Values
<code>sOutputFormat</code>	Specifies the format in which to open the target document.	<ul style="list-style-type: none">• <code>H</code> (HTML)• <code>P</code> (PDF)• <code>E</code> (Microsoft Excel (97-2003) - Crystal reports only)• <code>W</code> (Rich Text Format (RTF) - Crystal reports only)

Note

Defaults to HTML if this parameter is not specified.

Example

```
http://<servername>:<port>/BOE/OpenDocument/opensoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sOutputFormat=E
```

5.4.3 sViewer

Table 26:


Syntax	Description	Values
sViewer	Specifies the selected report viewer.	<ul style="list-style-type: none">htmlpart (Crystal reports only)

Example

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sViewer=html
```

5.4.4 noDocument

Table 27:

Syntax	Description	Values
noDocument	<p>A value of <code>true</code> forces a report to open in design mode using the existing report template.</p> <div> Note Only supported by Web Intelligence documents.</div>	Boolean value: <code>true</code>

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

```
http://<server>:8080/BOE/OpenDocument/opendoc/openDocument.jsp?
iDocID=6471&noDocument=true
```

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
- the values for the variable

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:

```
&1sSZ_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

- &1sC[1]Z_VAR06==1;[]5,20;![]10,15 would be encoded to look like
- %26amp%3B1sC%5B1%5DZ_VAR06%3D%3D1%3B%5B%5D5%2C20%3B!%5B%5D10%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.

```
http://<server>:<port>/BOE/OpenDocument/opensdoc/openDocument.jsp
?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 &IsC[4]Z_VAR06.

- The first connection `lsS[1]` includes a single `COUNTRY_HIERARCHY_01` value applied to the `Z_VAR01` variable, and a single `EUROPE` value applied to the `Z_VAR04` variable.
- The second connection `lsS[2]` includes a single date (2011, 11, 11) value applied to the `Z_VAR02` variable, and a single `SALES_HIERARCHY_03` value applied to the `Z_VAR03` variable.
- The third connection `lsS[3]` includes multiple values 1, 2, and 3, which are associated to the `Z_VAR05` variable.
- The fourth connection `lsS[4]` includes complex variable values applied to `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.

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

Open Document tags used by Analysis, edition for OLAP

Note

Tags used in construction of A-OLAP OpenDocument links include:

- [IsC\[NAME\] - complex variables \[page 32\]](#) (complex variables)
- [IsM\[NAME\] - multiple value variables \[page 33\]](#) (multiple value variables)
- [IsR \[NAME\] - interval variables \[page 34\]](#) (interval variables)
- [IsS \[NAME\] - single value variables \[page 34\]](#) (single value variables)

5.5.1.1 [IsC \[NAME\]](#) - complex variables

The following example shows a typical use of the `IsC` 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.

```
http://<server>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp?  
?sIDType=CUID  
&iDocID=AeGNibIUS.1NmV45dz3jeP4  
&IsC_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.

```
http://<server>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp?  
?sIDType=CUID  
&iDocID=&<ID>  
&IsC[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 `IsC` parameter for Analysis URLs.

Syntax	Description	Example
[]	within range	lsCZ_VAR01=[]5,10
! []	outside of range	lsCZ_VAR01=! []5,10
=	equal	lsCZ_VAR01==5
!=	not equal	lsCZ_VAR01!=5
>	greater than 5	lsCZ_VAR01=>5
<	less than 5	lsCZ_VAR01=<5
>=	equal or greater than	lsCZ_VAR01=>=5
<=	equal or less than	lsCZ_VAR01=<=5

5.5.1.2 `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 `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.

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 `&lsM`), followed by the variable's technical name `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.

```
http://<server>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?sIDType=CUID
&iDocID=<ID>
&lsMVAR_HN_S=1,3
```

5.5.1.3 `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):

```
&lsR[2]VAR_20=2000..2009
```

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), `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 `2000..2009` is the range, which is the start and end value in years.

5.5.1.4 `lsS [NAME]` - single value variables

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

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

Refer to the [lsS\[NAME\] - single prompt \[page 25\]](#) parameter reference for more information.

Example

Setting a date parameter

Here is an example of setting a date variable `ZCR_DT` to a value of August 3, 2011 `20110803`:

```
http://<servername>:<port>/BOE/OpenDocument/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&lsSZCR_DT=20110803
```

Example

Setting a currency parameter

This example sets the value of the currency variable `Z_CUR_MD` to `EUR`, which is the technical name for the Euro.

```
&lsSZ_CUR_MD=EUR
```

Example

Setting a hierarchy and hierarchy node

This example assigns a hierarchy node variable `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 `Z_VAR011` which is set to the value `COUNTRY_HIERARCHY_02` which is the key for "Country Hierarchy 2".

```
&lsSZ_VAR013=34  
&lsSZ_VAR011=COUNTRY_HIERARCHY_02
```

Example

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:

Note

The angle-bracketed items are comments that are NOT included in the string. They are shown in the code snippet to show which type of single variable is being demonstrated.

```
http://<servername>:<port>/BOE/OpenDocument/openDocument.jsp  
?iDocID=Aa6GrrM79cRAmaOSMGoadKI  
&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
```

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, 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

```
http://<servername>:<port>/BOE/OpenDocument/openDocument.jsp
?iDocID=ASsonFDFQtVomHZZJJtJuSo
&sIDType=CUID
&lsMSelectState=[Alberta],[Washington]
```

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 lsS[NAME] - single prompt

Crystal reports

If the target is a Crystal report, [NAME] is the parameter name.

Example

Setting a Crystal report parameter

```
http://<servername>:<port>/BOE/OpenDocument/openDocument.jsp
?iDocID=ASsonFDFQtVomHZZJJtJuSo
&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:

Syntax	Description	Values
<code>sOutputFormat</code>	Specifies the format in which to open the target document.	<ul style="list-style-type: none">• <code>H</code> (HTML)• <code>P</code> (PDF)• <code>E</code> (Microsoft Excel (97-2003) - Crystal reports only)• <code>W</code> (Rich Text Format (RTF) - Crystal reports only)

Note


Defaults to HTML if this parameter is not specified.

Example

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sOutputFormat=E
```

5.5.2.5 `sPartContext`

Table 29:

Syntax	Description	Values
<code>sPartContext</code>	<p>Specifies the data context of a report part. Use in conjunction with <code>sReportPart</code>.</p> <div> Note Only supported by Crystal reports.</div>	The name of the report part data context.

Note

Only mandatory if a value is specified for `sReportPart`.

Example

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&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:

Syntax	Description	Values
<code>sReportMode</code>	Indicates whether the link should open the full target Crystal report or just the report part specified in <code>sReportPart</code> .  Note Only supported by Crystal reports.	<ul style="list-style-type: none">• <code>Full</code>• <code>Part</code>

Note

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

Example

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sReportPart=Part1
```

&sReportMode=Part

5.5.2.8 sReportName

Table 31:

Syntax	Description	Values
sReportName	Specifies the report to open if the target document contains multiple reports.	The report name for Web Intelligence documents and page name for A-OLAP Intelligence reports.

Note


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

Example

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sReportName=First+Report+Tab
```

5.5.2.9 sReportPart

Table 32:

Syntax	Description	Values
sReportPart	Specifies the part of the target Crystal report to open. <div><h4> Note</h4><p>Only supported by Crystal reports.</p></div>	Name of the Crystal report part.

Example

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sReportPart=Part1
```

Note

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

5.5.2.10 `sViewer`

Table 33:

Syntax	Description	Values
<code>sViewer</code>	Specifies the selected report viewer.	<ul style="list-style-type: none">• <code>html</code>• <code>part</code> (Crystal reports only)

Example

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&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>
&sType=wid
&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

```
http://<server>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?<doc identifier>
&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 `lsM[NAME]` - Multiple value variables

Web Intelligence documents

The character `?` is a reserved prompt value for Web Intelligence documents in an OpenDocument URL. Setting the prompt value to `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.

Note

The following examples show how to use the authorization variable with and without the `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.

Example

1. The authorization variable is selected, `lsM<VAR>=no_value` and `sRefresh=Y`.

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

```
BW System XE8 returned state : USER_NOT_AUTHORIZED.
```

```
Message = WARNING EYE (007): You do not have sufficient
authorization"
```

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sType=wid
&sRefresh=Y
&lsMVAR_AUTH_MULT=no_value
```

Example

2. The authorization variable is selected, `lsM<VAR>=?` and `sRefresh=Y`.

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

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sType=wid
&sRefresh=Y
&lsMVAR_AUTH_MULT=?
```

Example

3. The authorization variable is not selected, `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:

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sType=wid
&sRefresh=N
&lsSVAR_PRODUCT_OPT_DEF=no_value
```

Example

4. The authorization variable is not selected, `lsM<VAR>=?` and `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:

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&sType=wid
&sRefresh=N
&lsSVAR_PRODUCT_OPT_DEF=?
```

5.5.3.4 `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 `lsS [NAME] = ?` in the URL forces the *Prompts* dialog box to appear for that particular prompt.

5.5.3.5 `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 34:

Syntax	Description	Values
<code>NAII</code>	Indicates whether to force the display of the prompt selection page. Note Only supported by Web Intelligence documents.	<ul style="list-style-type: none"><code>Y</code> (prompt values that are passed with <code>lsS</code>, <code>lsM</code>, or <code>lsR</code> in the URL are applied and not displayed in the <i>Prompts</i> dialog box)

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.

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&lsSYear=FY1999
&NAII=Y
&sRefresh=Y
```

5.5.3.6 noDocument

Table 35:

Syntax	Description	Values
noDocument	<p>A value of <code>true</code> forces a report to open in design mode using the existing report template.</p> <div><p>Note</p><p>Only supported by Web Intelligence documents.</p></div>	Boolean value: <code>true</code>

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

```
http://<server>:8080/BOE/OpenDocument/opendoc/openDocument.jsp?  
iDocID=6471&noDocument=true
```

5.5.3.7 sReportName

Table 36:

Syntax	Description	Values
sReportName	Specifies the report to open if the target document contains multiple reports.	The report name for Web Intelligence documents and page name for A-OLAP Intelligence reports.

Note

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

Example

```
http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp  
?iDocID=Aa6GrrM79cRAmaOSMGoadKI  
&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.

```
http://<server>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?sIDType=CUID
&iDocID=Ac4WR2LgKMpPtqEc_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:

```
http://<server>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
?sIDType=CUID
&iDocID=Ac4WR2LgKMpPtqEc_npw_Nk
&fans=Region:_]_Europe_]_EEMEA]
```

URL encoding of special characters

i 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_]_EEMEA]` would be encoded to look like

-
- %26amp%3Bfans%3DRegion_%3A_Europe_%5D_EEME

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 willful misconduct. All links are categorized for transparency (see: <http://help.sap.com/disclaimer>).

www.sap.com/contactsap

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

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.