ORACLE CRM On Demand

Oracle CRM On Demand Report Services API Developer's Guide

Release 34

February 2017

Contents

Index

1 What's New in This Release

What's New in Oracle CRM On Demand Report Services API Developer's Guide, Release 34

Table 1 lists changes made in this version of the documentation to support Release 34 of the software.

Table 1. What's New in Oracle CRM On Demand Report Services API Developer's Guide, Release 34

Topic	Description
Calling the ReportExecute Method on page 14	Modified topic. Added a description of the optional Action parameter of the ReportExecute method. For the Action parameter, there is one valid action: Filter.
Using Filters in the ReportExecute Method on page 16	New topic. It describes how to use the Filter action to filter reports.
Guidelines for Using the ReportExecute Method Efficiently on page 17	Modified topic. Added an example of using the Filter action.
Troubleshooting the ReportExecute Method on page 19	Modified topic. Added troubleshooting information for the Filter action.

2 Overview of the Oracle CRM On Demand Report Services API

This chapter provides an overview of the Oracle CRM On Demand Report Services application programming interface (API) available with Oracle CRM On Demand. It contains the following topics:

About the Oracle CRM On Demand Report Services API

The Oracle CRM On Demand Report Services API is a Web services based framework that provides a mechanism for administrators and users to list the reports in the company shared folders or to execute any specific report and download the output in specified formats to their local machines. The following methods are available for users:

- ReportExecute method. For more information, see Chapter 3, “Using the ReportExecute Method.”
- ReportList method. For more information, see Chapter 4, “Using the ReportList Method.”

The Oracle CRM On Demand Report Services API supports only company shared reports. It does not support prebuilt reports or personal reports. All reports executed by the Oracle CRM On Demand Report Services API are subject to the same limitations (time limits, row limits, data visibility, and so on) that apply to reports that run within Oracle CRM On Demand.

About Downloading Reports with the ReportExecute Method

The ReportExecute method provides a means to execute and download reports without using the Oracle CRM On Demand user interface. You can then share those reports with others.

You can download reports in the following formats:

- Comma separated values (CSV)
- Excel spreadsheet (XLS)
- MIME HTML (MHTML)
- Portable document format (PDF)
- Text (Txt)

Users can use the Oracle CRM On Demand Report Services API to obtain specific reports on a regular basis. Administrators can use software tools to automate report generation and send the reports on a scheduled basis. Also, administrators can provide faster response times for popular analytic reports by generating them every morning, thus calculating some of the values used in future runs of the report.

To use the ReportExecute method you must create reports in Oracle CRM On Demand and save those reports in a shared folder. Also, you must acquire a download tool, such as cURL or Wget or use a scripting language such as JavaScript or Perl. The download tool's purpose is to call the Oracle CRM On Demand Report Services API to generate a report and then to retrieve the report to your computer. cURL and Wget are examples of such download tools but you can use any tool of your choice that can post a request to the API and get a response back.

About Listing Reports with the ReportList Method

The ReportList method provides a means to list the reports in the company shared folders without using the Oracle CRM On Demand user interface. You can use the ReportList method to generate a summary list of reports in the company shared folder. The ReportList method accesses each folder and creates a zipped UTF-8 encoded CSV file with the summary list of reports.

To use the ReportList method you must create reports in Oracle CRM On Demand and save those reports in a shared folder. Also, you must acquire a download tool, such as cURL or Wget or use a scripting language such as JavaScript or Perl. The download tool's purpose is to call the ReportList method to generate a summary report list and then to retrieve it to your computer. cURL and Wget are examples of such download tools but you can use any tool of your choice that can post a request to the API and get a response back.

About Data Visibility Rules

When you use the Oracle CRM On Demand Report Services API, the user account you use to sign on to Oracle CRM On Demand through the API determines what visibility rules are applied. The Oracle CRM On Demand Report Services API retrieves the same data that users get when the report is run from within Oracle CRM On Demand. All Manager, Team, Book and Role-Based Record Type visibilities are maintained.

If you sign on with an account that has full visibility, then the API delivers a report that shows all applicable rows of data. It might not be appropriate to share this report with users who do not normally have access to such data.

The user account must have the Access Analytics Reports privilege and also have access permissions to the shared folder that contains the reports.

Companies that want to automate executing and retrieving reports by using their infrastructure must consider their data security and privacy policies.

About the Download Tool

If you do not already have a download tool, then download one from the Internet. Most are available for free. You can use any tool of your choice that can post a request to the API and get a response back.

For example, the Wget tool is a command line download tool that you can use. All commands are issued from a command line. You can also use a scripting language to do the same function.

Sequence of Operation

The following steps describe the basic sequence of how to use the Oracle CRM On Demand Report Services API. This applies to all methods included with the API:

1. Create reports in Oracle CRM On Demand and save them in a shared folder.
2. Using the download tool or scripting language, sign on to Oracle CRM On Demand with a command that creates a session cookie file.

   This step captures the Web session ID into a file for use in the next step. For more information, see “Creating a Session Cookie File” on page 9.
3. Using the download tool or scripting language, call the selected method:
   - “Using the ReportExecute Method” on page 13
   - “Using the ReportList Method” on page 23
4. Sign off from Oracle CRM On Demand to close the session.

   For more information, see “Signing Off from Oracle CRM On Demand” on page 10.

Creating a Session Cookie File

Before you can call a method, you must sign on to the Oracle CRM On Demand and establish a Web session. The following example shows the format of the command for the Wget tool:

```
wget  --content-on-error  --keep-session-cookies --save-cookies "cookiefile1.txt" --post-data "j_username=repuser1&j_password=pass1&langCode=ENU" https://secure-<servername>/OnDemand/authenticate
```

Substitute your specific information for the following parameters as shown in Table 2.

Table 2. Wget Command Parameters for Creating the Session Cookie File

Parameter	Value in Example	Description
`--save-cookies`	`"cookiefile1.txt"`	The name of the file into which Wget stores the Web session ID. The quotation marks around the file name are mandatory in Wget.
j_username	`repuser1`	The user name for the account you are using to sign on to Oracle CRM On Demand. The account you use controls the data visibility used to create the report. The user name must be URL Encoded, which means that special characters, such as spaces must be rendered in their hexadecimal equivalents. For example, the space character is ASCII 32, which in hexadecimal is %20.
j_password	`pass1`	The password associated with the user name. This password must also be URL Encoded. The password is transmitted unencrypted. To prevent compromising your personal data, do not use your own account. Instead, create a special account solely for the use of accessing reports.
langcode	`ENU`	The three letter code for your language.
servername	`<servername>`	The name or address of the server hosting Oracle CRM On Demand.

Signing Off from Oracle CRM On Demand

When you have retrieved all the reports you want, close the session with Oracle CRM On Demand by signing off. The following is an example of signing off using Wget:

```
wget  --load-cookie "cookiefile1.txt" https://secure-<servername>/OnDemand/
logoff.jsp
```

Displaying the Oracle CRM On Demand Report Services API Allotments

To ensure that all users have equitable use of report services, the Oracle CRM On Demand Report Services API is subject to a set of service allotment limitations. You can display the allotments in the Service Allotment List.

To display service allotments

- Navigate to Admin, Company Administration, and then Service Allotment Administration.

To display the current usage of your allotment

- Navigate to Admin, Company Administration, and then Service Allotment Administration.

To display the usage history of your allotment

- Navigate to Admin, Company Administration, and then to Service Allotment Usage History.

Table 3 describes the report services allotments.

Table 3. Report Services Allotments

Service Allotment Resources	Service Allotments	Description
Report Services Bandwidth Allotment	100 MB/day	Determines the total size in megabytes (MB) of the Report Service results that can be used by a company within a 24-hour period.
Report Services Concurrent Request Allotment	5	Determines the maximum number of concurrent Report Service requests that a company can process. Abandoned requests typically take an hour to clear, depending on the value of the SESSION_TIMEOUT control.
Report Services Operations Allotment	20/day	Determines the number of Report Service operations that a company can perform within a 24-hour period.

3 Using the ReportExecute Method

This chapter describes how to access the ReportExecute method. It includes the following topics:

Overview of the ReportExecute Method

The ReportExecute method accesses existing reports within Oracle CRM On Demand and creates portable files from them. A *portable file* is a file, such as a PDF file that can be shared and displayed outside of Oracle CRM On Demand.

To access existing reports, you need a scripting language, such as JavaScript or Perl, or a command line download tool such as Wget or cURL. The examples in this guide use the Wget tool, but the concepts are the same for all scripting languages and download tools.

The download tool calls the ReportExecute method and specifies the report (path and name), optional report filters, and the delivery format for the report. The Oracle CRM On Demand Report Services API can deliver reports in any of the following formats: PDF, XLS, TXT, CSV, and MHTML. The download tool then transfers the file generated by the ReportExecute method and stores it in the same location from where the report execution service call is made or in a location specified with the output file.

The ReportExecute method uses the data visibility controls in Oracle CRM On Demand. The user account you use to sign on to Oracle CRM On Demand through the ReportExecute method by way of the download tool, determines the data you have visibility into. If the reports you are downloading are to be shared with other users, then you must consider what data they are permitted to see and make sure that you do not access the ReportExecute method with an account that has access to restricted records.

Sequence of Operations for Using the ReportExecute Method

The ReportExecute method is not a stand-alone command. First, you must make sure you have a download tool as described in "About the Download Tool" on page 9. Then use the following sequence:

1. Sign on to Oracle CRM On Demand and create a cookie file (see "Creating a Session Cookie File" on page 9).
2. Call the ReportExecute method (see "Calling the ReportExecute Method" on page 14).
3. Close the Oracle CRM On Demand session (see "Signing Off from Oracle CRM On Demand" on page 10).

Calling the ReportExecute Method

When you have the Web session ID stored in a cookie file, you can then use the download tool to make the call to the ReportExecute method and get the report in the format you specify. When using Wget, the format for the call is as follows:

```
wget --content-on-error --load-cookie "cookiefile1.txt" --output-document
"pathname.pdf" "https://secure-<servername>/OnDemand/user/
ReportService?Method=ReportExecute&Path=<ReportPathandName>&Format=pdf
&Action=Filter&P0=1&P1=<op>&P2=<FilterColumn1Name>&P3=<FilterColumn1Value>&Refresh
=Y" --max-redirect=100
```

Repeat the call to the method for each report you want to generate. You must substitute your specific information for the following parameters as shown in Table 4.

Table 4. Wget Command Parameters for Calling the ReportExecute Method

Parameter	Value in Example	Description
--load-cookie	"cookiefile1.txt"	The name of the file into which Wget stored the Web session ID. The quotation marks around the file name are mandatory in Wget.
--output-document	"pathname.pdf"	The path and file name of the report file to be stored. Make sure that the file extension used matches the format specified in the Format parameter. For example, if you are generating a PDF file, then make sure the file name ends with .pdf. The quotation marks around the path name are mandatory in Wget.

Table 4. Wget Command Parameters for Calling the ReportExecute Method

Parameter	Value in Example	Description
servername	<servername>	The name or address of the server hosting Oracle CRM On Demand. Use one set of double quotes to enclose the portion of the URL containing the following parameters: ■ servername ■ Method ■ Path ■ Format ■ Action (optional) ■ Refresh
Method	ReportExecute	This is the call to the ReportService API. The method name is ReportExecute.
Path	<ReportPathandName>	The folder path to the report in the company shared folder, including the report name. For example: ■ *Region1Sales* is a report in the shared folder named Region1Sales. ■ *Sales/Opportunities* is a report named Opportunities in the Sales folder within the company shared folder. Report Path and Name must be URL Encoded.
Format	pdf	The report can be delivered in any of the following formats: PDF, XLS, TXT, CSV, and MHTML. This value is not case-sensitive.
Action	&Action=Filter&P0=1&P1=<op>&P2=<FilterColumn1Name>&P3=<FilterColumn1Value>	*Optional*. This parameter supports application of column filters during the report generation. The values shown in the sample are specific to the Filter action. For more information on using filters in reports, see "Using Filters in the ReportExecute Method"
Refresh	Y	Must be N (for No) or Y (for Yes). If N, (recommended) then the ReportExecute method reuses the results of previously executed reports, which provides the fastest query performance for Analytical (Historical) reports. If Y, then ReportExecute method refreshes the report values, which provides the most current data.
max-redirect	100	This is the number of attempts or redirects before the request is cancelled. The default value is 20, a value of 100 is sufficiently large enough for most reports.

Using Filters in the ReportExecute Method

You can apply filters to the report generation using the optional Action parameter. The Action parameter supports a single action, which is Filter. You can apply up to six filters to a report. The format of a filter is:

```
&Action=Filter&P0=1&P1=<op>&P2=<FilterColumn1Name>&P3=<FilterColumn1Value>
```

The components of the Filter action are:

- **&Action=Filter.** Specifies the optional Action parameter is being used and the selected action is Filter.
- **&P0=1.** Specifies how many filters to apply. The maximum number of filters is six. Each filter uses three parameters: operator, column, and value. The first filter parameters are specified in P1-P3, subsequent filters are specified in sets, namely P4-P6, P7-P9, P10-12, P13-P15, and P16-P18.
  For example, to filter a report where SalesRegion = 7 and SalesForecast is greater than 10,000, the filters would look like this:

  ```
  &Action=Filter&P0=2&P1=eq&P2=Sales.Region&P3=7&P4=gt&P5=Sales.Forecast&P6=10000
  ```
- **&P1=<op>.** Specifies the operator used in the filter, such as eq (equals), lt (less than), or bwith (begins with). For a complete listing of all operators, see Table 5.
- **&P2=<FilterColumn1Name>.** Specifies the column to be filtered. The column must exist in the report. To specify a table and a column, separate the table and columns names with a period (.) as follows: ttt.ccc.
  If the table name or column name contains special characters (such as spaces), use hexadecimal encoding (for example %20 in place of a space) and enclose the table name or column name with double quotes (%22). For example to specify the *Dollar Sales* column in the Measures table:
  `Measures.%22Dollar%20Sales%22`

  NOTE: Use original column names and not display names. Also, make sure that the columns you specify in <FilterColumnName> are set up in the report as *Is Prompted* filters or as specific default filters.
- **&P3=<FilterColumn1Value>.** Specifies the value to use in the filter. If the operator requires multiple values, indicate the number of values and separate the values using a plus sign (+). For example, the bet (Between) operator needs two values. To filter on SalesRegion between 3 and 7, use:
  `&P1=bet&P2=Sales.Region&P3=2+3+7`
  If any of the values are special characters then they need to be enclosed in double quotes (%22), for example, if you are searching for the cities of Boston, Los Angeles, and Sacramento, the space in Los Angeles must be URL encoded as %20 and the term Los Angeles must be surrounded with %22:
  `&Action=Filter&P0=1&P1=cany&P2=City&P3=3+Boston+%22Los%20Angeles%22+Sacramento`

Table 5 contains the list of operators you can use in filters.

Table 5. Filter Operators

Operator	Description
eq	Equal to or in
neq	Not equal to or not in
lt	Less than
gt	Greater than
ge	Greater than or equal to
le	Less than or equal to.
bwith	Begins with
ewith	Ends with
cany	Contains any. The *value* parameter can have multiple values, each separated with a plus sign (+). For example, to find Sales Regions 1,3, and 5: `&P1=cany&P2=Sales.Region&P3=3+1+3+5`
call	Contains all. The *value* parameter can have multiple values, each separated with a plus sign (+).
like	like. The *value* parameter must contain a value and end with the wildcard %25. For example, to find account names that are like Acme: `&P1=like&P2=Account.Name&P3=Acme%25`
top	top *n* items. For example, to find the top 6 sales revenues: `&P1=top&P2=Sales.Revenue&P3=6`
bottom	bottom *n* items. For example, to find the bottom 4 sales revenues: `&P1=bottom&P2=Sales.Revenue&P3=4`
bet	Between. The *value* parameter must have two values, separated by plus signs (+). For example, to filter on Sales Region between 3 and 7, use: `&P1=bet&P2=Sales.Region&P3=2+3+7`
null	Is null (the *value* parameter must be 0)
nnull	Is not null (the *value* parameter must be 0)

Guidelines for Using the ReportExecute Method Efficiently

The following suggestions can help you use the ReportExecute method more efficiently:

- Create a folder for storing the download tool, the session cookie files, and the downloaded reports, which means you do not have to specify path names.

- In the morning, execute all the Analytical (Historical) reports with the Refresh = Y parameter to calculate all the values for those reports. All remaining calls to those reports can use the Refresh = N parameter for faster processing.
- Create a text file that contains copies of the common commands you use on a regular basis. You can then copy the commands from the text file and paste them on to the command line.
- In the text file, create one section for all your sign on and session cookie commands and one section for the API calls.
- Put a text title before each command so you can quickly identify the commands.

The following is a sample text file of commands:

```
ReportExecute method Commands


=== Sign On/Session Cookie Commands ===


Full Visibility Sign On:

wget  --content-on-error  --keep-session-cookies --save-cookies "cookiefile1.txt" --post-data "j_username=repfullvis&j_password=pass1&langCode=ENU" https://secure-<servername>/OnDemand/authenticate


Supervisor Sign On:

wget  --content-on-error  --keep-session-cookies --save-cookies "cookiefile1.txt" --post-data "j_username=repsuper&j_password=pass2&langCode=ENU" https://secure-<servername>/OnDemand/authenticate


Sales Rep Sign On:

wget  --content-on-error  --keep-session-cookies --save-cookies "cookiefile1.txt" --post-data "j_username=repsales&j_password=pass3&langCode=ENU" https://secure-<servername>/OnDemand/authenticate


=== Report Calls ===


Sales Summary, PDF format:

wget --content-on-error --load-cookie "cookiefile1.txt" --output-document "SalesSummary.pdf" “https://secure-<servername>/OnDemand/user/ReportService?Method=ReportExecute&Path=<ReportPathandName>&Format=pdf&Refresh=Y” --max-redirect=100


Sales Summary, Text format with two filters:
```

```
wget --content-on-error --load-cookie "cookiefile1.txt" --output-document
"SalesSummary.txt" “https://secure-<servername>/OnDemand/user/
ReportService?Method=ReportExecute&Path=<ReportPathandName>&Format=txt
&Action=Filter&P0=2&P1=eq&P2=Sales.Region&P3=7&P4=gt&P5=Sales.Forecast&P6=10000
&Refresh=Y” --max-redirect=100
```

Troubleshooting the ReportExecute Method

The Oracle CRM On Demand Report Services API can encounter problems with the command line and parameters, and also with generating reports. The Oracle CRM On Demand Report Services API responds to error conditions by returning an error message. To resolve a problem, see Table 6.

Table 6. ReportExecution Method Error Messages

Problem	Cause	Solution
Your ReportService request failed. *VariableName* method is incorrect or not supported. Please check for supported methods and try again.	You specified an incorrect method name.	Check for supported methods and try again.
Your ReportService request failed (Required parameter *VariableName* missing in method *VariableName*). Please check your syntax and try again.	One or more parameters are missing.	Review the parameters used and revise the command.
Your ReportService request failed (Incorrect delivery format *VariableName* provided; supported delivery formats are *VariableName*). Please check your delivery format and try again.	You specified an incorrect delivery format.	Review the delivery format used and specify a supported format.
Your ReportService request failed (Incorrect Action parameter value provided; supported Action is 'Filter'). Please check your Action parameter value and try again.	You specified the Action=Filter parameter incorrectly.	Specify the Action=Filter parameter correctly.
Your ReportService request failed (Incorrect P0 parameter value provided; supported values only within range 1-6). Please check your P0 parameter value and try again.	You specified the P0 value (number of filters) outside the range of 1-6.	Review and specify the P0 parameter correctly within the range of 1-6.

Table 6. ReportExecution Method Error Messages

Problem	Cause	Solution
Your ReportService request failed (Specified P0 parameter value and subsequent parameters not matching). Please provide proper parameters and try again.	The number of filters specified in the P0 value does not match the subsequent filters provided.	Make sure that the number of filters provided matches with the P0 parameter value.
Your ReportService request failed (Operator is not in the specified list of operators). Please provide proper operator and try again.	You specified an operator that is not in the specified list.	Use only supported operators as shown in Table 5 on page 17.
Your ReportService request failed (Table column parameter is not existing or not specified properly). Please provide the table column parameter correctly and try again.	You specified the table column incorrectly.	Review and make sure the table column does exist and is specified properly, double-quotes (%22) are provided where needed and encoding is done for spaces (%20).
Your ReportService request failed (Column value parameter is not specified properly). Please provide the column value or values correctly and try again.	You specified the column value incorrectly.	Review and make sure that the number of values match the operator, multiple values are separated by a plus (+) sign, and are enclosed within double quotes (%22) where needed.

Table 6. ReportExecution Method Error Messages

Problem	Cause	Solution
Your ReportService request failed (Incorrect refresh option *VariableName* provided; supported refresh options are 'Y' or 'N'). Please check your refresh option and try again.	You specified an incorrect refresh option.	Review the refresh option used and specify a supported option.
Your ReportService Report Execution request failed (Internal error). Please ensure that you can run the report in CRM On Demand and try again.	The ReportService encountered a problem that is not related to a parameter in Table 4 on page 14. Possible causes include lack of permission to the report folder, the report timed out, memory issues, or the report name was misspelled or does not exist.	Using Oracle CRM On Demand and the same account, try to generate the report. If the report is generated, then check the file name to make sure the correct values are used, and that you have not exceeded service allotments.

Error messages are captured in the output file specified for the generated content in the ReportExecute method.

4 Using the ReportList Method

This chapter describes how to access the ReportList method. It includes the following topics:

Overview of the ReportList Method

The ReportList method generates a summary list of reports in the company shared folder.

To generate the summary list of reports through this method, you need a scripting language, such as JavaScript or Perl, or a command line download tool such as Wget or cURL. The examples in this guide use the Wget tool, but the concepts are the same for all scripting languages and download tools.

The download tool calls the ReportList method and specifies the folder path for which the list of reports is to be generated. The ReportList method accesses the specified folder and creates a zip archive file containing a UTF-8 encoded CSV file with a summary list of reports. The download tool then transfers the file generated by the ReportList method and stores it in the same location from where the report list service call is made or in a location specified with the output file.

The ReportList method uses the data visibility controls in Oracle CRM On Demand. The user account you use to sign on to Oracle CRM On Demand through the ReportList method by way of the download tool, determines the shared report folders you have visibility into.

Sequence of Operations for Using the ReportList Method

The ReportList method is not a stand-alone command. First, you must make sure you have a download tool (see "About the Download Tool" on page 9). Then use the following sequence:

1. Sign on to Oracle CRM On Demand and open a cookie file (see "Creating a Session Cookie File" on page 9).
2. Call the ReportList method (see "Calling the ReportList Method" on page 24).
3. Close the Oracle CRM On Demand session (see "Signing Off from Oracle CRM On Demand" on page 10).

Calling the ReportList Method

When you have the Web session ID stored in a cookie file, you can then use the download tool to make the call to the ReportList method and get the summary list of reports.

When using Wget, the format for the call is as follows:

```
wget --content-on-error --load-cookie "cookiefile1.txt" --output-document
"reportlist.zip" "https://secure-<servername>/OnDemand/user/
ReportService?Method=ReportList&Path=<ReportFolderPath>"
```

Substitute your specific information for the following parameters as shown in Table 7.

Table 7. Wget Command Parameters for Calling the ReportList Method

Parameter	Value in Example	Description
--load-cookie	"cookiefile1.txt"	The name of the file into which Wget stored the Web session ID. The quotation marks around the file name are mandatory in Wget.
--output-document	"reportlist.zip"	The path and file name of the zip file to be stored. The quotation marks around the path name are mandatory in Wget.
servername	<servername>	The name or address of the server hosting Oracle CRM On Demand. Use one set of double quotes to enclose the portion of the URL containing the following parameters: ■ servername ■ Method ■ Path (Optional)
Method	ReportList	This is the call to the API. The method name is ReportList.
Path	<ReportFolderPath>	Optional. The path to the company shared folder or a folder within the company shared folder. ReportFolderPath must be URL Encoded. If the ReportFolderPath is not specified in the URL, then the ReportList method fetches the list of reports from the parent company shared folder. If the ReportFolderPath is specified, then the ReportList method generates the list of reports from the specified folder and sub-folders. The ReportList method always respects access permissions to folders and their contents.

The following is a sample of a ReportList method call:

```
wget --content-on-error --load-cookie "cookiefile1.txt" --output-document
"reportlist.zip" "https://secure-<servername>/OnDemand/user/
ReportService?Method=ReportList&Path=Sales%20Activity%20Reports"
```

The ReportList method generates a summary list of reports in the company shared folder. The ReportList method accesses each folder and creates a zip archive file containing a UTF-8 encoded CSV file with the summary list of reports. Table 8 shows the fields available in the summary list report.

Table 8. Fields in the Summary List Report

Field	Description
Report Name	The name of the report in the language in which it was entered.
Shared Folder Path	The company shared folder hierarchy (full path excluding the company shared folder name itself).
Created	The date the report was created in the format for the user's locale, not adjusted for daylight saving time.
Modified	The date the report was modified in the format for the user's locale, not adjusted for daylight saving time.

The Created and Modified times provided by the ReportList method might be different from the times reported by other products because the ReportList method does not adjust for daylight saving time.

Troubleshooting the ReportList Method

The ReportList method can encounter problems with the command line and parameters, and also with generating report lists. The ReportList method responds to error conditions by returning an error message. To resolve a problem, see Table 9.

Table 9. ReportList Method Error Messages

Problem	Cause	Solution
Your ReportService request failed. The folder path is either incorrect or you don't have permission to view the specified folder.	Incorrect folder path or insufficient access	Review the folder path provided and ensure that it is correct. Also ensure that you have sufficient permissions to access the given shared folder
Your ReportService request failed (missing folder path in parameter "Path" in method "ReportList"). Please check your syntax and try again.	Incomplete folder path	If the Path parameter is used, review that the folder path provided is complete and accurate. Otherwise leave out the Path parameter completely to get a list of all reports under the company shared folder that you have access to.
Your ReportService Report List request failed (Internal error). Please ensure that you can run the report in CRM On Demand and try again.	The ReportService encountered a problem that is not related to a parameter. Possible causes include lack of permission to the report folder, memory issues and so on.	Using Oracle CRM On Demand and the same account, try to view the shared folder. If you are able to view the folder, then check that you have not exceeded service allotments.

Error messages are captured in the output file specified for the generated content in the ReportList method.

Index

U

W