Search & Export Report Data

Version: Draft 2

Abstract

This document describes how to export the data from a saved report document.

Document Revisions

Version	Date	Description of Changes
Draft 2	08/12/2005	Updated by Stanley Wang
Draft 1	08/09//2005	Initial draft.

Table Of Content

1. Introduction

Searching a document is a fundamental feature for various document formats including Word, PDF, etc. In a reporting context, searching not only means full-text search, but is also closely coupled with data extraction. For example, if a report contains a table, the data for the table would have been stored at report generation time in the report document and therefore readily available for extraction; search, on the other hand, allows specification for other filtering criteria to limit the amount of data that is returned.

Accessing and extracting data in a report document is also important for using report as data source for other reports. Several usage scenarios demand for this feature. First, report document provides a snapshot of the database at a specific time, and is sometimes the only snapshot that is available; second, government regulations require that companies keep important documents for several years. Accessing information stored in the document when required is therefore essential.

This document provides a high-level summary of the API and UI level changes that support searching and extracting data from report document.

2. Search

We define search as text-based search and assume that searching data stored in the report document falls under the umbrella of data extraction.

2.1 Full-Text search

When a report is rendered to PDF, RTF, or Excel, full-text search is already supported by the viewers that open the file. No additional enhancement is needed in such viewing environment.

Full-text search in HTML environment is needed because HTML reports are usually paginated. Browser search can only identify matches in the current page.

BIRT engine will add a Search task that supports the following API for searching a report:[1]

```
int search(String searchString, int startOnPage, boolean exact,
           boolean caseSensitive);
```

This function searches a specific string in the report, starting from page *startOnPage*, and returns the page number where the first match is. It is assumed that once the specific page number is returned, the caller (for example, the viewer) can request for the specific page from engine and displays the page. Highlighting the match in the HTML page could be achieved by JavaScript code in browser. It is the responsibility of the caller to keep state if "Next" match is to be supported. In addition, if the page has more than one match, it is also assumed that moving from one to the next match is done through browser scripting.

Full text search matches strings that appear both as part of the report data or static content (such as labels). If there is a text element with HTML formatting tags, the tags are neglected. For example, a text item with "<B>abc<B>de" matches criterion string "abcde". If exact is false, search matches substrings, i.e., if the search criterion is "cd", text item "<B>abc<B>de" is counted as a match. Put differently, the matching rule is the same as searching an HTML page.

The search string is matched with data after formatting. That is, if the data is 2000, but is formatted as "2,000", a search for 2000 will not match the data item. Similarly, if a label reads as "Label" in one language, and is localized to "LEBAL" in the current viewing locale, search would result in a match.

Full text search is not available in BIRT 2.0.

2.2 Indexing Output File

Search engines such as Google and Lucene can search several document formats. For example, Google Desktop search can search Word files and Emails. Indexing such content is essential for high-performance search. It is desirable that outputs generated by BIRT can be indexed by such search engines. Except for file formats that are already supported by the search engine's indexing algorithm, BIRT 2.0 will not include any special code to assist searching by Google or Lucene. Such features, however, could be part of future release.

[1] The report document that search operates on is specified as part of the engine task.

3. Data Extraction from Report Document

3.1 Output Format

Data Extracted from a report document would either be displayed by an extraction UI, or returned in CSV, TSV or XML format. When BIRT starts to support Excel emitter, the search results could also be returned as an Excel file.

When exporting to CSV/TSV, the first row of the output includes the column names.

3.2 Data Extraction Metaphors

Data Extraction would be supported at several different levels:

1. Report Level. Assume we have a customers report, the report level data extraction fulfills the need such as “Get all the data in this report for those customers in California”. Because a report could have multiple data sets, the CSV/TSV format will include the first data set. The Excel format would place each data set in a separate sheet. The XML format could return data from multiple data set.
2. Dataset Level. Extraction based on dataset extracts all the data stored for a data set, with transformations applied.
3. Report Item. Extraction based on report item extracts all the data displayed in all the instances of a certain report item. We only support extraction on the following report items:
    a. Listing Item (table or list)
    b. Chart
    c. Crosstab (Not supported in BIRT 2.0)
4. Report Item Instance. Extraction based on report item instance extracts the data displayed for a specific report item instance.

For extraction at dataset level, it should be noted that report document may not have stored all the columns in the dataset query. For example, the database query might select 10 columns, but the report only uses 5. Then only 5 of the columns are available for data extraction. In addition, BIRT allows report designer to control how much data is stored, so that even if a column is not used, it is still included in report document for data extraction. For example, if the report developer has configured another 3 columns to be included, he can extract data for 8 columns.

As another interesting case, a report may include a query to select 10,000 rows, but displays the top 20 rows. The report document only stores the 20 rows. So extraction may not yield more than 20 rows.

The data rows extracted from a report document are in the same order as the data is stored in the report document, which could be different from the order data is received from DB, or specified in dataset query string. However, data transformations can be specified to change such order.

Extraction at the report level can be achieved by repeatedly retrieving data for each dataset.[2] Extraction at the report item instance level would include computed columns as normal columns.

3.3 Metadata

Because the extracted data may be different from what dataset query specifies, it is important for the user to query for metadata before extraction. BIRT provides the following capabilities for returning metadata information:

1. List all the datasets in a report.
2. List all the columns in a dataset, including their names and data types.
3. List all the columns for a report item, including their names and data types.
4. List all the columns for a report item instance, including their names and data types.

3.4 Data Transform

Frequently, a user wants to apply transformations before the extracted data are returned. BIRT supports two ways to specify transformations: an API-based approach and a query language based approach. The API approach is limited in functionality, but addresses the common use cases; the language based approach is similar to SQL and is more powerful.

3.4.1 API-Based Data Transform

BIRT Extraction API would allow specification of the following transformations:

1. Column Selection.
2. Simple Filters. The LHS of a filter must be a simple column (not an expression). The following operators are supported:
    a. >, <, =
    b. Started with, i.e., abc* matches all strings start with abc
3. Multiple Simple Filters. Only AND relationship is supported. Operators such as BETWEEN can be simulated by using the < and > operators.
4. Sort. Sort can be based on multiple columns, and each column supports ascending and descending. No complex sort expressions are supported.

3.4.2 Query Language Based Data Transformation

If one views report document as a special type of data source, a query language could be defined to access such a data source. BIRT will support a simple query language that covers only the features in the last subsection, which are also available through API. The query language will be enhanced in future releases. The purpose of having a query language is to handle free-form query and extraction needs, so the API based approach do not necessarily need enhancements when the query language is enhanced.

[2] Do we really need a report level extraction metaphor?

The query language will use the same syntax as SQL. It is very likely a (very small) subset of SQL.

Query language based data transformation is not supported in BIRT 2.0.

4. UI Specifications

The last two sections focus on search and extraction operations seen by application developers or system integrators. These people create UI for end users to access search and extraction functionalities. BIRT will provide a reference UI implementation in BIRT Viewer. The rest of this section describes the UI-related behaviors in BIRT Viewer.

4.1 Search UI

A search box is included in the viewer toolbar, from which a text string can be entered. BIRT engine returns the first match, and the page with the first match is displayed, with the matching string highlighted. There is a next button next to the search box, clicking on which moves to the next match. The next button is grayed out when search reaches the end of file. If search yields no matches, a window is popped up to let user know.

4.2 Extraction UI: Report & Dataset Level

Viewer UI provides a way for user to navigate through datasets within a report. The user can select a dataset and asks for extraction. Additional transformations can be specified. Metadata about a dataset helps guide the user for extraction.

The UI should offer preview of the extracted data. The user can retrieve all the data by selecting an output format and download a file.

There are no links from the previewed data to a specific report page.

4.3 Extraction UI: Report Item Level

An end user selects a table, chart, or crosstab and initiates data extraction by clicking on a toolbar button, or selecting an action from the context menu. By default, no transformation is required. This is to make the common case of downloading data for a chart or table as simple as possible. There are advance options that allows entering transformations.