Viewing Reports and Documents using URLs

2009-05-08

Contents

What's new in URL Reporting

chapter 1

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.

URL structure change in URL reporting

The installation of Crystal Reports no longer creates the virtual directory `businessobjects`. This affects both the URL structure used in URL reporting and the URL structure used with openDocument.

When performing URL reporting for Crystal Reports, the URL structure has changed. The previous URL structure was:

```
http://<servername>:<port>/businessobjects/viewrpt.cwr?<com
mand1>&<command2>&...<commandN>
```

The new URL structure is:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?<com
mand1>&<command2>&...<commandN>
```

To ensure that your application runs correctly, you must use the new URL structure or create the `businessobjects` virtual directory.

When performing URL reporting with openDocument, the URL structure has changed. The previous URL structure was:

```
http://<servername>:<port>/businessobjects/enterprise115/<plat
formSpecific>?<parameter1>&<parameter2>&...&<parameterN>
```

The new URL structure is:

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

To ensure that your application runs correctly, you must use the new URL structure or create the `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://websmp202.sapag.de/~sapidp/011000358700000560361996E/*.

1 What's new in URL Reporting

Overview

chapter 2

This section provides information about viewing reports and documents using URL reporting for Crystal Reports and URL reporting using openDocument.

Should I use URL reporting for Crystal reports or URL reporting using openDocument?

URL reporting for Crystal reports and URL reporting with openDocument both provide a fast way to work with reports using URLs.

You should use URL reporting for Crystal reports under these circumstances:

- You intend to use URL reporting with Crystal reports only. For example, you do not need to work with OLAP Intelligence reports or Web Intelligence documents.
- For more information, see *URL reporting for Crystal Reports* on page 11.

You should use URL reporting using openDocument under these circumstances:

- You intend to use URL reporting with Crystal reports, OLAP Intelligence reports, Desktop Intelligence documents, or Web Intelligence documents.
- You need multi-format support. For example, you plan to use URLs to view Crystal reports, OLAP Intelligence reports, Desktop Intelligence documents, or Web Intelligence documents.
- You want to create links between reports and documents.
- For more information, see *URL Reporting using openDocument* on page 49.

URL reporting for Crystal Reports

chapter 3

URL reporting for Crystal reports (URL reporting) provides URL access to Crystal reports by passing a URL string to a BusinessObjects Enterprise server. URL reporting provides commands to control how reports are generated and displayed.

For example, rather than have the user prompted for information, you can automatically assign values for database authentication, parameters, selection formulas and for other features.

This section provides a brief overview of URL reporting and includes a list of available commands. Details about the available commands, their specific uses, and relevant examples are also provided.

Note:

- The commands listed in this section are compatible with the commands used with viewrpt.cwr shipped in previous versions of Crystal Enterprise.
- URL reporting for Crystal reports can only be used with Crystal reports. Other report formats, including OLAP Intelligence reports and Web Intelligence documents are not supported. openDocument provides support for Crystal reports, OLAP Intelligence reports and Web Intelligence documents.

Related Topics

- *URL Reporting using openDocument* on page 49

Structuring a Crystal Reports URL

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

URL structure

A URL reporting URL is generally structured as follows:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?<com
mand1>&<command2>&...&<commandN>
```

- In a Java environment, `viewrpt.cwr` is a servlet mapping to `viewrpt`. This servlet is configured in a `web.xml` file installed with InfoView (`/CrystalReports/WEB-INF/web.xml`).
- In a .NET environment, `viewrpt.cwr` is redirected to `viewrpt.aspx`.

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.

Migration

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 Crystal Enterprise 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 reporting command overview

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. 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 are not provided, the user is prompted to log on to the CMS.

The following table lists the available viewer commands:

Command	Description
ID on page 16	Specifies the ID of the current report.
APSTOKEN on page 17	Specifies the logon token for the current CMS session.
APSUSER, APSPASSWORD, APSAUTHTYPE on page 18	Specifies the CMS user name, password, and authentication type.
INIT on page 19	Specifies the viewer.
CONNECT on page 20	Re-establishes a connection to the Page Server.
PASSWORD (see *USER# and PASSWORD#* on page 21 and *USER and PASSWORD* on page 22 for details)	Specifies the passwords for logging on to SQL, ODBC, or password-protected databases that are used by the report.
USER (see *USER# and PASSWORD#* on page 21 and *USER and PASSWORD* on page 22 for details)	Specifies the user IDs for logging on to SQL or ODBC databases that are used by the report.

Command	Description
PROMPTEX (Use Case 1) on page 24	Specifies values for parameter fields in a report and subreport.
PROMPT# (Use Case 1) on page 29	Specifies values for parameter fields in a report. Recommended you use `PROMTPEX` instead.
PromptOnRefresh on page 32	Specifies whether the report should prompt for parameter field values when the report refreshed.
SF on page 32	Specifies a selection formula.
GF on page 33	Specifies a group selection formula.
CMD and EXPORT_FMT on page 34	Specifies that the report should be exported to the indicated format.
EXPORT_OPT on page 35	Specifies the page range to export.
sReportMode on page 36	Specifies the mode to use to display the report.
sReportPart on page 37	Specifies the report part to view.
sPartContext on page 37	Specifies the context of a report part
sZoom on page 38	Specifies the zoom of the document displayed.

Command	Description
rptsrc on page 39	Specifies the report source used by the report.

URL reporting commands

This section details the URL commands that are available and gives examples on how to use the commands.

A URL reporting URL is generally structured as follows:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?<com
mand1>&<command2>&...&<commandN>
```

- In a Java environment, `viewrpt.cwr` is a servlet mapping to `viewrpt`. This servlet is configured in a `web.xml` file installed with InfoView (`/CrystalReports/WEB-INF/web.xml`).
- In a .NET environment, `viewrpt.cwr` is redirected to `viewrpt.aspx`.

Note:
Variables are denoted with angle brackets. You must substitute the proper value for these variables. For example, you must use the name of your server in place of `<servername>` where it is contained in the code samples below, and you must use your port number in place of `<port>`.

ID

Syntax	Description	Mandatory?	Values
`id`	Specifies the ID of the current report.	Yes	The report ID associated with a report in the repository (SI_ID Property).

Example:

This example shows you how to create a link that generates a value for the report ID:

```
<a target='_blank' href='http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=" & "CurrentReportID & " & _    apstoken="
 & LogonToken & "'> View Report</A>
```

The resulting URL for this link is as follows:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&ap
stoken=myAPS:A1B2
```

The ID of this report is “152", a number that allows the Central Management Server (CMS) to uniquely identify each report.

Note:
To obtain the report ID, navigate to the report within the Central Management Console (CMC). The properties page for the report contains the report ID and the CUID. Use this value for the `id` parameter.

APSTOKEN

Syntax	Description	Mandatory?	Values
`apstoken`	Specifies the logon token for the current CMS session. Logon tokens are not encrypted.	No	Consists of a set of numbers and letters that are assigned internally, and then recognized by the CMS.

Logon tokens ensure that your company's licensing information is accurate. Each time a user logs on to the BusinessObjects Enterprise system, the count for the number of licenses that are currently in use increases by one. If a logon token is not passed as a parameter, each time the user logs on to the CMS (for example, the user may want to log on to a different BusinessObjects Enterprise client component), the count increments. As a

result, an administrator or auditor may overestimate the number of licenses in use.

Example:

This example shows you how to display a report when the user clicks the View Report link.

```
<a target='_blank' href='http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=" & "CurrentReportID & " & _    apstoken="
 & LogonToken & "'> View Report</a>
```

The resulting URL for this link is as follows:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&ap
stoken=myAPS:A1B2
```

APSUSER, APSPASSWORD, APSAUTHTYPE

Syntax	Description	Mandatory?	Values

`apsuser`
`apspassword`
`apsauthtype`
	Authentication credentials used to log on to the CMS.	No	CMS user name, password, and authentication type (`secEnterprise`, `secLDAP`, `secWinAD`).

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, you may want to use the APSTOKEN command to log on to the CMS.

Example:

Here is an example that illustrates how to use the commands within a URL. The following values are specified within the URL:

- "Jlee" is specified for APSUSER.
- "secret" is specified for APSPASSWORD.
- "secEnterprise" is specified for APSAUTHTYPE.

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&ap
suser=JLee&apspassword=secret&apsauthtype=secEnterprise
```

INIT

Syntax	Description	Mandatory?	Values
`init`	The INIT command specifies the viewer that is used to view the report. If the INIT command is not specified, by default the DHTML viewer is used.	No	• `actx` (ActiveX) • `java` (Java using browser JVM) • `dhtml` (DHTML) • `part` (DHTML parts viewer)

Note:

- The default viewer can be configured on the server hosting URL reporting.
- 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>/CrystalReports/viewrpt.cwr?id=152&ap
stoken=A1B2&init=java
```

CONNECT

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.

Syntax	Description	Mandatory?	Values
`init=<viewer>:connect`	Re-establishes a connection to the Page Server.	No	N/A

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.

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>/CrystalReports/viewrpt.cwr?id=152&apstoken=A1B2&init=java:connect
```

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*.

USER# and PASSWORD#

Syntax	Description	Mandatory?	Values

`user#`
`password#`
`user#@subreport name`
`password#@subre portname`
	The USER# and PASSWORD# command allows you to pass logon credentials for the database that is used by the report and its subreports.	No	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.

Example:

Main Report:

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

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&apstoken=A1B2&user0=msmith&password0=1234
```

Subreport:

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

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&apstoken=A1B2&user0@Crosstab=msmith&password0@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>/CrystalReports/viewrpt.cwr?id=152&ap
stoken=A1B2&user0=msmith&password0=1234&user1=bsmith&pass
word1=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.

USER and PASSWORD

Syntax	Description	Mandatory?	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>`
	You can use the USER and PASSWORD command to pass logon credentials, as well as the server name and database name, for both the main report and subreport.	No	User name, password, server name, and database name.

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.

Example:

Main Report:

This example shows how to pass the following values to the 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>/CrystalReports/viewrpt.cwr?user-
systemdsn.xtreme=vantech&password-systemdsn.xtreme=1234
```

Subreport:

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>/CrystalReports/viewrpt.cwr?user-
systemdsn.pubs@sr=vantech&password-systemdsn.pubs@sr=vantech
```

PROMPTEX (Use Case 1)

Syntax	Description	Mandatory?	Values

`promptex-<promptname>`
 	
`promptex-<promptname>@<subrpt>`
	The PROMPTEX command allows you to specify each parameter by name, as well as by the value for the parameter.	No	<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:

Main Report:

In this example "hello" is passed as a value for the parameter called "sample":

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&apstoken=A1B2&promptex-sample="hello"
```

Subreport:

In this example "hello" is passed as a value for the parameter called "sample" for the subreport called "mysubrpt":

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&apstoken=A1B2&promptex-sample@mysubrpt="hello"
```

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	Mandatory?	Values

`promptex-sample="<valueA>","<value B>","<value C>"`
 	
`promptex-sample=["<valueA>"-"<value B>"]`
	The PROMPTEX command allows you to specify multiple values to a parameter.	No	<promptname> and <subrpt> are the same as PROMPTEX (Use Case 1). <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:

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

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&apstoken=A1B2&promptex-fruits="Apples","Oranges","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>/CrystalReports/viewrpt.cwr?id=152&aptoken=A1B2&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>/CrystalReports/viewrpt.cwr?id=152&aptoken=A1B2&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>/CrystalReports/viewrpt.cwr?id=152&aptoken=A1B2&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	Mandatory?	Values

`promptex-<promptname>="Date(YYYY,MM,DD)"`
 	
`promptex-<promptname>=["Date(YYYY,MM,DD)"-"Date(YYYY,MM,DD)"]`
	The PROMPTEX command allows you to specify Date or DateTime parameter values, using the Single Value or Date Range methods	No	Date or datetime parameters passed. A specific date or date range can be passed. For single value Date or DateTime parameters, the promptex-<promptname> command requires double quotes.

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>/CrystalReports/viewrpt.cwr?id=152&aptoken=A1B2&promptex-birthdate="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>/CrystalReports/viewrpt.cwr?id=152&aptoken=A1B2&promptex-DateRangeParameter=["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	Mandatory?	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.	No	<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.

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

Example:

In this example, PROMPTEX# passes "CA" as a value for the first parameter in the report:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&apstoken=A1B2&promptex0="CA"
```

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:

- 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.

PROMPT# (Use Case 1)

Syntax	Description	Mandatory?	Values
`prompt#`	The PROMPT# command allows you to specify 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.	No	<value> is a string. This arbitrary (and potentially empty) string 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.

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

Example:

In this example, PROMPT# passes "CA" as a value to the first parameter:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&aptoken=A1B2&prompt0=CA
```

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
```

Note:
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 also use

```
promptex-paramter=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:

- 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.

PROMPT# (Use Case 2)

Syntax	Description	Mandatory?	Values
prompt#=Date(YYYY,MM,DD)	The PROMPT# command allows you to specify 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.	No	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.

Example:

To pass a Date value of February, 02, 2002 for the second parameter within a report, use the following URL command:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&apstoken=A1B2&prompt2=Date(2002,02,02)
```

Note:

- 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.

PromptOnRefresh

Syntax	Description	Mandatory?	Values
`promptOnRefresh`	The PromptOnRefresh command specifies whether the report should prompt for a parameter field values when re-freshed.	No	PromptOnRefresh will only accept 0 or 1 as a value. 0 is for false and 1 is for true

Example:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&apstoken=A1B2&promptOnRefresh=1
```

SF

Syntax	Description	Mandatory?	Values
`sf`	Indicates which specific part of the target report to open.	No	Valid Crystal Report 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, and the newly specified SF command will be based on those existing selection formulas.

For example, if the report already contains a selection formula that selects the records for film studios in the state of California, and then the SF command is used to append a formula that selects the records for "Universal," information on that particular studio will be displayed. Had the SF command specified a value such as "Sony," the requested report would contain no data, because that studio is not located in California.

Example:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&ap
stoken=A1B2&sf={studio.Studio}&=&'Universal'
```

Note:
The new selection formula is not saved with the original report file. It is only valid for the job that is currently requested.

GF

Syntax	Description	Mandatory?	Values
`gf`	The GF command specifies a group selection formula for the report. This command is similar to the selection formula SF command.	No	Valid Crystal Report group selection formula.

Example:

This example shows how to pass a group 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=152&init=java&apstoken=A1B2&gf=Sum({cus
tomer.Sales},{customer.Region})>10000
```

- Pages are shared between reports that have the same SF and GF commands applied and that do not require logon information.
- The GF command does not work with the DHTML viewer. You must specify the

  ```
  init
  ```

  command in your URL with the ActiveX or Java viewer.

CMD and EXPORT_FMT

Syntax	Description	Mandatory?	Values

`cmd=EXPORT`
 	
`EXPORT_FMT=<EXPORT_FMT representation>`
	The CMD and EXPORT_FMT commands specify that the report must be exported to the indicated format.	No	See table below.

Below is a listing of possible values for the <EXPORT_FMT representation>:

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

Export Format	Export_FMT Representation
Rich Text Format (RTF)	U2FRTF:0
Microsoft Word - Editable (RTF)	U2FRTF:1
Microsoft Word (97-2003)	U2FWORDW:0
XML	U2FXML:0

Example:

If users want to download the report to their browser in Rich Text Format (RTF), they use the following URL:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&aptoken=A1B2&cmd=EXPORT&EXPORT_FMT=U2FRTF:0
```

EXPORT_OPT

Syntax	Description	Mandatory?	Values
`EXPORT_OPT=[firstPage-lastPage]`	The EXPORT_OPT command specifies the range of pages in the report to export. The default value is "(-)" which specifies that the whole report is exported.	No	In range page numbers, where firstPage is smaller than lastPage.

Example:

The following example exports the first four pages of a report to rich text format:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?id=152&ap
stoken=A1B2&cmd=EXPORT&EXPORT_FMT=U2FRTF:0&EXPORT_OPT=[1-4]
```

sReportMode

Syntax	Description	Mandatory?	Values
`sReportMode`	Indicates the mode to use to display reports.	No	• `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=152&init=html&sReportMode=part
```

sReportPart

Syntax	Description	Mandatory?	Values
`sReportPart`	Indicates which specific part of the target report to open.	No	Name of the report object to display initially. Defaults to initial part in the report if not specified.

Example:

The following example specifies the report object to be opened:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=152&sReportPart=graph3
```

Note:
The `sReportPart` and `sPartContext` parameters are supported with the regular DHML viewer (`init=dhtml`). Instead of the first page of the report, the DHTML viewer displays the page and context containing the part with the part highlighted.

sPartContext

Syntax	Description	Mandatory?	Values
`sPartContext`	The initial context of a report part .	No	Data context of the report part. Defaults to initial context in the report part if not specified.

Example:

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

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=152&sPartContext=/USA/CA
```

Note:
The `sReportPart` and `sPartContext` parameters are supported with the regular DHML viewer (`init=dhtml`). Instead of the first page of the report, the DHTML viewer displays the page and context containing the part with the part highlighted.

sZoom

Syntax	Description	Mandatory?	Values
`sZoom`	The zoom level.	No	An integer value for the zoom. Defaults to 100 if not specified.

Example:

The following example specifies the zoom level of a report:

```
http://<servername>:<port>/CrystalRe
ports/viewrpt.cwr?id=152&sZoom=50
```

rptsrc

Syntax	Description	Mandatory?	Values
`rptsrc`	The report source used in the report.	No	Name of the report source.

Example:

The following example specifies the report source:

```
http://<servername>:<port>/CrystalReports/viewrpt.cwr?rptsrc=reportSource
```

For more information see *rptsrc command* on page 39.

rptsrc command

In Java and .NET the interface `IReportSource` and the member `Report Source` respectively can be used as a session variable in order to programmatically link to a report using URL reporting. The following give examples of how to use this command in Java and .NET.

URL reporting with the report source as a session variable (.NET)

This sample shows you how to perform URL reporting with the report source as a session variable.

For this scenario to work, you must run your application on the same machine as your application server.

Example: ReportSourceURL.aspx.vb VB.NET

```
Visual Basic
Imports CrystalDecisions.Enterprise
Imports CrystalDecisions.Enterprise.Desktop
Imports CrystalDecisions.ReportAppServer.ClientDoc

Public Class ReportSourceURL
    Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()

    End Sub

   Private Sub Page_Init(ByVal sender As System.Object, ByVal
 e As System.EventArgs) Handles MyBase.Init
       'CODEGEN: This method call is required by the Web Form
 Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

   Private Sub Page_Load(ByVal sender As System.Object, ByVal
 e As System.EventArgs) Handles MyBase.Load
        Dim myInfoStore As InfoStore
        Dim myEnterpriseSession As EnterpriseSession

        myInfoStore = CType(Session("InfoStore"), InfoStore)
        myEnterpriseSession = CType(Session("EnterpriseSes
sion"), EnterpriseSession)

        Dim query As String = "Select SI_ID, SI_NAME " _
            & "From CI_INFOOBJECTS Where SI_KIND = 'CrystalRe
port' " _
          & "AND SI_INSTANCE=0 AND SI_NAME='Alerting Report'"


        Dim myInfoObjects As InfoObjects = myInfoS
tore.Query(query)
        Dim myInfoObject As InfoObject = myInfoObjects(1)

       Dim tempService As EnterpriseService = myEnterpriseSes
sion.GetService("", "RASReportFactory")
        Dim myReportAppFactory As ReportAppFactory =
CType(tempService.Interface, ReportAppFactory)
```

```
        Dim myReportClientDocument As ReportClientDocument =
 myReportAppFactory.OpenDocument(myInfoObject.ID, 0)

        Session("reportSource") = myReportClientDocument.Re
portSource

        Dim viewURL As String = "viewrpt.aspx?rptsrc=report
Source&init=actx"

        Response.Redirect(viewURL)
    End Sub

End Class
```

Example: ReportSourceURL.aspx.cs C#

C#

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using CrystalDecisions.Enterprise;
using CrystalDecisions.Enterprise.Desktop;
using CrystalDecisions.ReportAppServer.ClientDoc;

namespace FeatureExamplesCSharp
{
 /// <summary>
 /// Summary description for ReportSourceURL.
 /// </summary>
 public class ReportSourceURL : System.Web.UI.Page
 {
  private void Page_Load(object sender, System.EventArgs e)
  {
   InfoStore infoStore;
   EnterpriseSession enterpriseSession;

   infoStore = (InfoStore)Session["InfoStore"];
   enterpriseSession = (EnterpriseSession)Session["Enterpris
eSession"];
```

```
    string query = "Select SI_ID, SI_NAME "
     + "From CI_INFOOBJECTS Where SI_KIND = 'CrystalReport' "
     + "AND SI_INSTANCE=0 AND SI_NAME='Alerting Report'";

    InfoObjects infoObjects = infoStore.Query(query);
    InfoObject infoObject = infoObjects[1];

    EnterpriseService tempService = enterpriseSession.GetSer
vice("", "RASReportFactory");
    ReportAppFactory reportAppFactory = (ReportAppFacto
ry)tempService.Interface;
   ReportClientDocument reportClientDocument = reportAppFacto
ry.OpenDocument(infoObject.ID, 0);

   Session["reportSource"] = reportClientDocument.ReportSource;


   string viewURL = "viewrpt.aspx?rptsrc=reportSource&init=ac
tx";

    Response.Redirect(viewURL);
  }

  #region Web Form Designer generated code
  override protected void OnInit(EventArgs e)
  {
   //
   // CODEGEN: This call is required by the ASP.NET Web Form
 Designer.
   //
   InitializeComponent();
   base.OnInit(e);
  }

  /// <summary>
  /// Required method for Designer support - do not modify
  /// the contents of this method with the code editor.
  /// </summary>
  private void InitializeComponent()
  {
   this.Load += new System.EventHandler(this.Page_Load);
  }
  #endregion
 }
}
```

URL reporting with the report source as a session variable (Java)

This sample shows you how to perform URL reporting with the report source as a session variable.

To configure your application server to run this sample

For this scenario to work, you must run your application on the same machine as your application server.

1. Use the web.xml file from the `CrystalReports` virtual directory.
2. Configure the viewer.
3. Modify the web.xml file:
   - Map the `path.dhtmlViewer` parameter to your `crystalreportviewers` directory.
   - Map the `path.javaAppletViewer` parameter to your `crystalreportviewers` directory.
   - Map the `viewrpt.javaPluginPath` parameter to `crystalreportviewers/JavaPlugin/Win32/jre-1_5_0_11-i586-p.exe`.
   - Map the path.dhtmlViewer parameter to your `CrystalReports` virtual directory.

Example: ReportSourceURLReporting.jsp

```
<!-- begin JSP segment -->

<%@ page contentType="text/html; charset=UTF-8" %>

<%@ page import="com.crystaldecisions.sdk.exception.SDKExcep
tion" %>
<%@ page import="com.crystaldecisions.sdk.occa.infostore.*"
%>
<%@ page import="com.crystaldecisions.sdk.occa.pluginmgr.*"
%>
<%@ page import="com.crystaldecisions.sdk.plugin.*" %>
<%@ page import="com.crystaldecisions.sdk.plugin.desktop.re
port.*" %>
```

```
<%@ page import="com.crystaldecisions.sdk.occa.manage
dreports.IReportSourceFactory" %>
<%@ page import="com.crystaldecisions.sdk.occa.report.report
source.IReportSource" %>
<%@ page import="java.util.*"%>

<%
/*
 This sample shows you how to perform URL reporting with the
 report source as a session variable.
 For this scenario to work, you must run your application on
 the same machine as your application server.

 To configure your application server to run this sample:

 1. Use the web.xml file from the CrystalReports virtual di
rectory.

 2. Configure the viewer.

 3. Modify the web.xml file:
    - Map the path.dhtmlViewer parameter to your crystalre
portviewers directory.
    - Map the path.javaAppletViewer parameter to your crystal
reportviewers directory.
    - Map the viewrpt.javaPluginPath parameter to crystalre
portviewers/JavaPlugin/Win32/jre-1_5_0_11-i586-p.exe.
    - Map the path.dhtmlViewer parameter to your CrystalReports
 virtual directory.

 */
%>

<!-- begin HTML segment -->

<html>
<head>
</head>
<body>
<%
/* Retrieve the IInfoStore object of the current session.
 */
IInfoStore iStore = (IInfoStore) session.getAttribute("InfoS
tore");

/* Retrieve the IReportSourceFactory object. */
IReportSourceFactory reportSourceFactory = (IReportSourceFac
tory) session.getAttribute("ReportSourceFactory");

/* Retrieve the logon token. */
String token = (String) session.getAttribute("token");
```

```
/* Query for an alerting report.
 * Choose the first report returned by the query.
 */
try
{
   IInfoObjects reports = iStore.query("SELECT SI_ID, SI_NAME
 FROM "
                                     + "CI_INFOOBJECTS WHERE
SI_KIND='CrystalReport' AND "
                                     + "SI_NAME = 'Alerting Re
port' AND SI_INSTANCE=0");

 IReport report = (IReport) reports.get(0);

 /* Create the report source. */
 IReportSource reportSource = reportSourceFactory.openReport
Source(report, Locale.ENGLISH);

 /* Store the report source in session for URL reporting. */
    session.setAttribute("reportSource", reportSource);

    /* Specify the session variable name 'reportSource' as a
 parameter, and ActiveX viewer. */
   String viewURL = "viewrpt.cwr?rptsrc=reportSource&init=ac
tx";

    /* Redirect the user to the URL. */
    response.sendRedirect(viewURL);
}
catch(SDKException e)
{
 throw new Error("An error has occurred: "
                 + e.getMessage());
}
%>
</body>
</html>

<!-- end HTML segment -->
```

To configure your application server to run this sample

For this scenario to work, you must run your application on the same machine as your application server.

1. Use the web.xml file from the `CrystalReports` virtual directory.
2. Configure the viewer.
3. Modify the web.xml file:
   - Map the `path.dhtmlViewer` parameter to your `crystalreportviewers` directory.
   - Map the `path.javaAppletViewer` parameter to your `crystalreportviewers` directory.
   - Map the `viewrpt.javaPluginPath` parameter to `crystalreportviewers/JavaPlugin/Win32/jre-1_5_0_11-i586-p.exe`.
   - Map the path.dhtmlViewer parameter to your `CrystalReports` virtual directory.

Example: ReportSourceURLReporting.jsp

```
<!-- begin JSP segment -->

<%@ page contentType="text/html; charset=UTF-8" %>

<%@ page import="com.crystaldecisions.sdk.exception.SDKExcep
tion" %>
<%@ page import="com.crystaldecisions.sdk.occa.infostore.*"
%>
<%@ page import="com.crystaldecisions.sdk.occa.pluginmgr.*"
%>
<%@ page import="com.crystaldecisions.sdk.plugin.*" %>
<%@ page import="com.crystaldecisions.sdk.plugin.desktop.re
port.*" %>
<%@ page import="com.crystaldecisions.sdk.occa.manage
dreports.IReportSourceFactory" %>
<%@ page import="com.crystaldecisions.sdk.occa.report.report
source.IReportSource" %>
<%@ page import="java.util.*"%>

<%
/*
 This sample shows you how to perform URL reporting with the
 report source as a session variable.
 For this scenario to work, you must run your application on
 the same machine as your application server.

 To configure your application server to run this sample:

 1. Use the web.xml file from the CrystalReports virtual di
rectory.
```

```
 2. Configure the viewer.

 3. Modify the web.xml file:
    - Map the path.dhtmlViewer parameter to your crystalre
portviewers directory.
    - Map the path.javaAppletViewer parameter to your crystal
reportviewers directory.
    - Map the viewrpt.javaPluginPath parameter to crystalre
portviewers/JavaPlugin/Win32/jre-1_5_0_11-i586-p.exe.
    - Map the path.dhtmlViewer parameter to your CrystalReports
 virtual directory.

 */
%>

<!-- begin HTML segment -->

<html>
<head>
</head>
<body>
<%
/* Retrieve the IInfoStore object of the current session.
 */
IInfoStore iStore = (IInfoStore) session.getAttribute("InfoS
tore");

/* Retrieve the IReportSourceFactory object. */
IReportSourceFactory reportSourceFactory = (IReportSourceFac
tory) session.getAttribute("ReportSourceFactory");

/* Retrieve the logon token. */
String token = (String) session.getAttribute("token");

/* Query for an alerting report.
 * Choose the first report returned by the query.
 */
try
{
    IInfoObjects reports = iStore.query("SELECT SI_ID, SI_NAME
 FROM "
                                        + "CI_INFOOBJECTS WHERE
SI_KIND='CrystalReport' AND "
                                        + "SI_NAME = 'Alerting Re
port' AND SI_INSTANCE=0");

 IReport report = (IReport) reports.get(0);

 /* Create the report source. */
 IReportSource reportSource = reportSourceFactory.openReport
Source(report, Locale.ENGLISH);
```

```
 /* Store the report source in session for URL reporting. */
    session.setAttribute("reportSource", reportSource);

    /* Specify the session variable name 'reportSource' as a
 parameter, and ActiveX viewer. */
    String viewURL = "viewrpt.cwr?rptsrc=reportSource&init=ac
tx";

    /* Redirect the user to the URL. */
    response.sendRedirect(viewURL);
}
catch(SDKException e)
{
 throw new Error("An error has occurred: "
                + e.getMessage());
}
%>
</body>
</html>

<!-- end HTML segment -->
```

URL Reporting using openDocument

chapter 4

URL reporting using openDocument provides URL access to multiple document types by passing a URL string to a BusinessObjects Enterprise server. openDocument provides commands to control how reports are generated and displayed.

You can use openDocument in BusinessObjects Enterprise to create cross-system links to and from the following document types:

- .wid: Web Intelligence documents
- .rep: Desktop Intelligence documents
- .rpt: Crystal reports
- .car: OLAP Intelligence reports

Structuring an openDocument URL

The next sections explain how to use the openDocument function, and how to construct the URL.

An openDocument URL is generally structured as follows:

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

The exact syntax of the `<platformSpecific>` parameter depends on your implementation:

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

The URL is constructed using the parameters listed in *openDocument parameter overview* on page 53

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 URLEncoded 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.

For details on prompt parameters of the link, refer to *openDocument parameter overview* on page 53 .

Capitalization

All of the openDocument parameters are case sensitive.

Link length limit

The encoded URL cannot exceed 2083 total characters.

Parameter values in links to sub-reports

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

Using the lsS parameter with 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 a 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>?sType=car&sIDType=InfoObject&iDocID=440&lsS23CAA3C1-8DBB-
4CF3-BA%2CB8%2CD7%2CF0%2C68%2CEF%2C9C%2C6F=2
```

Example: Opening 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>?sType=car&sIDType=InfoObject&lsS8401682C-9B1D-4850-
8B%2C5E%2CD9%2C1F%2C20%2CF8%2C1%2C62=CATALOG%3DFood
Mart%202000,CUBE%3Dwarehouse&iDocID=616
```

Using the lsM parameter with OLAP Intelligence reports

If the target document is an OLAP Intelligence report (.car) you can use the lsM 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.

As was the case for the lsS parameter, lsM parameters are also passed in as a URL-encoded string using the unique name of the parameter set up in the OLAP Intelligence report.

Example: Opening a report

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

This is a memberset parameter opening up a report with Customers > Country > Mexico and Customers > Country > Canada in the view.

---

openDocument parameter overview

This section provides a brief overview of openDocument and includes a list of available parameters. Details about the available parameters, their specific uses, and relevant examples are also provided.

Note:
The document containing the openDocument link is called the parent document, and it resides on the parent system. The document to which the link points is called the target document, and it resides on the target system.

Table 4-1: Platform Parameters

Parameter	Description
iDocID on page 57	Document identifier.
sDocName on page 58	Document name.
sIDType on page 58	Crystal object type.
sKind on page 59	The file type of target Desktop Intelligence document.
sPath on page 60	The name of the folder and subfolder containing the target document.
sType on page 60	The file type of target document or report.
token on page 61	A valid logon token for the current CMS session.

Table 4-2: Input Parameters

Parameter	Description
lsC on page 62	Specifies a contextual prompt if there is an ambiguity during SQL generation (Business Objects and Web Intelligence documents only).

Parameter	Description
lsM[NAME] on page 63	Specifies multiple values for a prompt, `[NAME]` is the text of the prompt
lsR[NAME] on page 64	Specifies a range of values for a prompt, `[NAME]` is the text of the prompt.
lsS[NAME] on page 65	Specifies a value for a single prompt. `[NAME]` is the text of the prompt.
sInstance on page 66	Indicates which specific instance of the target report to open.
sPartContext on page 66	In Crystal Reports, a report part is associated to a data context.
sRefresh on page 67	Indicates whether a refresh should be forced when the target document or report is opened.
sReportMode on page 68	For Crystal targets only, indicates whether the link should open the full target report or just the report part specified in `sReportPart`.
sReportName on page 68	Indicates which report to open if target document is multi-report.
sReportPart on page 69	Indicates which specific part of the target report to open.

Table 4-3: Output Parameters

Parameter	Description
NAII on page 70	Forces the display of the prompt selection page.
sOutputFormat on page 71	Indicates the format in which the target document is opened.
sViewer on page 72	Indicates the selected report viewer (CR & CA only).
sWindow on page 73	Indicates whether the target report will open in the current browser window or whether a new window will be launched.

openDocument platform parameters

This section details the openDocument platform parameters that are available and gives examples on how to use the commands.

An openDocument URL is generally structured as follows:

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

The exact syntax of the `<platformSpecific>` parameter depends on your implementation:

- For Java implementations, use `openDocument.jsp` in place of the `<platformSpecific>` parameter.
- For .NET implementations, 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 server in place of `<servername>` where it is contained in the code samples below, and you must use your port number in place of `<port>`.

iDocID

Syntax	Description	Mandatory?	Values
`iDocID`	Document identifier.	Yes*	Document identifier (InfoObjectID).

Note:
*One of `sDocName` or `iDocID` is mandatory.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sType=wid&sDocName=SalesReport&iDocID=2010
```

Note:
To obtain the document ID, navigate to the document within the Central Management Console (CMC). The properties page for the document contains the document ID and the CUID. Use this value for the `iDocID` parameter.

sDocName

Syntax	Description	Mandatory?	Values
`sDocName`	Document name without extension If multiple documents have the same name, specify the correct document with `iDocID`.	Yes*	Document name.

Note:
*One of `sDocName` or `iDocID` is mandatory.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sPath=[Sales+Reports]&sDocName=Sales+in+200
```

---

sIDType

Syntax	Description	Mandatory?	Values
`sIDType`	Central Management Server (CMS) object identifier type.	Yes*	• `CUID` • `GUID` • `RUID` • `ParentID` • `InfoObjectID` (default)

Note:
*Only mandatory if the target is a Crystal report or OLAP Intelligence report (`sType=rpt` or `sType=car`) in an Object Package. Otherwise, use `sPath` and `sDocName`.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?iDocID=2010&sIDType=CUID
```

sKind

Syntax	Description	Mandatory?	Values
`sKind`	The file type of target Desktop Intelligence document.	Yes*	• `FullClient`

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

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sKind=FullClient
```

sPath

Syntax	Description	Mandatory?	Values
`sPath`	The name of the folder and subfolder containing the target document.	Yes*	Folder and/or subfolder: `[folder],[subfolder]`

Note:
*Only mandatory if a value is specified for `sDocName` and is not unique.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sPath=[Sales+Reports]&sDocName=Sales+in+2005
```

---

sType

Syntax	Description	Mandatory?	Values
`sType`	The file type of target document or report.	Yes	• `wid` • `rpt` • `car`

Note:
This parameter is ignored for agnostic documents.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sType=wid
```

---

token

Syntax	Description	Mandatory?	Values
`token`	A valid logon token for the current CMS session.	No	The logon token for the current CMS session.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sType=wid&sDocName=Sales+in+2003&token=<logonToken>
```

openDocument input parameters

This section details the openDocument input parameters that are available and gives examples on how to use the commands.

An openDocument URL is generally structured as follows:

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

The exact syntax of the `<platformSpecific>` parameter depends on your implementation:

- For Java implementations, use `openDocument.jsp` in place of the `<platformSpecific>` parameter.
- For .NET implementations, 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 server in place of `<servername>` where it is contained in the code samples below, and you must use your port number in place of `<port>`.

lsC

Syntax	Description	Mandatory?	Values
`lsC`	Specifies a contextual prompt if there is an ambiguity during SQL generation (Business Objects and Web Intelligence documents only). **Note:** Not supported by OLAP Intelligence	No	A prompt value that resolves the ambiguity in the SQL generation.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sType=wid&sDocName=SalesReport&iDocID=2010&lsC=Sales
```

lsM[NAME]

Syntax	Description	Mandatory?	Values
`lsM[NAME]`	Specifies multiple values for a prompt. `[NAME]` is the text of the prompt.	No	• Multiple prompt values, separated by a comma. If the target is a Crystal report, each value must be enclosed in square brackets. If the target is a OLAP Intelligence report, use the MDX WITH clause (refer to *Using the lsS parameter with OLAP Intelligence reports* on page 52 and *Using the lsM parameter with OLAP Intelligence reports* on page 53). • `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/<platformSpe
cific>?sType=rpt&sDocName=SalesReport&lsMSe
lect+Cities=[Paris],[London]
```

lsR[NAME]

Syntax	Description	Mandatory?	Values
`lsR[NAME]`	Specifies a range of values for a prompt. `[NAME]` is the text of the prompt. **Note:** Not supported by OLAP Intelligence	No	• A range of values for the prompt, separated by a double period (..). 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). • `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/<platformSpe
cific>?sType=rpt&sDocName=SalesReport&lsRTime+Peri
od:=[2000..2004)
```

lsS[NAME]

Syntax	Description	Mandatory?	Values
`lsS[NAME]`	Specifies a value for a single prompt. `[NAME]` is the text of the prompt.	No	• A single prompt value (refer to *Using the lsS parameter with OLAP Intelligence reports* on page 52 and *Using the lsM parameter with OLAP Intelligence reports* on page 53). • `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/<platformSpecific>?sType=wid&sDocName=SalesReport&iDocID=2010&lsSSelect+a+City=Paris
```

sInstance

Syntax	Description	Mandatory?	Values
`sInstance`	Indicates which specific instance of the target report to open.	No	• `User` (Link to latest instance owned by current user) • `Last` (Link to latest instance for report) • `Param` (Link to latest instance of report with matching parameter values)

Note:
Use this parameter in combination with `sDocName`.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sPath=[Sales+Reports]&sDocName=Sales+in+2003&sReportPart=Part1&sInstance=User
```

sPartContext

Syntax	Description	Mandatory?	Values
`sPartContext`	In Crystal Reports, a report part is associated to a data context.	Yes*	Data context of the report part.

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

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sPath=[Sales+Reports]&sDocName=Sales+in+2005&sReportPart=Part1&sPartContext=0-4-0
```

Note:
The `sReportPart` and `sPartContext` parameters are supported with the regular DHML viewer (`sViewer=html`). Instead of the first page of the report, the DHTML viewer displays the page and context containing the part with the part highlighted.

sRefresh

Syntax	Description	Mandatory?	Values
`sRefresh`	Indicates whether a refresh should be forced when the target document or report is opened.	No	• `Y` (forces the document's refresh) • `N` (note that the refresh on open feature overrides this value)

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sType=wid&sDocName=SalesReport&iDocID=2010&sRefresh=Y
```

---

sReportMode

Syntax	Description	Mandatory?	Values
`sReportMode`	For Crystal targets only, indicates whether the link should open the full target report or just the report part specified in `sReportPart`.	No	• `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/<platformSpecific>?sPath=[Sales+Reports]&sDocName=Sales+in+2003&sReportPart=Part1&sReportMode=Part
```

---

sReportName

Syntax	Description	Mandatory?	Values
`sReportName`	Indicates which report to open if target document is multi-report.	No	Report name for Web Intelligence documents, sub-report for Crystal Reports, pages for OLAP Intelligence reports.

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

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpe
cific>?sType=wid&sDocName=Sales+in+2003&sReportName=First+Re
port+Tab
```

---

sReportPart

Syntax	Description	Mandatory?	Values
`sReportPart`	Indicates which specific part of the target report to open.	No	Name of the report part.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpe
cific>?sPath=[Sales+Reports]&sDocName=Sales+in+2003&sReport
Part=Part1
```

Note:
The `sReportPart` and `sPartContext` parameters are supported with the regular DHML viewer (`sViewer=html`). Instead of the first page of the report, the DHTML viewer displays the page and context containing the part with the part highlighted.

openDocument output parameters

This section details the openDocument output parameters that are available and gives examples on how to use the commands.

An openDocument URL is generally structured as follows:

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

The exact syntax of the <platformSpecific> parameter depends on your implementation:

- For Java implementations, use openDocument.jsp in place of the <platformSpecific> parameter.
- For .NET implementations, 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 server in place of <servername> where it is contained in the code samples below, and you must use your port number in place of <port>.

NAII

Syntax	Description	Mandatory?	Values
NAII	Forces the display of the prompt selection page. **Note:** Only supported by Web Intelligence documents.	No	• Y (all prompts whose values are passed with lsS, lsM or lsR are pre-selected)

Note:

- If the document prompt values are purged, then NAII=Y raises the prompt for any values not specified in the URL.
- If the document prompt values are not purged, then NAII=Y raises the prompt for any values not specified in the URL and pre-selects the default values.
- If all prompt values are specified in the URL, the prompt window does not appear even if NAII=Y is specified.

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpe
cific>?sType=wid&sDocName=SalesReport&iDocID=2010&NAII=YsRe
fresh=Y
```

---

sOutputFormat

Syntax	Description	Mandatory?	Values
`sOutputFormat`	Indicates the format in which the target document is opened.	No	• `H` (HTML) • `P` (PDF) • `E` (Excel) • `W` (Word)

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

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpe
cific>?sPath=[Sales+Reports]&sDocName=Sales+in+2003&sOutput
Format=E
```

---

sViewer

Syntax	Description	Mandatory?	Values
`sViewer`	Indicates the viewer that is used to view the document.	No	• `html` • `part` (Crystal reports only) • `actx` (Crystal reports only) • `java` (Crystal reports only)

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sPath=[Sales+Reports]&sDocName=Sales+in+2003&sViewer=html
```

Example:

Note:
In order to use parameters in the URL with the ActiveX viewer, `:connect` must be appended to the URL, followed by the parameters.

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sPath=[Sales+Reports]&sDocName=Sales+in+2003&sViewer=actx:connect&IsMCountry=[Thailand],[Norway]
```

---

sWindow

Syntax	Description	Mandatory?	Values
`sWindow`	Indicates whether the target report will open in the current browser window or whether a new window will be launched.	No	• `Same` (current browser window) • `New` (new browser window is launched)

Example:

```
http://<servername>:<port>/OpenDocument/opendoc/<platformSpecific>?sType=wid&sDocName=SalesReport&iDocID=2010&sWindow=New
```

Contextual report linking

The openDocument feature allows you to create contextual links between Crystal reports, OLAP Intelligence reports and Web Intelligence documents. To do this, you construct a URL using the openDocument syntax and then insert the URL into a Crystal report, OLAP Intelligence report or Web Intelligence document.

Contextual report linking allows report designers to specify associations for documents residing in either a Crystal Reports environment (unmanaged) or a BusinessObjects Enterprise environment (managed). Once these associations are created, users follow the resulting navigational paths embedded in the linked documents.

This feature enables you to invoke Business Objects and Web Intelligence documents from Crystal Reports and vice versa. This feature relies on functionality that allows the user to do the following:

- Link Web Intelligence or Business Objects documents within the document domain.
- Link report objects in Crystal Reports.

Click the appropriate link to jump to that section:

To insert a link into a Crystal report

You can use openDocument to create hyperlinks in Crystal Reports. To create a link to another report or document, use the Hyperlink Tab of the field Format Editor.

1. Open the source report in Crystal Reports.
2. Right-click the field in which you want to insert the openDocument link and select **Format** from the shortcut menu.
3. In the Format Editor, select the **Hyperlink** tab.
4. Select **A website on the Internet**.
5. In the “Hyperlink information□? area, leave the **Website Address** field empty and click the **Format Formula Editor** button.
6. Enter the openDocument link in the following format:

```
"http://[openDocument parameters]"+{Article_lookup.Fami
ly_name}
```

   Where [openDocument parameters] are described in *openDocument parameter overview* on page 53, and the {Article_lookup.Family_name} enables the report to pass context-dependent data.

   Note:
   Test your link in a browser window before inserting it into a report or document.

7. Click **Save and Close** to leave the Formula Workshop.
8. Click **OK** in the Formula Editor to save the link.

To create a link to another report or document from an OLAP Intelligence report

You can use openDocument to create hyperlinks in OLAP Intelligence reports.

1. Open the source report in the OLAP Intelligence designer.
2. On the **Tools** menu, select **Action Manager** .
3. Click **New** to create a new action.
4. Enter an action name.
5. Select the area to which the action (the link) will apply.
6. Enter the openDocument link using the parameters and syntax described in this document.

   Tip:
   Test your link in a browser window before inserting it into a report or document.

7. Click **OK** to save the link.
8. Close the Action Manager dialog box.
9. Create an Analysis Button on the source report.
10. Right-click the Analysis Button.
11. In the drop-down menu, select **Properties** and then **Edit** .
12. Select **Launch an action** .
13. Select the action that corresponds to the openDocument link created in steps 3 through 6.
14. Click **OK** .

Creating links in Web Intelligence documents

You can define objects in a universe that allow Web Intelligence and BusinessObjects users to create reports whose returned values include links to other reports and documents.

When these reports are exported to the repository, users can click returned values displayed as hyperlinks to open another related document stored in the document domain of the repository.You create these links using the openDocument function in the definition of an object in Designer.

More information

For full information on creating links in Web Intelligence reports, see the *Building Reports Using the WebIntelligence Java Report Panel* guide.

You enable report linking in a universe by creating an object (the link object) whose returned values are the same as the values used as input to a prompt in an existing report (the target report).

The openDocument function allows the values for the link object to be returned as hyperlinks. When the user clicks the hyperlink, its value is used as the prompt input for the target report.

You can create documents using the link object as you would with any other object. Users can then click the hyperlinks to access more detailed documents related to the link object.

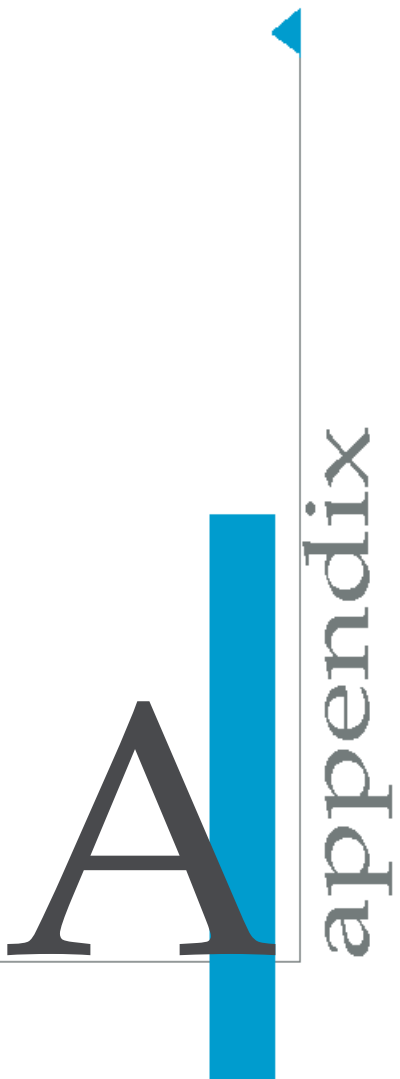
To create a link object, use the openDocument function in the object's Select statement. The Select statement for a link object follows this order:

```
'<a href="http://<servername>:<port>/OpenDocument/<platformSpe
cific>?sDocName=<document name>&sType=<document type>&iDo
cID=<document id>&lsS<prompt message>='+object SELECT+'">'+ob
ject SELECT+'</a>'
```

The concatenation operator (+) applies for Microsoft Access databases. Use the operator appropriate to your target RDBMS.

For more details on the Select statement, creating link objects, and using link objects in InfoView, refer to the *Designer's Guide*.

More Information



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-instguides* • 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 customized embedding technology.

Index

Index