PUBLIC
SAP BusinessObjects Business Intelligence platform
Document Version: 4.2 SP7 – 2019-02-28

Viewing Documents Using OpenDocument

THE BEST RUN **SAP**

Content

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	- Updated the "lsM[NAME] - multiple value variables" section of "Crystal Reports" - Updated the "lsM[NAME] - multiple value variables" section of "Web Intelligence" - Updated the "lsM[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"
SAP BusinessObjects Business Intelligence platform 4.2	November 2015	Updated the guide with branding changes.
SAP BusinessObjects Business Intelligence platform 4.2 SP 3	August 2016	Added new parameter "lang" under Crystal Reports and Design Studio of section "Product specific parameters or usage".
SAP BusinessObjects Business Intelligence platform 4.2 SP 4	April, 2017	- Added What's New in SAP BusinessObjects Business Intelligence platform 4.2 SP4. - Added new parameter "pvl" under Crystal Reports, of the section "Product specific parameters or usage". - Added the parameters "pvl" and "lang" under Web Intelligence, of the section "Product specific parameters or usage".
SAP BusinessObjects Business Intelligence platform 4.2 SP5	December 2017	Added new parameters "sReportName" and

Version	Date	Description
		"sReportMode" under Web Intelligence, of the section "Product specific parameters or usage". • Updated parameter "sViewer" and "sOutputFormat" under Web Intelligence, of the section "Product specific parameters or usage.
SAP BusinessObjects Business Intelligence Platform 4.2 SP6	July 2018	• Added new "lsS[NAME] - Complex values answer" topic under Web Intelligence, of the section "Product specific parameters or usage". • Updated the "sReportMode", "sOutputFormat", "lsM[Index]", "lsS[NAME] - Single value" and "lsM[NAME] - Multiple value answers" parameters under Web Intelligence, of the section "Product specific parameters or usage". • Deleted the "Passing BEx prompt variables in an OpenDocument URL" topic under Web Intelligence, of the section "Product specific parameters or usage".
SAP BusinessObjects Business Intelligence Platform 4.2 SP7	February 2019	Added a new topic Viewing a content object through an OpenDocument link [page 14]

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 4.2 SP4

Send To option in OpenDocument for Web Intelligence and Crystal Report Documents

The *Send To* option is now available in Web Intelligence and Crystal Report documents using OpenDocument. By using this option, you can share your documents to other users by sending them to the following:

- *Mail*
- *User* (or *BI Inbox*)
- *FTP*

> i Note
>
> - In Web Intelligence reports, you can see the *Send To* icon (📧) on the toolbar. In case of Crystal Reports, you need to choose *File* in order to see the *Send To* option.
> - The administrator can control *Send To* rights based on users and user groups.

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`

> i 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.

> i 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.

> i Note
>
> Deprecated and obsolete members as of SAP BusinessObjects Enterprise XI 3.1 Service Packs are also listed for reference.

Deprecated Parameters

Parameter	Description	Replace with
`sIDType=GUID` ⓘ **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` ⓘ **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. ⓘ **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.

Obsolete Parameters

Parameter	Description	Replace with
`sWindow` **i Note** Obsolete as of SAP BusinessObjects Enterprise XI 3.1 SP3	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 `target` attribute or an equivalent. For example: `<a href="..." target="_blank">...</a>`

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 Viewing a content object through an OpenDocument link

OpenDocument links provide a direct link to a content object so recipients don't need to navigate folders or categories.

Recipients must have access rights to the document in order to view it through the OpenDocument link.

When a recipient accesses an OpenDocument link in a browser, a logon dialog box does not appear if the BI Launch pad session is active. OpenDocument will consume the existing BI Launch pad session to open the document. If the session is not active, the BI launch pad logon dialog box appears. After the recipient enters valid logon credentials, the document opens.

When the OpenDocument session is about to time out, a *Session Timeout Warning* dialog box appears with a warning message, *Your user session will expire in "N" minute(s). Would you like to continue your user session?*. If the recipient clicks *Continue*, the session will be active for the next "N" minutes, where "N" is the value of session timeout.

When the session times out, a *Session Timeout* dialog box appears with a warning message, *Your session has expired. Close your browser window and restart the application.* To continue viewing the document, the recipient must close the browser window and re-access the OpenDocument link in the browser.

5   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.

5.1   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
>     );
> }
> ```
>
> > **i 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.

5.2   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);
> }
> ```
>
> > i 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/opendoc/openDocument.jsp
>     ?iDocID=<documentID>
>     &sIDType=CUID
>     &token=<logonToken>
> ```
>
> > i 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.

5.3 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.

> i 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.

6 Parameter reference

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

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

Session Management Parameters

Parameter	Description
serSes [page 19]	Specifies a valid serialized Enterprise session.
token [page 20]	Specifies a valid logon token for the current Enterprise session.

Document Identifier Parameters

Parameter	Description
iDocID [page 22]	Specifies the unique identifier of the viewable document in the CMS. Use in conjunction with `sIDType`.
sDocName [page 22]	Specifies the name of the viewable document in the CMS.
sIDType [page 23]	Specifies the type of object identifier used to specify the viewable document. Use in conjunction with `iDocID`.
sInstance [page 24]	Specifies the scheduled instance of the target document to open. Use in conjunction with `sDocName` or `iDocID`.

Input Parameters

Parameter	Description
lsC[NAME] - complex variables [page 24]	Specifies a contextual prompt for Web Intelligence documents if there is an ambiguity during SQL generation.
lsI[NAME] - index [page 24]	Specifies index or key values for a prompt. `[NAME]` is the text of the prompt.
lsM[NAME] - multiple value variables [page 25]	Specifies multiple values for a prompt. `[NAME]` is the text of the prompt.
lsR[NAME] - range prompts [page 26]	Specifies a range of values for a prompt. `[NAME]` is the text of the prompt.
lsS[NAME] - single prompt [page 27]	Specifies a value for a single prompt. `[NAME]` is the text of the prompt.

Parameter	Description
sPartContext [page 38]	Specifies the data context of a Crystal report part. Use in conjunction with `sReportPart`.
sRefresh [page 28]	Indicates whether a database refresh should be forced when the target document is opened.
sReportMode [page 39]	Indicates whether the link should open the full target Crystal report or just the report part specified in.
sReportName [page 39]	Specifies the report to open if the target document contains multiple reports.
sReportPart [page 40]	Specifies the part of the target Crystal report to open.

Output Parameters

Parameter	Description
NAII [page 29]	Indicates whether to force the display of the prompt selection page for Interactive Analysis Desktop prompts.
sOutputFormat [page 41]	Specifies the format in which to open the target document.
sViewer [page 40]	Specifies the selected report viewer.
noDocument [page 46]	Used with Web Intelligence reports, a value of `true` automatically forces a report to open in design mode.

6.1 Session management parameters

6.1.1 `serSes`

Syntax	Description	Values
`serSes`	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
    );
}
```

i 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.

6.1.2 `token`

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
  );
}
```

i 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>
```

i 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.

6.2 Document identifier parameters

6.2.1 iDocID

Syntax	Description	Values
iDocID	Specifies the unique identifier of the viewable document in the CMS. Use in conjunction with sIDType.	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.

> i 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
> ```

6.2.2 sDocName

Syntax	Description	Values
sDocName	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.

> ### i 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
> ```

6.2.3 `sIDType`

Syntax	Description	Values
`sIDType`	Specifies the type of object identifier used to specify the viewable document. Use in conjunction with `iDocID`.	• `InfoObjectID` • `ParentID` • `CUID`

> ### i 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
> ```

6.2.4 `sInstance`

Syntax	Description	Values
`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
>
> ```
> http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
> ?sDocName=Sales+in+2003
> &sInstance=User
> ```

6.3 Input parameters

6.3.1 `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.

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
> ```

6.3.2 `lsI[NAME]` - index

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

Syntax	Description	Values
lsI[NAME]	Specifies index or key value. This parameter must be associated with one of the parameters lsS[NAME] - single prompt [page 27], lsM[NAME] - multiple value variables [page 25] or lsR[NAME] - range prompts [page 26].	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
> ```

6.3.3 lsM[NAME] - multiple value variables

The lsM parameter allows the use of multiple values in a hierarchy node.

Syntax	Description	Values
lsM[NAME]	Specifies multiple values for a prompt. [NAME] is the text of the prompt.	• Web Intelligence: Use semicolon(;) to separate multiple prompt values • Crystal Reports: Use comma(,) to separate multiple prompts values • no_value (only for optional parameters)

> **i 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

6.3.4 lsR[NAME] - range prompts

The lsR parameter allows a range to be specified.

Syntax	Description	Values
lsR[NAME]	Specifies a range of values for a prompt. [NAME] is the text of the prompt.	• A range of values for the prompt, separated by a double period (..). • no_value (only for optional parameters)

> **i 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)]
> ```

6.3.5 lsS[NAME] - single prompt

Syntax	Description	Values
lsS[NAME]	Specifies a value for a single prompt. [NAME] is the text of the prompt.	• A single prompt value. • no_value (only for optional parameters)

> ### 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)
> ```

6.3.6 sRefresh

Syntax	Description	Values
sRefresh	Indicates whether a database refresh should be forced when the target document is opened.	• 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
> ```

6.3.7 `sReportName`

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.

> **i 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
> ```

6.4   Output parameters

6.4.1 `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.

Syntax	Description	Values
`NAII`	Indicates whether to force the display of the prompt selection page.	• `Y` (prompt values that are passed with `lsS`, `lsM`, or `lsR` in the URL are applied and not displayed in the *Prompts* dialog box)

> **i 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
```

6.4.2 sOutputFormat

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

i 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
```

6.4.3 sViewer

Syntax	Description	Values
sViewer	Specifies the selected report viewer.	- html - part

❖ Example

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

6.4.4 `noDocument`

Syntax	Description	Values
`noDocument`	A value of `true` forces a report to open in design mode using the existing report template.	Boolean value: `true`

> **i 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
> ```

6.5 Product specific parameters or usage

6.5.1 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:

```
&lsSZ_VAR01==20120619
```

URL encoding of special characters

> i 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%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/opendoc/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

> **i Note**
>
> Tags used in construction of A-OLAP OpenDocument links include:
>
> - lsC[NAME] - complex variables [page 33] (complex variables)
> - lsM[NAME] - multiple value variables [page 34] (multiple value variables)
> - lsR [NAME] - interval variables [page 35] (interval variables)
> - lsS [NAME] - single value variables [page 35] (single value variables)

6.5.1.1   `lsC[NAME]` - complex variables

The following example shows a typical use of the `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.
>
> ```
> http://<server>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
> ?sIDType=CUID
> &iDocID=AeGNibIUS.1Nmv45dz3jeP4
> ```

```
&lsCZ_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>
> &lsC[1]Z_VAR06==1;[]5,20;![]10,15
> ```

> i 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.

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

6.5.1.2 `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
> ```

6.5.1.3 `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.

6.5.1.4 `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 27] 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:

> **i Note**
>
> The angle-bracketed items are comments that are NOT included in the string. The 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
```

6.5.2 Crystal Reports

6.5.2.1   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 parmeter named `SelectState` and sets its value to `Alberta` and `Washington`.

6.5.2.2   `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).

6.5.2.3   `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 parmeter named `SelectState` and sets its value to California.

6.5.2.4 `sPartContext`

Syntax	Description	Values
`sPartContext`	Specifies the data context of a report part. Use in conjunction with `sReportPart`. > i Note > > Only supported by Crystal reports.	The name of the report part data context.

> i 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
> ```

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

6.5.2.5 `sRefresh`

Crystal reports

The `sRefresh` parameter is only supported with the `html` and `part` Crystal report viewers, and not the `actx` and `java` viewers.

6.5.2.6 sReportMode

Syntax	Description	Values
sReportMode	Indicates whether the link should open the full target Crystal report or just the report part specified in `sReportPart`. **i Note** Only supported by Crystal reports.	• Full • Part

i 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
```

6.5.2.7 sReportName

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.

i 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
```

6.5.2.8 sReportPart

Syntax	Description	Values
sReportPart	Specifies the part of the target Crystal report to open.	Name of the Crystal report part.

⁍ Example

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

i Note

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

6.5.2.9 sViewer

Syntax	Description	Values
sViewer	Specifies the selected report viewer.	• html • part

⁍ Example

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

6.5.2.10 sOutputFormat

Syntax	Description	Values
`sOutputFormat`	Specifies the format in which to open the target document.	`H` (HTML) `P` (PDF) `E` (Microsoft Excel 97-2003) `W` (Rich Text Format (RTF) - Crystal reports only)

> **i 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
> ```

6.5.2.11 lang

Syntax	Description	Values
`lang`	Defines the language in which OpenDocument framework elements are displayed.	Language. i.e en for English

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

> **i Note**
>
> All report specific elements like prompt texts, prompt values, report headers, and report data or values are displayed based on the browser setting, and not defined by the URL Parameter.
>
> This is a client specific implementation and defaults to browser settings if not specified.

6.5.2.12 pvl

Syntax	Description	Values
pvl	Sets the preferred viewing locale of a Crystal Report OpenDocument.	Language. For example, en for English

⁂ Example

```
http://BOE/OpenDocument/opendoc/openDocument.jsp?
iDocID=Aa6GrrM79cRAmaOSMGoadKI
&sIDType=CUID
&pvl=en
```

ⓘ Note

This parameter is independent of the lang parameter.

lang and pvl parameters can be used together or separately, depending on your requirement.

6.5.3 Web Intelligence

6.5.3.1   sReportMode

Syntax	Description	Values
sReportMode	Defines how Web Intelligence displays the document.	• Full: The document is opened in the full Web Intelligence interface. • Light: The document is opened in a simplified interface. In HTML, only the status bar is displayed. In Web Intelligence Interactive Viewer, only the report selector and vanishing toolbar are available.

ⓘ Note

- You can define the Web Intelligence client using the sViewer parameter.
- sReportMode=Light is not compatible with sViewer=Applet.

⁂ Example

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

6.5.3.2 `sReportName`

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.

i 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
```

6.5.3.3 **sReportPart**

Syntax	Description	Values
`sReportPart`	Specifies the part of the target Web Intelligence report to open.	Name of the Web Intelligence report part.

⁕ Example

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

i Note

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

6.5.3.4 sOutputFormat

Syntax	Description	Values
`sOutputFormat`	Specifies the format in which to open the target document.	• `H` (Reading mode) • `E` (XLSX) • `P` (PDF)

> **i Note**
>
> PDF and Excel output formats do not support the `sReportName`, `sReportMode`, `sReportPart`, and `sViewer=Applet` parameters.

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

6.5.3.5 sViewer

Syntax	Description	Values
`sViewer`	Specifies the selected report viewer.	• `html` • `fiori` (Web Intelligence Interactive Viewer) • `applet`

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

6.5.3.6 pvl

Syntax	Description	Values
`pvl`	Sets the preferred viewing locale of a WebIntelligence OpenDocument.	Language. For example, en for English

> ⁘ Example
>
> ```
> http://BOE/OpenDocument/opendoc/openDocument.jsp?
> iDocID=Aa6GrrM79cRAmaOSMGoadKI
> &sIDType=CUID
> &pvl=en
> ```
>
> > ℹ Note
> >
> > This parameter is independent of the lang parameter.
> >
> > lang and pvl parameters can be used together or separately, depending on your requirement.

6.5.3.7 lang

Syntax	Description	Values
`lang`	Defines the language in which OpenDocument framework elements are displayed.	Language. For example, en for English

> ⁘ Example
>
> ```
> http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp
> ?iDocID=Aa6GrrM79cRAmaOSMGoadKI
> &sIDType=CUID
> &lang=en
> ```
>
> > ℹ Note
> >
> > All report specific elements like prompt texts, prompt values, report headers, and report data or values are displayed based on the browser setting, and not defined by the URL Parameter.
> >
> > This is a client specific implementation and defaults to browser settings if not specified.

6.5.3.8 `noDocument`

Syntax	Description	Values
`noDocument`	A value of `true` forces a report to open in design mode using the existing report template.	Boolean value: `true`

> **i 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
> ```

6.5.3.9 `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.

Syntax	Description	Values
`NAII`	Indicates whether to force the display of the prompt selection page.	- `Y` (prompt values that are passed with `lsS`, `lsM`, or `lsR` in the URL are applied and not displayed in the *Prompts* dialog box)

> **i 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
```

6.5.3.10 `lsS[NAME]` - Single value

Syntax	Description	Values
`lsS[Name]=[Value]`	Specifies a single value for a prompt. `[Name]` is the text of the prompt.	`[Value]` is the answer to the prompt.

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.

For SAP BW, `[Name]` is the variable technical name and `[Value]` is the value key. When you use an opendoc URL in a Web Intelligence document and pass the value of an object to answer a prompt, if the prompt is based on a SAP BW variable or an object with index awareness, make sure to use the .key suffix. For example:

```
&lsSCountry=[Country].key
```

6.5.3.11 lsM[NAME] - Multiple values answer

Syntax	Description	Values
`lsM[Name]=[Value1]; [Value2];…`	Specifies a multiple values for a prompt. `[Name]` identifies the prompt.	`[Value1]`, `[Value2]`, … are the answers to the prompt.

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.

For SAP BW, `[Name]` is the variable technical name and `[Value1]`, `[Value2]` are the value key. When you use an opendoc URL in a Web Intelligence document and pass the value of an object to answer a prompt, if the prompt is based on a SAP BW variable or an object with index awareness, you need to use the .key suffix. For example:

```
&lsMCountry=[Country1].key;[Country2].key
```

6.5.3.12 lsM[Name] – Complex values answer

For SAP BW and SAP HANA, lsM can also be used to answer complex values, in addition to multiple values (see previous page).

Syntax	Description	Values
lsM[Name]=[Value]	Equal	[Value] is the operand for the Equal operator. For example: lsMLevel=20
lsM[Name]=![Value]	Not Equal	[Value] is the operand for the Not Equal operator. For example: lsMLevel=!45
lsM[Name]=[Value1] – [Value2]	Interval Between	[Value1] and [Value2] are the range limit for the Between operator: For example: lsMLevel=20 – 50
lsM[Name]=![Value1] – [Value2]	Not Between	[Value1] and [Value2] are the range limit for the Not Between operator: For example: lsMLevel=!20 – 50
lsM[Name]=>[Value]	Greater than	[Value] is the operand for the Greater than operator. For example: lsMLevel=>20
lsM[Name]=<[Value]	Lower than	[Value] is the operand for the Lower than operator. For example: lsMLevel=<50
lsM[Name]=>=[Value]	Greater or equal to	[Value] is the operand for the Greater or equal to operator. For example: lsMLevel=>=20
lsM[Name]=<=[Value]	Lower or equal to	[Value] is the operand for the Lower or equal to operator. For example: lsMLevel=<=50

You can aggregate different operators by using semicolon(;). For example:

```
&lsMLevel=20;21;22;>12;<=50;!45
```

For SAP BW, [Name] is the variable technical name and [Value] is the value key.

6.5.3.13 lsI[NAME] - Index

Syntax	Description	Values
lsI[Name]=[Index]	Specifies an index for a variable based on a SAP BW or object with aggregate awareness. [Name] identifies the prompt.	[Index] is the index for the answer to the prompt.

For SAP BW, `[Name]` is the variable technical name and `[Index]` is the value key. When you use an opendoc URL in a Web Intelligence document and pass the value of an object to answer a prompt, if the prompt is based on a SAP BW variable or an object with index, make sure you use the .key suffix. For example:

```
&lsICountry=[Country].key
```

6.5.4 Design Studio

6.5.4.1 `lang`

Syntax	Description	Values
lang	Defines the language in which OpenDocument framework elements are displayed.	Language. i.e en for English

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

> ⓘ Note
>
> All report-specific elements like prompt texts, prompt values, report headers, and report data or values are displayed based on the browser setting, and not defined by the URL Parameter.
>
> This is a client specific implementation and defaults to browser settings if not specified.

6.5.5 eView and Information Spaces

6.5.5.1 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 paramaters 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_]_EEME
```

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_]_EEME` would be encoded to look like
> - `%26amp%3Bfans%3DRegion_%3A_Europe_%5D_EEME`

7 UI Customization Parameters

This feature allows you to customize the Crystal Report DHTML Viewer depending on your business requirements.

You can append the parameters to the OpenDoc URL as shown below:

Example: http://<servername>:<port>/BOE/OpenDocument/opendoc/openDocument.jsp?iDocID=<Document-ID>&sIDType=CUID`&sGroupTree=hide` , this hides the group tree toggle in the left panel.

The following table provides the list of parameters that you can use to customize the Crystal Report DHTML Viewer:

Parameter Table

Parameter	Value	Description
sAllUIElements	hide	It hides the top toolbar, left panel and status bar and hence, you can see the report without any UI element.
toolbar	hide	It hides the top toolbar so, you can see the left panel and status bar along with the report.
sLeftPanel	hide	It hides the complete left panel.
refresh	hide	It hides *Refresh* in the toolbar.
export	hide	It hides *Export* in the toolbar.
print	hide	It hides *Print* in the toolbar.
sGroupTree	hide	It hides *group tree* toggle in the left panel.
sSearch	hide	It hides *Search* toggle in the left panel
sParameterPanel	hide	It hides *Parameter panel* toggle in the left panel.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:

    - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.

    - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.

- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

THE BEST RUN **SAP**