Viewing Reports and Documents using URLs

BusinessObjects Enterprise XI 3.1

Copyright

2010-03-25

Contents

OpenDocument

1

chapter

Getting started

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 BusinessObjects Enterprise 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 BusinessObjects Enterprise, see the *BusinessObjects Enterprise Web Application Deployment Guide*.

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 BusinessObjects Enterprise deployment is beneficial.

About OpenDocument

OpenDocument is one of many deployed web applications within an BusinessObjects Enterprise system. 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 InfoView. 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>/OpenDocument/opendoc/openDoc
ument.jsp?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sIDType=CUID
```

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 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
- Voyager workspaces
- InfoView dashboards
- Xcelsius visualizations published as SWF files.

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.

What's new with OpenDocument

URL syntax change

The virtual directory `businessobjects` is no longer created during installion of BusinessObjects Enterprise. This affects the URL syntax for OpenDocument links.

The previous URL syntax was:

```
http://<servername>:<port>/businessobjects/enter
prise115/<platformSpecific>?<parameter1>&<parame
ter2>&...&<parameterN>
```

The new URL syntax is:

```
http://<servername>:<port>/OpenDocument/<platformSpecif
ic>?<parameter1>&<parameter2>&...&<parameterN>
```

The exact syntax of the `<platformSpecific>` parameter depends on your BusinessObjects Enterprise deployment:

- For Java deployments, use `openDocument.jsp` in place of the `<platformSpecific>` parameter.
- For .NET deployments , use `opendocument.aspx` in place of the `<platformSpecific>` parameter.

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 BusinessObjects Enterprise 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.

Links to documents must use the new URL syntax, or you must create the appropriate `businessobjects` virtual directory.

sReportMode

`sReportMode` is a new URL parameter to view a Crystal Report in different modes depending on the option passed to the parameter. Possible values include:

- `part` displays part of a report using the parts viewer to render the report.
- `printlayout` displays the report in a print preview layout.
- `weblayout` displays the report as a web page layout.

Interactive Parameters

Crystal Reports parameter prompt values passed into the URL are always be applied to the report, even if the report instance contains saved data. This is a change in behavior from previous releases.

Forcing the Prompts dialog for Web Intelligence prompts

The character `?` is now a reserved prompt value for Web Intelligence documents in an OpenDocument URL. Setting the prompt value to `lsM[NAME]=?` or `lsS[NAME]=?` in the URL forces the "Prompts" dialog box to appear for that particular prompt.

OpenDocument syntax

Basic URL syntax

The basic syntax for an OpenDocument URL is as follows:

```
http://<servername>:<port>/OpenDocument/opendoc/<platform
Specific>?<parameter1>&<parameter2>&...&<parameterN>
```

The exact syntax of the `<platformSpecific>` parameter depends on your BusinessObjects Enterprise deployment:

- For Java deployments, use `openDocument.jsp` in place of the `<platformSpecific>` parameter.
- For .NET deployments , use `opendocument.aspx` in place of the `<platformSpecific>` parameter.

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 BusinessObjects Enterprise 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.

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.

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 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>/OpenDocument/open
doc/<platformSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID" target="_blank">hyperlink text</a>
```

Session management

Normally when using an OpenDocument link to access password-protected documents, the user will be prompted for credentials. OpenDocument allows you to pass 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.

Logon tokens

Logon tokens can be used in OpenDocument by inserting the `token` parameter into the OpenDocument URL. Logon tokens allow users to access password-protected files 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:

The following example uses the BusinessObjects Enterprise Java SDK to pass in a logon token to the OpenDocument URL. For more information on the `ILogonTokenMgr.createLogonToken` method, see the *BusinessObjects Enterprise Java API Reference*.

```
String openDocumentToken() throws SDKException, Unsup
portedEncodingException
{
 IEnterpriseSession sess = CrystalEnterprise.getSession
Mgr().logon ("username", "password", "<cms>:</port>",
"secEnterprise");
 String token = sess.getLogonTokenMgr().createLogonTo
ken("",120,100);
 String tokenEncode = URLEncoder.encode(token, "UTF-8");

 return ("http://<server>:<port>/OpenDocument/open
doc/openDocument.jsp?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&token=" + tokenEncode);
}
```

Note:

- Replace <server> with the server name and <port> with the port number of your web server.
- 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.

User sessions

When OpenDocument is used from InfoView or 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 in the OpenDocument `web.xml` file.
- The OpenDocument URL uses 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 `token` parameter, the existing session will be closed and a new session will be created. That is, you can use `token` parameter to over-ride an existing user session. The OpenDocument application will look for an existing user session in the Web application session and in cookies.

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

If the new `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.

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 1-1: Session Management Parameters

Parameter	Description
token	Specifies a valid logon token for the current Enterprise session.

Table 1-2: Document Identifier Parameters

Parameter	Description
iDocID	Specifies the unique identifier of the viewable document in the CMS. Use in conjunction with `sIDType`.
sDocName	Specifies the name of the viewable document in the CMS. Use in conjunction with `sPath` and `sType`.
sIDType	Specifies the type of object identifier used to specify the viewable document. Use in conjunction with `iDocID`.
sInstance	Specifies the scheduled instance of the target document to open. Use in conjunction with `sDocName` or `iDocID`.
sKind	Specifies the file type of the target Desktop Intelligence document. Use in conjunction with `sDocName` and `sPath`.

Parameter	Description
sPath	Specifies the name of the folder and subfolder containing the target document. Use in conjunction with `sDocName` and `sType`.
sType	Specifies the file type of the target document. Use in conjunction with `sDocName` and `sPath`.

Table 1-3: Input Parameters

Parameter	Description
lsC	Specifies a contextual prompt for Web Intelligence documents if there is an ambiguity during SQL generation.
lsM[NAME]	Specifies multiple values for a prompt. `[NAME]` is the text of the prompt.
lsR[NAME]	Specifies a range of values for a prompt. `[NAME]` is the text of the prompt.
lsS[NAME]	Specifies a value for a single prompt. `[NAME]` is the text of the prompt.
sPartContext	Specifies the data context of a Crystal report part. Use in conjunction with `sReportPart`.

Parameter	Description
sRefresh	Indicates whether a database refresh should be forced when the target document is opened.
sReportMode	Indicates whether the link should open the full target Crystal report or just the report part specified in.
sReportName	Specifies the report to open if the target document contains multiple reports.
sReportPart	Specifies the part of the target Crystal report to open.

Table 1-4: Output Parameters

Parameter	Description
NAII	Indicates whether to force the display of the prompt selection page for Web Intelligence prompts.
sOutputFormat	Specifies the format in which to open the target document.
sViewer	Specifies the selected report viewer.

Session management parameters

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:

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

```
String openDocumentToken() throws SDKException, Unsup
portedEncodingException
{
 IEnterpriseSession sess = CrystalEnterprise.getSession
Mgr().logon ("username", "password", "<cms>:<port>",
"secEnterprise");
 String token = sess.getLogonTokenMgr().createLogonTo
ken("",120,100);
 String tokenEncode = URLEncoder.encode(token, "UTF-8");

 return ("http://<server>:<port>/OpenDocument/open
doc/openDocument.jsp?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&token=" + tokenEncode);
}
```

Note:

- Replace <server> with the server name and <port> with the port number of your web server.
- 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.

Document identifier parameters

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 InfoView applications. The properties page for each document contains the document ID and the CUID. You can also obtain the identifier programmatically using the BusinessObjects Enterprise 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>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=2010
```

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID
```

sDocName

Syntax	Description	Values
`sDocName`	Specifies the name of the viewable document in the CMS. Use in conjunction with `sPath` and `sType`.	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.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?sDocName=Sales+in+2003&sPath=[Sales+Re
ports]&sType=rpt
```

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`

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>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID
```

sInstance

Syntax	Description	Values
`sInstance`	Specifies the scheduled instance 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)

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?sDocName=Sales+in+2003&sPath=[Sales+Re
ports]&sType=rpt&sInstance=User
```

sKind

Syntax	Description	Values
`sKind`	Specifies the file type of the target Desktop Intelligence document. Use in conjunction with `sDocName` and `sPath`.	• `FullClient`

Note:
Only mandatory if the target is a Desktop Intelligence document. Otherwise, use `sType`.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?sDocName=Sales+in+2001&sPath=[Sales+Re
ports]&sKind=FullClient
```

sPath

Syntax	Description	Values
`sPath`	Specifies the name of the folder and subfolder containing the target document. Use in conjunction with `sDocName` and `sType`.	Folder and/or subfolder: `[folder],[subfolder]`

`sPath` is used only with subfolders of the `Public Folders` folder. If your document is outside of the `Public Folders` folder, for example in the `My Favorites` folder, use the `iDocID` parameter instead of `sPath` and `sDocName`.

Do not add `[Public+Folders]` to the path; start with the name of the first subfolder within `Public Folders`.

```
Public Folders
    folder 1
      folder 1.1
         folder 1.1.1
```

If your document were in folder 1.1.1, you would set `sPath` to: `[folder+1],[folder+1.1],[folder+1.1.1]`.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?sDocName=Sales+in+2003&sPath=[Sales+Re
ports]&sType=rpt
```

sType

Syntax	Description	Values
`sType`	Specifies the file type of the target document. Use in conjunction with `sDocName` and `sPath`.	• `wid` • `rpt` • `car`

Note:
This parameter is ignored for agnostic documents.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?sDocName=Sales+in+2003&sPath=[Sales+Re
ports]&sType=rpt
```

Input parameters

lsC

Syntax	Description	Values
`lsC`	Specifies a contextual prompt if there is an ambiguity during SQL generation. **Note:** Only supported by Web Intelligence documents.	A prompt value that resolves the ambiguity in the SQL generation.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&lsC=Sales
```

lsM[NAME]

Syntax	Description	Values
`lsM[NAME]`	Specifies multiple values for a prompt. `[NAME]` is the text of the prompt.	- Multiple prompt values, separated by a comma. - `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>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&sRefresh=Y&lsMSelect+Cities=[Paris],[London]
```

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&sRefresh=Y&lsMparamStringDR=[c],[d]&lsMparam
NumberDR=[3],[4]&lsMparamDate
DR=[Date(2003,6,3)],[Date(2003,6,4)]&lsMparamDateTime
DR=[DateTime(2003,6,1,3,1,1)],[DateTime(2003,6,1,4,1,1)]
```

Crystal reports

If the target is a Crystal report, each value must be enclosed in square brackets.

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.

Olap Intelligence reports

If the target document is an OLAP Intelligence report (.car) you can use the IsM parameter to specify prompts. The parameters are passed in as a URL-encoded string using the unique name of the parameter set up in the OLAP Intelligence report.

Example: Setting a memberset parameter

```
http://<servername>:<port>/OpenDocument/<platformSpecif
ic>?iDocID=544&sIDType=InfoObject&sType=car&lsMADC216EA-
D9A5-42B5-AE%2C21%2C84%2CA9%2CF9%2C6E%2C31%2C7=[%5BCus
tomers%5D.%5BCountry%5D.%26%5BMexico%5D],[%5BCus
tomers%5D.%5BCountry%5D.%26%5BCanada%5D]
```

This example opens up an OLAP Intelligence report with a memberset parameter to Customers > Country > Mexico and Customers > Country > Canada in the view.

lsR[NAME]

Syntax	Description	Values
`lsR[NAME]`	Specifies a range of values for a prompt. `[NAME]` is the text of the prompt. **Note:** Not supported by OLAP Intelligence reports.	- 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>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&sRefresh=Y&lsRTime+Period:=[2000..2004)
```

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&sRefresh=Y&lsRparamStringDR=[h..i]&lsRparam
NumberDR=[7..8]&lsRparamCurrencyDR=[3..4]&lsRparamDate
DR=[Date(2003,6,7)..Date(2003,6,8)]&lsRparamDateTime
DR=[DateTime(2003,6,1,7,1,1)..Date
Time(2003,6,1,8,1,1)]&lsRparamTime
DR=[Time(1,1,7)..Time(1,1,8)]&lsRparamUnbound1=(..6)&lsR
paramUnbound2=[6..)&lsRparamStringR=[a..d]&lsRparamNum
berR=[1..3]&lsRparamCurrencyR=[1..3]&lsRparam
DateR=[Date(2003,6,1)..Date(2003,6,3)]&lsRparamDate
TimeR=[DateTime(2003,6,1,1,1,1)..Date
Time(2003,6,1,3,1,1)]&lsRparam
TimeR=[Time(1,1,1)..Time(3,1,1)]
```

Crystal reports

If the target is a Crystal report, 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).

lsS[NAME]

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)

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>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&sRefresh=Y&lsSparamString=h&lsSparamNumber=1&
lsSparamCurrency=121&lsSparamDate=Date(2003,6,11)&
lsSparamDateTime=DateTime(2003,6,11,14,38,37)&lsSparam
Boolean=false&
lsSparamTime=Time(12,39,2)&lsSparamStringDR=a&lsSparam
DateDR=Date(2003,6,1)
```

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.

OLAP Intelligence reports

If the target document is an OLAP Intelligence report (.car) you can use the lsS parameter to specify prompts. The parameters are passed in as a URL-encoded string using the unique name of the parameter set up in the OLAP Intelligence report.

Example: Opening an OLAP report to a specific page

If `23CAA3C1-8DBB-4CF3-BA%2CB8%2CD7%2CF0%2C68%2CEF%2C9C%2C6F` is the URL-encoded unique name for the page parameter in the OLAP Intelligence report, you would use the following URL to open the OLAP Intelligence report to page 2:

```
http://<servername>:<port>/OpenDocument/<platformSpecif
ic>?iDocID=440&sIDType=InfoObject&sType=car&lsS23CAA3C1-
8DBB-4CF3-BA%2CB8%2CD7%2CF0%2C68%2CEF%2C9C%2C6F=2
```

Example: Setting a cube parameter

If `8401682C-9B1D-4850-8B%2C5E%2CD9%2C1F%2C20%2CF8%2C1%2C62` is the URL-encoded unique name for the cube parameter opening the warehouse cube in the catalogue FoodMart 2000 on MSAS, you would use the following URL to open this cube parameter:

```
http://<servername>:<port>/OpenDocument/<platformSpecif
ic>?iDocID=616&sIDType=InfoObject&sType=car&lsS8401682C-
9B1D-4850-8B%2C5E%2CD9%2C1F%2C20%2CF8%2C1%2C62=CATA
LOG%3DFoodMart%202000,CUBE%3Dwarehouse
```

sPartContext

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

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

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&sReportPart=Part1&sPartContext=0-4-0
```

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

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>/OpenDocument/opendoc/<platformSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sIDType=CUID&sRefresh=Y
```

Crystal reports

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

sReportMode

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

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

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&sReportPart=Part1&sReportMode=Part
```

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 OLAP Intelligence reports.

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

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&sReportName=First+Report+Tab
```

sReportPart

Syntax	Description	Values
`sReportPart`	Specifies the part of the target Crystal report to open. **Note:** Only supported by Crystal reports.	Name of the Crystal report part.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&sReportPart=Part1
```

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

Output parameters

NAII

Syntax	Description	Values
NAII	Indicates whether to force the display of the prompt selection page. **Note:** Only supported by Web Intelligence documents.	- Y (prompt values that are passed with lsS, lsM, or lsR in the URL are applied and not displayed in the "Prompts" 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>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&lsSYear=FY1999&NAII=Y&sRefresh=Y
```

sOutputFormat

Syntax	Description	Values
`sOutputFormat`	Specifies the format in which to open the target document.	• `H` (HTML) • `P` (PDF) • `E` (Excel - Crystal Reports only) • `W` (Word - Crystal Reports only)

Note:
Defaults to HTML if this parameter is not specified.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&sOutputFormat=E
```

sViewer

Syntax	Description	Values
`sViewer`	Specifies the selected report viewer.	• `html` • `part` (Crystal reports only) • `actx` (Crystal reports only) • `java` (Crystal reports only)

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<plat
formSpecific>?iDocID=Aa6GrrM79cRAmaOSMGoadKI&sID
Type=CUID&sViewer=html
```

Crystal Reports URL Reporting

chapter 2

Getting started

About this documentation

This documentation provides you with information for constructing parameterized URLs that link to Crystal reports in an BusinessObjects Enterprise system. A command reference, including syntax and usage examples, is provided for each URL command.

For information about deploying the CrystalReports web application (that contains URL reporting) after the installation of BusinessObjects Enterprise, see the *BusinessObjects Enterprise Web Application Deployment Guide*.

Who should use this documentation?

This documentation is for anyone creating URLs to Crystal reports with the URL reporting syntax. We recommend consulting this guide if you are:

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

Familiarity with the management and organization of the reports in your BusinessObjects Enterprise deployment, as well as knowledge about Crystal Reports design concepts are beneficial.

Note:
URL reporting only supports Crystal reports (.rpt). If you want to create URLs to additional document formats, such as Web Intelligence documents, Voyager workspaces, InfoView dashboards, or Xcelsius SWF files, use the OpenDocument web application and its URL syntax and parameters. For more information on OpenDocument, see *About OpenDocument*.

About Crystal Reports URL reporting

Crystal Reports URL reporting (`viewrpt.cwr`) is one of many deployed web applications within an BusinessObjects Enterprise system. It processes incoming URL requests for Crystal reports in the Central Management Server (CMS), and delivers the correct report to the end user in the appropriate viewer. This allows you to send users direct links to a report and avoid having them navigate through a folder hierarchy such as in InfoView. The URL reporting syntax and its commands allow you to construct URLs that link to these reports. For example, consider the following URL:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1783
```

This URL accesses the report in the CMS with the unique identifier of `1783` and the report is rendered to the end user in a default Crystal Reports viewer. In this example, `id` is one of many URL commands. These commands specify how to access a particular report in the CMS, or determine how to display the report to the end user. You can also automatically assign values for report database authentication, parameter prompts, and selection formulas.

The Crystal Reports designer provides a GUI-based editor to help you create and embed hyperlinks to other reports and documents stored in the CMS. Consult the *Crystal Reports User's Guide* for information on this feature.

What's new with Crystal Reports URL reporting

URL structure change in URL reporting

The virtual directory `businessobjects` is no longer created during installion of BusinessObjects Enterprise. This affects the URL syntax for Crystal Reports URL reporting.

The previous URL syntax was:

```
http://<servername>:<port>/businessob
jects/viewrpt.cwr?<command1>&<command2>&...<commandN>
```

The new URL syntax is:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?<command1>&<command2>&...<commandN>
```

Links to reports must use the new URL syntax, or you must create the appropriate `businessobjects` virtual directory.

For information on supporting legacy URL Reporting applications with Business Objects XI 3.0, refer to SAP Note ID: 1197099, on the SAP Notes Database, here: *https://websmp208.sap-ag.de/notes*.

Please note that you will need your SAP Service Marketplace USER ID and PASSWORD to access this material. If you do not have the necessary credentials contact your SAP support center: *https://web smp202.sapag.de/~sapidp/011000358700000560361996E/*.

sReportMode

`sReportMode` is a new URL command to view a Crystal Report in different modes depending on the option passed to the command. Possible values include:

- `part` displays part of a report using the parts viewer to render the report
- `printlayout` displays the report in a print preview layout
- `weblayout` displays the report as a web page layout.

Interactive Parameters

Parameters passed into the URL are always be applied to the report, even if the report instance contains saved data. This is a change in behavior from previous releases.

Migrating your links

In previous versions, URL Reporting was managed from the root folder and therefore a request to `http://<servername>/viewrpt.cwr` or to any virtual folder was supported. To increase security, the access of the request has been reduced to a specific virtual folder. Due to this change, applications that use URL reporting to link to reports in Crystal Enterprise deployments need to be updated to reference the specific BusinessObjects Enterprise virtual folder.

The default location is:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr
```

If you do not want to change the calling application, you can configure the web server to redirect requests to the default `viewrpt.cwr` virtual folder location. For more information, refer to your web server or web application server documentation.

URL syntax

Basic URL syntax

The following sections explains how to use URL reporting, and how to construct the URL.

A URL reporting URL is generally structured as follows:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?<command1>&<command2>&...&<commandN>
```

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 server where `viewrpt.cwr` is hosted in place of `<servername>` and you must use the correct port number in place of `<port>` to access the `viewrpt.cwr` web application.

Deployment

BusinessObjects Enterprise can operate with a Java Application server or a .NET application server. Depending on where the application is configured the server name and port number will be dependant on the web server, however the calling convention is application server agnostic.

URL syntax considerations

You can pass URL reporting commands in any combination and order. All commands are optional - except the `id` command. If you do not specify any optional commands, the default viewer displays the report and prompts the user for any required information.

Remember that a number of factors determine whether the user is prompted for information when accessing a Crystal report by URL. The user is prompted under the following circumstances:

- The report requires the user to enter parameter values or authentication information.
- The report does not contain saved data; it needs to access a database.
- The user has refreshed the report; it needs to access a database.
- The values for the prompts have not already been set or the prompts have been enabled through the SDK or CMC.
- If `apstoken` or `apsuser`, `apspassword`, and `apsauthtype` values are not provided, the user is prompted to log on to the Central Management Server (CMS).

Command reference

This section provides details about the available URL reporting commands, their specific uses, and relevant examples.

Table 2-1: Authentication Commands

Command	Description
apstoken	Specifies a valid logon token for the current Enterprise session.

Command	Description
apsuser, apspassword, apsauthtype	Specifies authentication credentials for logging on to a CMS.
connect	Re-establishes a connection to the Page Server.
PASSWORD (see *user# and password#* and *user and password* for details)	Specifies logon credentials for the database that is used by the report and its subreports.
USER (see *user# and password#* and *user and password* for details)	Specifies logon credentials for the database that is used by the report and its subreports.

Table 2-2: Document Identifier Commands

Command	Description
id	Specifies the unique identifier of the viewable document in the CMS.
rptsrc	Specifies a session variable that references a report source object.

Table 2-3: Input Commands

Command	Description
gf	Specifies a group selection formula for the report.
prompt# (Use Case 1)	Specifies values for parameter fields in a report. It is recommended that you use `PROMTPEX` instead.
promptex (Use Case 1)	Specifies values for parameter fields in a report and subreport. See also *promptex (Use Case 2)*, *promptex (Use Case 3)*, and *promptex#*.
promptOnRefresh	Indicates whether the report should prompt for parameter field values when refreshed.
sf	Specifies a selection formula to further filter records by.
sPartContext	Specifies the data context of a report part. Use in conjunction with `sReportPart`.
sReportMode	Specifies the mode to display the report in.
sReportPart	Specifies the part of the target report to view.

Table 2-4: Output Commands

Command	Description
cmd and EXPORT_FMT	Specifies that the report be exported to the indicated format. Used in conjunction with `EXPORT_OPT`.
EXPORT_OPT	Specifies the range of pages in the report to export. Used in conjunction with `cmd=EXPORT` and `EXPORT_FMT`.
init	Specifies the viewer to display the report with.
sZoom	Specifies the magnification percentage to display the report at.

Authentication commands

apstoken

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

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

Example:

This example uses the BusinessObjects Enterprise Java SDK to pass in a logon token to the URL. For more information on the `ILogonTokenMgr.createLogonToken` method, see the *BusinessObjects Enterprise Java API Reference*. You can create logon tokens in a similar fashion using other BusinessObjects Enterprise SDK platforms such as .NET and Web Services.

```
String viewReportURLToken() throws SDKException, Unsup
portedEncodingException
{
 IEnterpriseSession sess = CrystalEnterprise.getSession
Mgr().logon ("username", "password", "<cms>:<port>",
"secEnterprise");
 String token = sess.getLogonTokenMgr().createLogonTo
ken("",120,100);
 String tokenEncode = URLEncoder.encode(token, "UTF-8");

 return ("http://<server>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&apstoken=" + tokenEncode);
}
```

apsuser, apspassword, apsauthtype

Syntax	Description	Values

`apsuser`
`apspassword`
`apsauthtype`
	Specifies authentication credentials for logging on to a CMS.	Valid user name, password, and authentication type (`secEnterprise`, `secLDAP`, `secWinAD`) for logging onto the CMS.

You may need to use these commands under special circumstances, such as when a user receives a report through email and must log on to the CMS to view it. In most cases, however, it is recommended to use the `apstoken` command to pass a valid Enterprise session to your URL.

Example:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&apsuser=JLee&apspassword=se
cret&apsauthtype=secEnterprise
```

connect

Syntax	Description	Values
`init=<viewer>:connect`	Re-establishes a connection to the Page Server.	N/A

The `connect` command re-establishes a connection to the Page Server and must be appended to the INIT command. By re-establishing a connection to the Page Server, the `connect` command allows the user to reset the report's parameters and logon information, and re-process the report if necessary—without the need to start a new browser session.

That is, if you use viewer A to display a report, and then you specify viewer B to view the same report in the same browser session, you will not be prompted for parameter values or database logons, and a new report job will not be opened. But, if you specify ":connect" along with the request for viewer B, the connection to the Page Server will be re-established. That means, if necessary, the user will be prompted for parameter values and logon information, and the report will be run again.

Note:
When you re-establish a connection to the Page Server with the `connect` command, the report is not refreshed against the database. While the connection to the Page Server enables the user to reset parameter values and view a different set of information, if the report contains saved data, it will not access the database for that information. For more details on refreshing a report, see the *BusinessObjects Enterprise Java SDK Developer Guide*.

Example:

This example specifies that the report will re-establish its connection to the Page Server once the URL has been processed:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&init=java:connect
```

user and password

Syntax	Description	Values

`user-<server name>.<database name>`
`user-<server name>.<database name>@<subreport name>`
`password-<server name>.<database name>`
`password-<server name>.<database name>@<subreport name>`
	Specifies logon credentials for the database that is used by the report and its subreports.	Databse name, server name, user name, and password.

Note:
Sending a password over the URL is not secure. It is strongly recommended that the database logon information is set through the Central Management Console (CMC).

Example:

This example shows how to pass the following values to the primary report:

- Server name: "systemdsn".
- Database name: "xtreme".
- user name: "vantech".
- password: "1234".

Note:
For Oracle databases, substitute the schema name for the database name.

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&user-systemdsn.xtreme=van
tech&password-systemdsn.xtreme=1234
```

This example shows how to pass the following values to the subreport:

- Server name: "systemdsn".
- Database name: "pubs".
- User name: "vantech".
- Password: "1234".
- Subreport: "sr".

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&user-systemdsn.pubs@sr=van
tech&password-systemdsn.pubs@sr=vantech
```

user# and password#

Syntax	Description	Values

`user#`
`password#`
`user#@subreportname`
`password#@subreportname`
	Specifies logon credentials for the database that is used by the report and its subreports.	Database user name and password.

Note:
Sending a password over the URL is not secure. It is strongly recommended that the database logon information is set through the Central Management Console (CMC).

Example:

This example passes user name "msmith" and the password "1234" to the report:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&user0=msmith&password0=1234
```

This example shows how to pass the user name "msmith" and password "1234" to the subreport called "Crosstab":

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&user0@Crosstab=msmith&pass
word0@Crosstab=1234
```

If the report accesses more than one password-protected database, you can pass multiple user names and passwords, by incrementing the `user` and `password` index number:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&user0=msmith&password0=1234&us
er1=bsmith&password1=1234
```

Note:
You can specify passwords in the URL in any order. For example, `password1` can appear before `password0`. However, index numbers must match the order of password-protected databases that appear in the report.

Document identifier commands

id

Syntax	Description	Values
`id`	Specifies the unique identifier of the viewable document in the CMS.	A numerical identifier associated with the document in the CMS.

You can see identifier values for a document within the Central Management Console (CMC) or InfoView applications. The properties page for each document contains the document ID. You can also obtain the identifier programmatically using the BusinessObjects Enterprise SDK. For example, in the Java SDK the `com.crystaldecisions.sdk.occa.infostore.IInfoObject` interface contains a `getID` method which you can pass to the URL.

Example:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152
```

rptsrc

Syntax	Description	Values
rptsrc	Specifies a session variable that references a report source object.	Name of the report source.

You can save a programmatic report source as a session variable in your web application and pass this session variable to `viewrpt.cwr` using the `rptsrc` command. Use this command if you are programmatically modifying the report at runtime before redirecting to a `viewrpt.cwr` URL. Otherwise, it is recommended that you use the `id` command instead to access a report in the CMS. The report source class and method to retrieve it depends on the platform and SDK you are developing your application with. For example:

- Using the Report Application Server (RAS) Java SDK:

  The `com.crystaldecisions.sdk.occa.report.application.ReportClientDocument.getReportSource` method returns a `com.crystaldecisions.sdk.occa.report.reportsource.IReportSource` report source object.

- Using the Report Application Server (RAS) .NET SDK:

  The `CrystalDecisions.ReportAppServer.ClientDoc.ReportClientDocument.ReportSource` property returns a `CrystalDecisions.ReportAppServer.Controllers.ReportSource` report source object.

Note:
You must deploy and configure your custom web application that retrieves and saves the report source object along with the CrystalReports web application that contains `viewrpt.cwr`. The two applications must be able to share the same session data for the `rptsrc` command to work. For information about deploying the CrystalReports web application after the installation of BusinessObjects Enterprise, see the *BusinessObjects Enterprise Web Application Deployment Guide*.

Example:

This example code snippet assumes you already have an `IReportSource` java object called `reportSource` and are saving it as a session variable in a JSP page.

```
...
session.setAttribute("rs", reportSource);
response.sendRedirect("http://<server>:<port>/CrystalRe
ports/viewrpt.cwr?&rptsrc=rs");
```

Input commands

gf

Syntax	Description	Values
`gf`	Specifies a group selection formula for the report.	A valid Crystal Reports group selection formula.

Note:

- Pages are shared between reports that have the same `sf` and `gf` commands applied and that do not require logon information.
- You cannot use the `gf` command with the DHTML viewer. You must specify the `init` command in your URL and choose the ActiveX or Java viewer.

Example:

This example shows how to pass a group selection formula that selects all groups where the sum of all customer sales in each region is greater than 10,000:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&init=java&gf=Sum({cus
tomer.Sales},{customer.Region})>10000
```

prompt# (Use Case 1)

Syntax	Description	Values
`prompt#`	Specifies each parameter by value. Parameter values are specified that way in earlier versions of Crystal Reports (for example, Crystal Reports 7). While it is not recommended, parameter values can still be specified that way.	A string (potentally empty) that is the new value of the prompt. Values are assigned to parameters in the same order that they appear in the report. Do not use quotation marks around parameter values to indicate string values.

If the report contains more than one parameter field, you can pass multiple values to parameters by incrementing the `prompt#` index value. For example, `prompt0=CA&prompt1=1000`. You can pass NULL values to a parameter by leaving the right-hand side of the statement blank. For example, `prompt0=&prompt1=1000` sets `prompt0` to NULL. You can specify prompts in the URL in any order; for example, `prompt1` can appear before `prompt0`. However, index numbers must match the order of the prompts that appear in the report.

Note:

- Parameters passed into the URL are always be applied to the report, even if the report instance contains saved data.

- Reports that have the `prompt#` command applied do not have their pages shared. Caching will be by user. That is, a page that is stored in the cache is reserved for the user who last viewed it.
- The `prompt#` command can only be used to pass values to parameters in the main report. You must use the `prompt` command or the `promptex` command to pass values to parameters in a subreport.

Example:

This example uses `prompt#` to pass "CA" as a value to the first parameter:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&prompt0=CA
```

prompt# (Use Case 2)

Syntax	Description	Values
`prompt#=Date(YYYY,MM,DD)`	Specifies Date or DateTime parameter values. Parameter values are specified that way in earlier versions of Crystal Reports (for example, Crystal Reports 7). While it is not recommended, parameter values can still be specified that way.	Date or DateTime parameter values. For single value Date or DateTime parameters, the `prompt#` command does not require double quotes.

Note:

- Parameters passed into the URL are always be applied to the report, even if the report instance contains saved data.
- Reports that have the `prompt#` command applied do not have their pages shared. Caching will be by user. That is, a page that is stored in the cache is reserved for the user who last viewed it.

- The `prompt#` command can only be used to pass values to parameters in the main report. You must use the `prompt` command or the `promptex` command to pass values to parameters in a subreport.

Example:

This example passes a Date value of February, 02, 2002 for the second parameter within a report:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&prompt2=Date(2002,02,02)
```

promptex (Use Case 1)

Syntax	Description	Values

`promptex-<promptname>`
`promptex-<promptname>@<subrpt>`
	Specifies values for a parameter by name.	`<promptname>` and `<subrpt>` are non-empty strings that represent names of a parameter field prompt and a subreport, which are defined in the report. `<value>` is a single string.

Note:
Parameters passed into the URL are always be applied to the report, even if the report instance contains saved data.

Example:

This example passes "hello" as a value for the parameter called "sample":

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&promptex-sample="hello"
```

This subreport example passes "hello" as a value for the parameter called "sample" for the subreport called "mysubrpt":

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&promptex-sample@mysubrpt="hel
lo"
```

Note:

- If an existing report is inserted as the subreport, the subreport name includes the file extension (.rpt). However, the file extension may be missing from the subreport name, if the subreport was created inside the main report (using the Report Expert to create the new report, and then using Insert Subreport). In that case, the subreport name appears as "user0@subreportname," unless an extension is added in the "Report Name" text box of the Insert Subreport dialog box.
- A backslash (\) acts as an escape, so it is substituted by the character that follows it. Quotation marks and backslashes must be escaped because they are reserved URL characters. You must escape "@", "." or "\" when they are used in the subreport name, server name, database name or parameter name.

promptex (Use Case 2)

Syntax	Description	Values

`promptex-sample="<valueA>","<value B>","<value C>"`
`promptex-sample=["<valueA>"-"<value B>"]`
	Specifies multiple values to a parameter.	`<promptname>` and `<subrpt>` are non-empty strings that represent names of a parameter field prompt and a subreport, which are defined in the report. `<value A>`, `<value B>`, and `<value C>` are strings. See table below for interval bounding.

Note:
Parameters passed into the URL are always be applied to the report, even if the report instance contains saved data.

Example:

This example below specifies "Apples, Oranges, and Grapes" as values for the parameter called "fruits":

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&promptex-fruits="Apples","Or
anges","Grapes"
```

Example:

A square bracket indicates that the interval is closed at that end, and that the specified number is included in the range; a round bracket indicates that the interval is open at that end, and that the specified number is not included in the range. For example:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&promptex-sample=("5"-"11")
```

The round brackets specify a range of all values between 5 and 11, but does not include 5 and 11.

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&promptex-sample=["5"-"11")
```

The combination of a square bracket and round bracket specifies a range of all values between 5 and 11, which includes 5 but not 11.

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&promptex-sample=(-"11")
```

The brackets and minus sign specifies a range of all values up to, but not including, 11.

The following table lists the types of bounded and unbounded intervals you can use.

Bounded interval	Unbounded intervals
["<value>"-"<value>"]	("<value>"-)
("<value>"-"<value>"]	["<value>"-)
["<value>"-"<value>")	(-"<value>")
("<value>"-"<value>")	(-"<value>"]

promptex (Use Case 3)

Syntax	Description	Values

`promptex-<prompt name>="Date(YYYY,MM,DD)"`
`promptex-<prompt name>=["Date(YYYY,MM,DD)"-“Date(YYYY,MM,DD)"]`
	Specifies Date or DateTime parameter values, using the Single Value or Date Range methods.	Date or datetime parameters passed. A specific date or date range can be passed. For single value Date or DateTime parameters, double quotes are required.

Note:
Parameters passed into the URL are always be applied to the report, even if the report instance contains saved data.

Example:

To pass a Date value of February, 02, 2002 for the "birthdate" parameter, use the following URL command:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&promptex-birth
date="Date(2002,02,02)"
```

Example:

This example shows that "DateRangeParameter" is the parameter name; the square brackets that surround the values indicate that the specified date is included in the range:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&promptex-DateRangeParame
ter=["date(1996,02,18)"-"Date(1996,09,10)"]
```

The type of brackets that surround the date value can specify whether the value should be included or excluded from the date range:

- Square brackets [ ] that surround the values indicate that the specified date is included in the range.
- Round brackets ( ) that surround the values indicate that the specified date is excluded in the range.

promptex#

Syntax	Description	Values
`promptex#`	The `promptex#` command is an enhanced version of the older `prompt#` command. In the enhanced notation, quotation marks are used around parameter values to indicate string values. All parameter values are passed to the report as strings, and intended numeric values are translated from strings to numbers by the report.	`<promptname>` and `<subrpt>` are non-empty strings that represent names of a parameter field prompt and a subreport, which are defined in the report. `<value>` is a single or multivalued string.

If the report contains more than one parameter field, you can pass multiple values to parameters by incrementing the `prompt#` index value. For example: `promptex0="CA"&promptex1="1000"`. You can specify prompts in the URL in any order; for example, promptex1 can appear before promptex0. However, index numbers must match the order of the prompts that appear in the report.

Note:

- Parameters passed into the URL are always be applied to the report, even if the report instance contains saved data.
- Reports that have the `promptex#` parameter applied do not have their pages shared. Caching will be by user. That is, a page that is stored in the cache is reserved for the user who last viewed it.
- The `promptex#` command can only be used to pass values to parameters in the main report. You must use the `prompt` command or the `promptex` command to pass values to parameters in a subreport.

Example:

This example passes "CA" as a value for the first parameter in the report:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&promptex0="CA"
```

promptOnRefresh

Syntax	Description	Values
`promptOnRefresh`	Indicates whether the report should prompt for parameter field values when refreshed.	Value must be either 0 or 1. 0 is for false and 1 is for true.

Example:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&promptOnRefresh=1
```

sf

Syntax	Description	Values
`sf`	Specifies a selection formula to further filter records by.	A valid Crystal Reports selection formula.

Selection formulas that are passed through the URL with the `sf` command will be appended to selection formulas that are already contained in the report. That is, the generated report will be based first on existing selection formulas saved with the report, and the selection formula specified by the sf command will be applied against that set of records.

For example, assume a report already contains a selection formula that selects the records for film studios in the state of California. The sf command is then used to append a formula that selects the records for a particular studio such as "Universal". Information on that particular studio will be displayed if there are studios with a value of "Universal" in the state of California. However, if the sf command specifies a studio value that does not exist in the subset of records already selected according to the state of California, the requested report would contain no data.

Note:
The new selection formula is not saved with the original report file. It is only valid for the current URL request.

Example:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&sf={studio.Studio}&=&'Univer
sal'
```

sPartContext

Syntax	Description	Values
`sPartContext`	Specifies the data context of a report part. Use in conjunction with `sReportPart`.	The name of the report part data context.

Note:
The `sReportPart` and `sPartContext` commands are only supported with the DHML parts viewer (`init=part`).

Example:

The following example specifies the data context of a report part:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&sPartContext=/USA/CA
```

sReportMode

Syntax	Description	Values
`sReportMode`	Specifies the mode to display the report in.	• `part` • `printlayout` • `weblayout`

Note:

- The default value when using this parameter is `printlayout`. Therefore, if the incorrect value is given, the command will use the default display mode.
- `sReportMode` will only be applicable when `init=html` or `init=dhtml` or the default viewer is selected to be dhtml from web.xml.
- `sReportMode=part` when `init=html` or default viewer is set to `html` in `web.html` is the same as saying `init=part` in the URL.

Example:

The following example allows us to view part of the report:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&init=html&sReportMode=part
```

sReportPart

Syntax	Description	Values
`sReportPart`	Specifies the part of the target report to view.	Name of the report part.

Note:
The `sReportPart` and `sPartContext` commands are only supported with the DHML parts viewer (`init=part`).

Example:

The following example specifies the report part to be viewed:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&sReportPart=graph3
```

Output commands

cmd and EXPORT_FMT

Syntax	Description	Values

`cmd=EXPORT`
`EXPORT_FMT=<EXPORT_FMT representation>`
	Specifies that the report be exported to the indicated format. Used in conjunction with `EXPORT_OPT`.	See table below for export format values.

Table 2-26: Export Formats

Export Format	Export_FMT Representation
PDF	U2FPDF:0
Crystal Reports (RPT)	U2FCR:0
Microsoft Excel (97-2003)	U2FXLS:3
Microsoft Excel (97-2003) Extended	U2FXLS:4
Rich Text Format (RTF)	U2FRTF:0
Microsoft Word - Editable (RTF)	U2FRTF:1
Microsoft Word (97-2003)	U2FWORDW:0
XML	U2FXML:0

Example:

The following example exports a report to Rich Text Format (RTF):

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&cmd=EXPORT&EXPORT_FMT=U2FRTF:0
```

EXPORT_OPT

Syntax	Description	Values
`EXPORT_OPT`	Specifies the range of pages in the report to export. Used in conjunction with `cmd=EXPORT` and `EXPORT_FMT`.	A valid page range of integers enclosed in square brackets, in the format `[firstPage-lastPage]`. The value of `firstPage` must be less than the value of `lastPage`.

Note:
If no value is specified, the whole report is exported by default. This is equivalent to setting the value to "[-]".

Example:

The following example exports the first four pages of a report to Rich Text Format (RTF):

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&cmd=EXPORT&EX
PORT_FMT=U2FRTF:0&EXPORT_OPT=[1-4]
```

init

Syntax	Description	Values
`init`	Specifies the viewer to display the report with.	• `actx` (ActiveX viewer) • `java` (Java using browser JVM) • `dhtml` (DHTML viewer) • `part` (DHTML parts viewer)

Note:

- If no value is specified, the DHTML viewer is used to display the report by default.
- The DHTML and DHTML parts viewers correspond to both the Java and .NET Web Form versions.

Example:

This example specifies that the Java viewer is used to view the report:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&init=java
```

sZoom

Syntax	Description	Values
`sZoom`	Specifies the magnification percentage to display the report at.	An integer value for the magnification percentage. Defaults to 100 if not specified.

Example:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=1152&sZoom=50
```

More Information

A

appendix

Information Resource	Location
SAP BusinessObjects product information	http://www.sap.com
SAP Help Portal	Select http://help.sap.com > SAP BusinessObjects. You can access the most up-to-date documentation covering all SAP BusinessObjects products and their deployment at the SAP Help Portal. You can download PDF versions or installable HTML libraries. Certain guides are stored on the SAP Service Marketplace and are not available from the SAP Help Portal. These guides are listed on the Help Portal accompanied by a link to the SAP Service Marketplace. Customers with a maintenance agreement have an authorized user ID to access this site. To obtain an ID, contact your customer support representative.
SAP Service Marketplace	http://service.sap.com/bosap-support > Documentation • Installation guides: https://service.sap.com/bosap-inst guides • Release notes: http://service.sap.com/releasenotes The SAP Service Marketplace stores certain installation guides, upgrade and migration guides, deployment guides, release notes and Supported Platforms documents. Customers with a maintenance agreement have an authorized user ID to access this site. Contact your customer support representative to obtain an ID. If you are redirected to the SAP Service Marketplace from the SAP Help Portal, use the menu in the navigation pane on the left to locate the category containing the documentation you want to access.
Developer resources	https://boc.sdn.sap.com/ https://www.sdn.sap.com/irj/sdn/businessobjects-sdklibrary

Information Resource	Location
SAP BusinessObjects articles on the SAP Community Network	*https://www.sdn.sap.com/irj/boc/businessobjects-articles* These articles were formerly known as technical papers.
Notes	*https://service.sap.com/notes* These notes were formerly known as Knowledge Base articles.
Forums on the SAP Community Network	*https://www.sdn.sap.com/irj/scn/forums*
Training	*http://www.sap.com/services/education* From traditional classroom learning to targeted e-learning seminars, we can offer a training package to suit your learning needs and preferred learning style.
Online customer support	*http://service.sap.com/bosap-support* The SAP Support Portal contains information about Customer Support programs and services. It also has links to a wide range of technical information and downloads. Customers with a maintenance agreement have an authorized user ID to access this site. To obtain an ID, contact your customer support representative.
Consulting	*http://www.sap.com/services/bysubject/businessobjectsconsulting* Consultants can accompany you from the initial analysis stage to the delivery of your deployment project. Expertise is available in topics such as relational and multidimensional databases, connectivity, database design tools, and cus tomized embedding technology.

Index