A User-Oriented Approach to Scheduling Collection Building in Greenstone

Wendy Osborn[1], David Bainbridge[2], and Ian H. Witten[2]

[1] Department of Mathematics and Computer Science
University of Lethbridge
Lethbridge, Alberta, Canada
wendy.osborn@uleth.ca
[2] Department of Computing Science
University of Waikato
Hamilton, New Zealand
{davidb,ihw}@cs.waikato.ac.nz

Abstract. We propose a user-oriented approach for the automated and scheduled maintenance of Greenstone digital library collections. Existing systems require the user either to add new data manually to a collection, or to have programming knowledge in order to use existing application programming interfaces (APIs) in order to automate scheduled collection updates. The Greenstone Scheduler can automate the construction of any existing collection, and schedule the construction to occur periodically. This is accomplished through incorporating a module specific to this purpose into the Greenstone Librarian Interface.

1 Introduction

A wide range of applications generate multimedia documents, such as images and video on a daily basis. For example, many municipalities have a photo-radar program to catch vehicles traveling over the speed limit, or traveling in bus lanes during rush hour. A vast number of pictures of vehicle license plates are created every day. If these images are organized into a digital library, for instance, the collection would need to be updated regularly to incorporate new images. Another example of an application that requires periodic updating includes a resource of information across several post-secondary institutions [6].

When documents and metadata are added to a digital library collection on a regular basis, such as hourly or daily, an automated and scheduled approach to collection maintenance is preferred. An automated approach should not be time consuming for the user, leaving their time available for other important tasks.

Existing digital library software such as DSpace [8] and Fedora [2] require that items be added manually to the collection. In Fedora, data is retrieved at the time of viewing. However, the location needs to be manually configured. Further, although Fedora and DSpace do provide application programming interfaces (APIs) to extend functionality, programming knowledge is required for using an API and setting up tools based on it.

G. Buchanan, M. Masoodian, S.J. Cunningham (Eds.): ICADL 2008, LNCS 5362, pp. 236–245, 2008.

A module for automating and scheduling collection maintenance in Greenstone has been proposed recently [5]. The Scheduler can automate the construction of any existing collection, and schedules the construction to occur on an hourly, daily or weekly basis. In addition, the owner of a collection is still free to update the collection manually whenever they want. Further, the Scheduler interacts with the existing task scheduling mechanism on the host system, which keeps the Scheduler minimal, yet powerful.

In this paper, we present the incorporation of the Scheduler into the Greenstone Librarian Interface [10], a user-friendly graphical interface for creating digital library collections. By providing a user interface for the Scheduler, we improve its usability and provide further abstraction of the scheduling process.

2 Greenstone

Greenstone [11] is a software suite for creating digital library collections and making them available globally via the internet or removable media. A collection can contain documents of different formats. Over thirty formats are supported through a plugin architecture including images, postscript, PDF, audio, and formatted and unformatted text that extract full text and metadata where possible. A Greenstone collection can be customized in many ways. For example, a collection owner can customize the types of documents that can appear in the collection, the appearance of the interface of the collection, and the indices and classifiers that other users can use to access the collection.

A collection is created or modified by following four steps:

1. *document addition.* New documents that will become part of a collection are copied into the import directory for the collection.
2. *document importation.* The documents in the import folder are now processed for the collection by specifying an import command. Different importing options can be specified by the user. All imported documents are located in the archive directory.
3. *accessor creation.* Indices and classifiers are created by using a build command. The resulting indices and classifiers are placed in the build directory.
4. *collection activation.* Finally, the collection is activated by copying the indices and classifiers from the building directory to the index directory.

3 Greenstone Librarian Interface

The Greenstone Librarian Interface [10] is a graphical user interface that provides a user-friendly method to build and configure Greenstone collections. It incorporates the four steps above. In addition, the Librarian Interface allows a user to create new collections, select metadata sets, configure which document types to allow, and select import and build command options.

Importing converts source document into a canonical XML format; building processes the canonical XML format to create the necessary full-text index and metadata files.

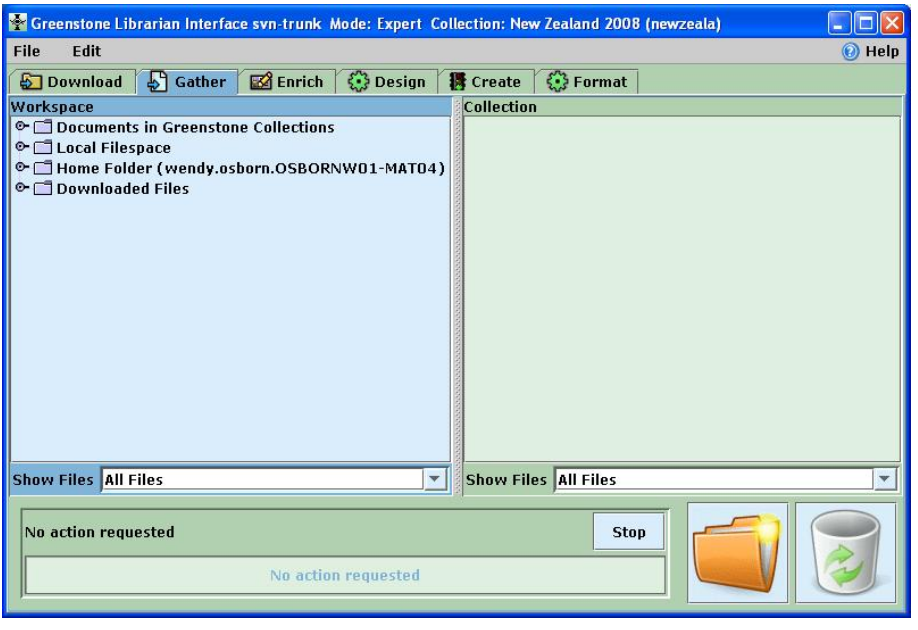


Fig. 1. The Greenstone Librarian Interface

3.1 Command-Line Scheduling

The Scheduler is a command-line program that is responsible for both creating a building script for a specific collection and for interacting with the scheduling service on the operating system [5]. The original version of the Scheduler takes as command-line arguments the name of the collection, the import and build commands (and all of their required arguments), and the frequency of execution (hourly, daily or weekly). The output from the Scheduler is a customized Perl script that rebuilds the collection, and modifications to the scheduling service to execute the script at the frequency specified. The scheduling service employed by the Scheduler is Cron [3], because it is available for all major platforms. Unix, Linux and Mac OS X run Vixie Cron [9]. Versions of Cron that exist for Windows includes Pycron [7].

For example, suppose we want to schedule a daily build of the collection *pics*. A call to the Scheduler would resemble the following:

```
schedule.pl pics "import.pl -removeold pics"
                 "buildcol.pl -removeold pics" daily
```

Figure 2 shows the resulting build script and corresponding *crontab* record for the collection *pics*.

```
#!/usr/bin/perl
$ENV{'GSDLHOME'}="/home2/gsdl/gsdl";
$ENV{'GSDLOS'}="linux";
$ENV{'GSDLLANG'}="";
$ENV{'PATH'}="/bin:/usr/local/sbin:/usr/local/bin:/sbin:/usr/sbin:
/usr/bin:/usr/X11R6/bin/:/usr/local/gsdl/bin/script:
/usr/local/gsdl/bin/linux";
system("import.pl -removeold pics");
system("buildcol.pl -removeold pics");
system("\\rm -r /gsdl/collect/pics/index/*");
system("mv /gsdl/collect/pics/building/*
             /gsdl/collect/pics/index/");
system("chmod -R 755 /gsdl/collect/pics/index/*");

00 0 * * * /gsdl/collect/pics/gsdl.pl
```

Fig. 2. Sample Building Script and Crontab Record[5]

4 Scheduling in the Librarian Interface

The Scheduler is a minimal, yet powerful tool for maintaining Greenstone collections. It hides the details concerning the collection building process, and also the details for scheduling a Cron task. However, the Scheduler has two main limitations. The first is that the user is still required to know the syntax and command-line arguments for both the import and build commands in order to perform scheduling. The second is that, currently, notification of collection building success—or failure—is still dependent on the version of Cron (and indirectly, the operating system it is running on) that is used.

The Librarian Interface provides a user-friendly tool for building collections, including configuring the import and build commands. Therefore, it is ideal to extend the Librarian Interface to allow the configuration of the Scheduler as well. Figure 4 depicts the Librarian Interface extension to support the scheduling of collection builds. Currently, the Create panel is displaying most of the arguments (i.e. Schedule Options) for the Scheduler.

Notice that no explicit options exist for the import and build commands. This is because the import and build commands are created based on the arguments selected by the user from the Import Options and Build Options. Therefore, the user no longer needs to know the exact syntax of the import and build commands!

5 Example: Collecting Pictures While Traveling

Before we discuss extensions to both the scheduler and the Greenstone Librarian Interface, we present a simple application of scheduling from the Librarian Interface. In this scenario, we have a traveler who wants to post pictures of their trip in a Greenstone collection for her friends to view. Instead of waiting until

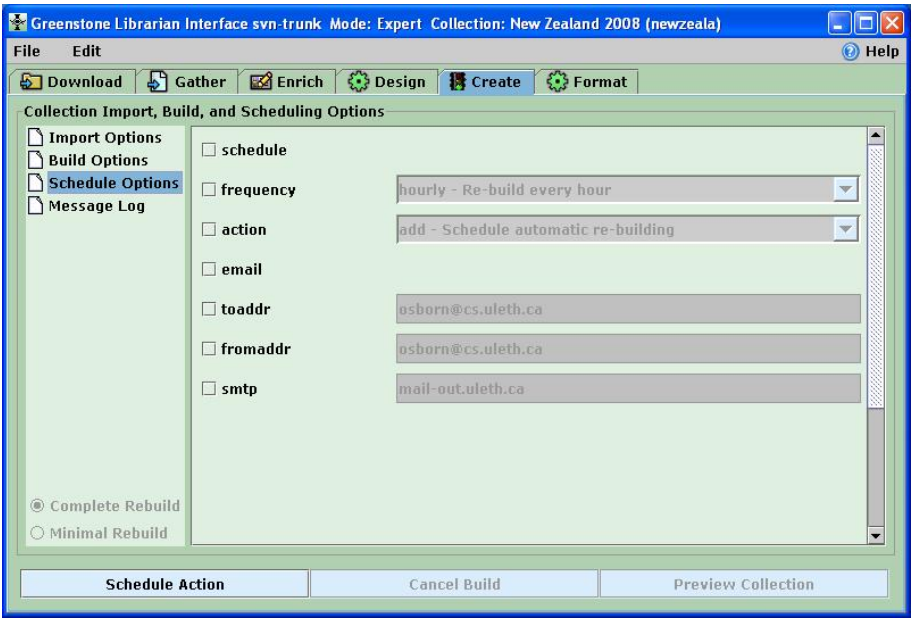


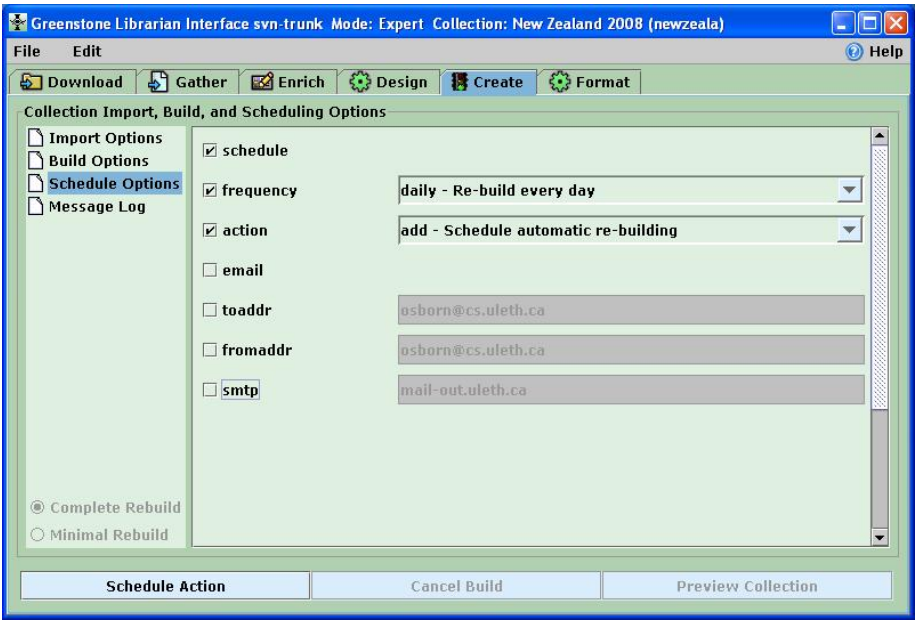
Fig. 3. The Schedule Options Pane

the end of the trip, the traveler wants to post her pictures from each day, incrementally adding to the collection on a daily basis. The traveler does not want to worry about obtaining the Librarian Interface to rebuild the collection while traveling. Instead, she simply wants to upload the pictures to the import folder of her collection, and have her collection rebuilt automatically and on a daily basis. This can be accomplished by setting up a scheduled, automatic rebuild of the collection of travel photos from the Librarian Interface.

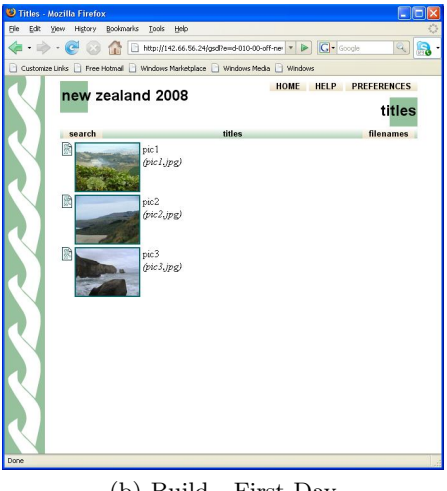
First, before departing, the traveler runs the Librarian Interface and creates a new collection. Then, the user selects the Create tab to display the collection creation pane. From here, clicking on Schedule Options will display the available options for setting up a scheduled, automatic collection build of the collection of travel pictures. Figure 4(a) depicts the available scheduling options, which are displayed with default and derived values as appropriate. Here, the traveler selects schedule, which indicates that she wants to set up a scheduled, automatic build. Also, she selects a frequency of hourly and an action of add (or, to create a new scheduled build).

Next, the traveler clicks on Schedule Build. This will set up the building script for the collection of travel pics, as well as the *crontab* record that will indicate to Cron that the collection is to be rebuilt daily. The collection is now ready to be re-built while the traveler is away.

At the end of the first day of travel, she uploads three pictures, which are added to the collection when the collection is re-built automatically overnight. The updated collection is depicted in Figure 4(b). The next day, the traveler



(a) Scheduling Options Pane



(b) Build - First Day

(c) Build - Second Day

Fig. 4. Collection Build Scheduling

uploads three more pictures. When the collection is re-built overnight, these pictures are added to the existing collection. Figure 4(c) depicts the updated collection with the new pictures.

Although not shown here, the user can switch to the Import Options and Build Options and select any options that are required for collection importing

and building. These options are incorporated into the automatic building script that is created for the collection. In addition, the user can manually build and configure their collection as many times as necessary to confirm the right sequence is being performed, before setting up a scheduled, automatic build.

6 Implementation

In order to enable scheduled, automatic building from the Librarian Interface, several issues needed to be addressed. Some issues require modification of the Greenstone scheduler itself, while others are modifications to the Librarian Interface. We present and discuss the top issues here.

6.1 User Modes

The Librarian Interface has four modes of use— Library Assistant, Librarian, Library Systems Specialist, and Expert—stepwise increasing the functionality available to the user [10]. It was important to determine which modes would have access to the Scheduling Options, and for those modes that were granted access, how much access each would be granted. It was decided that Library Systems Specialists and Experts would be provided access to the Scheduling Options. In addition to determining which options to make available to each user mode, it is also necessary to determine how the Scheduler would interact with the existing import and build functionality in the Librarian Interface.

Mode Level for Options. The Scheduler was first modified so that the passing of command-line arguments conformed with that of other Greenstone commands, such as import and build. This also allowed us to easily specify which user mode could have access to each argument when it is added as a Schedule Option in the Librarian Interface. The Expert user mode is granted full access to all Schedule Options. The Librarian Systems Specialist is only granted limited access to options. This user mode can only specify whether or not to schedule a collection build with the default values—a frequency of hourly, and no sending of email.

Scheduling and Building. Another important decision that needed to be made was how the Scheduler would interact with the collection building functionality of the Librarian Interface. The question asked was, should scheduling be done at the same time as collection building, or be a completely separate task?

For Expert mode, the functionality for Scheduling is separate from that of collection building. This is because an expert user may want to configure and manually re-build their collection before scheduling an automatic re-build of it. For Library System Specialist mode, the Scheduling functionality is done at the same time as collection building. If the specialist chooses to schedule, a new scheduled build is created. If the specialist chooses not to schedule, any existing scheduled builds are deleted.

6.2 Cron Event Logs

It is important to maintain logs that keep track of the outcome of a scheduled collection build. Both Vixie Cron and Pycron maintain a log that keeps track of the attempted execution of all scheduled tasks. Neither scheduling service keeps track of the success or failure of a scheduled task, nor do they keep track of the output of a task. In addition, to view logs created by Vixie Cron requires the user to have root access.

Another desirable feature of maintaining logs is to be able to only record output that is considered important, and to disregard all other task output. For example, if the output from the input and build commands of Greenstone is required, but the output from moving indices and classifiers is not considered important, the log should reflect this.

Thererfore, the Greenstone scheduler is modified in two ways to handle the logging of building script output. The first is to create a custom log for each execution of the building script. The collection building script creates a unique filename every time it is executed by using a timestamp. The second is to specify in the collection building script which actions will have its output redirected to the logfile. The actions whose output is to be ignored will have its output redirected to the 'bit bucket' (e.g. /dev/null/ in Linux).

6.3 Email Notification

It is also important than an email notification service be provided, which will inform users of the outcome of their scheduled collection build. We handle email notification from the Scheduler and Librarian Interface for the following reasons:

1. *User notification.* Whether Cron notifies users about the outcome of a scheduled task depends on the its implementation. For Vixie Cron, the outcome of a task (i.e. output from either successful task completion, or error output) is emailed to the owner of the task. Pycron does not send email notification.
2. *Flexibility of Notification.* In Vixie Cron it is possible to suppress email notification, either by setting an environment variable to null, or by redirecting all output to a file or the 'bit bucket'. However, this is normally an all-or-nothing event—either all output, or none, is sent by email. Similar to logging, a desirable feature would be to send email that contains only the most important parts of the building process, and ignores other parts of the building process.
3. *Greenstone Email Support.* Greenstone comes with a Perl email script, that is a wrapper for the Perl sendmail command. The email script is platform-independent. Therefore, it is ideal to use it to provide a uniform way to send email concerning the execution of a scheduled task. In addition, the script does not require the piping of task output directly to it, but instead can send the contents of a file.

Therefore, both the Librarian Interface and the Scheduler are modified so that email notification is handled in a uniform and user-friendly manner across all operating systems.

First, options have been added to the Scheduler that are required for the Perl email script—specifically, a flag to specify that email will be sent (-email), the sender (-fromaddr), receiver (-toaddr) and the email server that will be used to send the email (-smtp). Also, the corresponding fields exist in the Schedule Options pane of the Librarian Interface. In order to assist users in using the email features of scheduling, the Librarian Interface attempts to populate the fields -toaddr, -fromaddr, and -smtp in the Schedule Options pane by consulting the configurations for the Librarian Interface and Greenstone. If suitable values are available from these sources, they are assigned to the appropriate fields.

Second, the output from the building script must be captured and re-directed to the Perl email script. The capturing of output already takes place, in the event log. This serves as the file that the email script will send to the user. Also, since it contains only the output that is considered important, this will be reflected in the email message as well.

Finally, the Scheduler is modified so that, if specified, the generated build script will send an email message containing the contents of the log to the specified recipient. An added bonus is that if email is not specified, the log can still be consulted by the user if required.

6.4 Scheduled Building in Isolation

An important modification to the Scheduler is to ensure that a scheduled build is completed in its entirety without interference from another scheduled build. A build may take a significant amount of time depending on the size of the collection—from seconds for a small one to 33 hours for a collection containing 20 GB of raw text and 50 GB of metadata [1]. To handle this, the Perl script for the collection checks for a lock file, which indicates that a collection build is underway. If the file exists, the collection owner is notified via email and information is placed in the event log. Otherwise, a lock file is created before the scheduled build begins, and is removed when the build finishes. This ensures that multiples builds do not occur concurrently.

7 Conclusion

In this paper, we propose and discuss the incorporation of the Greenstone Scheduler into the Librarian Interface. This overcomes two limitations of the Schedule—the requirement to know the syntax and command-line arguments for both the import and build commands, and the inconsistency of notification of collection building success or failure. Providing an interface to the Scheduler improves its usability and provide further abstraction of the scheduling process from the user.

Some future directions of work include the following. The first is to allow the user to select a specific period of time (e.g. 20 minutes after the hour) for their collection to be re-built. Currently, collection building occurs at the top of the hour (hourly), at midnight (daily) and on Sunday at midnight (weekly). The

second is support for dependencies between fields in the Librarian Interface. For example, if a user selects the email option, it requires -toaddr, -fromaddr, and -smtp. Currently, the user must ensure that these are also selected, as it is not done automatically.

References

1. Boddie, S., Thompson, J., Bainbridge, D., Witten, I.H.: Coping with very large digital collections using greenstone. In: Proceedings of the ECDL Workshop on Very Large Digital Libraries (September 2008)
2. Lagoze, C., Payette, S., Shin, E., Wilper, C.: Fedora: an architecture for complex objects and their relationships. International Journal on Digital Libraries 6(2), 124–138 (2006)
3. Nemeth, E., Snyder, G., Hein, T.R.: Linux Administration Handbook. Prentice-Hall, Englewood Cliffs (2007)
4. New Zealand Digital Library Project. Greenstone digital library software (last visited, July 2008), `http://www.greenstone.org`
5. Osborn, W., Fox, S.: Automatic and scheduled maintenance of digital library collections. In: Proceedings of the 2nd IEEE International Conference on Digital Information Management (ICDIM 2007) (October 2007)
6. Osborn, W., Fox, S., O'Shea, S.: A unified resource for post-secondary program information. In: Proceedings of the 2008 International Conference on Information Resources Management (Conf-IRM) (2008)
7. Schapira, E.: Pycron cron - great cron for windows (last visited, July 2008), `http://sourceforge.net/projects/pycron`
8. Tansley, R., Bass, M., Smith, M.: Dspace as an open archival information system: Status and future directions. In: Proceedings of the 10th European Conference on Digital Libraries (ECDL 2005) (September 2006)
9. Vixie, P.: Vixie cron for FreeBSD (last visited, July 2008), `http://www.freebsd.org/cgi/cvsweb.cgi/src/usr.sbin/cron/`
10. Witten, I.H.: Creating and customizing collections with the Greenstone Librarian Interface. In: Proceedings of the International Symposium on Digital Libraries and Knowledge Communities in Networked Information Society (March 2004)
11. Witten, I.H., Bainbridge, D.: How to Build a Digital Library. Morgan Kaufmann, San Francisco (2002)