BPS22 Page-on-Demand Viewing

Draft 1: August 16, 2005

Abstract

This document describes the page-on-demand viewing features for BIRT 2.0

Document Revisions

Draft	Date	Primary Author(s)	Description of Changes
1	August 16, 2005	Stanley Wang	Initial Draft

Contents

1. Introduction

Page-on-demand, also referred to as demand paging, is a mechanism for dealing efficiently with large report documents over a remote connection such as the Internet. Instead of reading an entire report into memory or sending an entire report over a network, pages are requested by the application and sent as needed. This method of data transfer improves response time for the end user and optimizes resource usage.

When integrating BIRT into applications, particularly web applications, page-on-demand capabilities are needed to ensure that the application can be built with acceptable performance.

A table of contents shows the hierarchical structure of the report to the end user. The hierarchy is defined by the group and sub-report structure of the report, with customizations possible by the report developer -- such as providing custom text for each node in the hierarchy. When the user clicks on a TOC link, he is taken to the first page of that group. This creates another way to quickly navigate to a section of the report.

When a report is large, generating the report could take a long time. Having the end user waiting before the screen until report generation is finished is not efficient or user-friendly, because most reports are generated sequentially, and the pages that are already generated seldom change and are ready fro viewing access. Progressive viewing allows a user to view a report while it is still generating.

This document specifies the behaviors end user sees about page-on-demand viewing, TOC and progressive viewing. It also covers the operations that application developers can access to build such UI.

2. UI Features

2.1 Page-On-Demand Viewing

End user sees a paginated viewing tool bar with several buttons/links or input box. The toolbar has an input box for user to enter a specific page. If the number entered is smaller than -1 or larger than the last page, the first or last page is displayed. The UI has buttons for going to next, previous, first, or last page. If the user is already on 1st (last) page, the first and previous (last and next) page buttons are disabled.

The UI should also display the number of viewable pages in the report. Different users may be able to view different number of pages in the report if the report has certain security features built in.

During progressive viewing or interactive viewing environments when the total number of pages can not be decided, it may be displayed as a question mark.

2.2 TOC

TOC is displayed as a separate tree to the left of the report content. TOC trees are expanded incrementally, meaning that clicking the root of the tree only expands the next level. As a result, TOC performance is bounded by the number of items in the expanded level, not the total size of the TOC entries.

If each TOC level has more than 500 entries, the TOC is truncated there, with a 501st entry added reading “More entries truncated”.[1]

Clicking on each TOC entry navigates to the first report page that displays the TOC group.

2.3 Progressive Viewing

Progressive viewing is only relevant when the user wants to “generate and view” a report. Right after first page is generated, viewing could start. The end user would see the total number of pages a X+, where X is the number of pages that are already generated. The first page will be automatically refreshed every 10 seconds. User can navigate to different pages as long as the page has been generated. Summary charts or report items that use summary information may come initially as missing images, but will appear when the report finishes generation.

If the user has navigated to page Y, the next refresh should retrieve page Y instead of page 1.

When report generation finishes, X+ is replaced by the actual page number. From then on, no auto-refresh will happen.

TOC and search are not available before report is fully generated.

3. Report Engine operations

This section describes report engine operations that application developers can use to construct the page-on-demand viewing UI.

3.1 Page-on-Demand Viewing

Given a report document, report engine supports two page-related operations:

1. Render page X.
2. Get Page Count.

It is application developer’s responsibility to keep an active viewing session, which tracks the page that is currently viewed. Knowing the page count and current page, going to next, first, previous or last page can all be implemented accordingly.

BIRT viewer provides a reference implementation of page navigation UI.

3.2 TOC

At design time, the `toc` property for each group and subgroup is by default set to the same as the group key; the toc property is left blank for other report items. Users can customize the toc expression or add toc expressions to other report items. TOC index is generated at factory time and stored into report document.

Given a report document, report engine supports an operation to get TOC nodes as an array. The caller can also specify the TOC parent node, the TOC depth, and the

[1] If a report has more than 500 TOC entries at one level, the report developer should consider adding additional group levels for the TOC to be useful.

maximum number entries returned for each TOC level. By default, the parent node is report root, the TOC depth is 1, and the number of TOC entries is infinite. To use only the first 500 entries in TOC, one can specify the maximum entries to be returned for a level to 500.

3.3 Progressive Viewing

BIRT 1.0 supports an operation "RunAndView", and implements the feature by avoiding using a report document. In 2.0, the same operation will be used to support progressive viewing. Report generation process will support a CheckPoint event. It defines a check-pointing event handler interface for the factory host to use. By implementing the interface, the host is informed after the 1st, 5th, 10th and 50th page is generated, and for every additional 50 pages that are generated. The host can decide what to do on the viewing side to achieve the progressive viewing behavior.

For example, if the factory and presentation engines coexist in the same application server, the "RunAndView" request could be achieved by adding an event handler to start the rendering part in a different thread after 1st page is rendered.[2] If they are separate, a message from factory engine to the presentation engine may be necessary.[3]

The get page count operation returns the number of pages that are committed through checkpointing so far.

3.4 Security Considerations

If a report has security features implemented, different users may see different report content. Report pages seen by each user is based on the content that the user could actually see. The TOC only includes groups that are viewable by the end user. Progressive viewing is on the other hand not available until the report generation is finished.

BIRT 2.0 does not implement advanced security features.

[2] In fact, the rendering thread may have been created before but is blocking, and the event handler only unblocks it.

[3] File locking may be the synchronization mechanism used to achieve progressive viewing across processes. Notice only the checkpoint file, i.e., a small file (in size) in the whole report document archive needs to be locked.