CiteSeerX Crawler Interface and Implementations

Isaac Councill
August 23, 2007

Crawler Interface Specification

This section details the interface through which crawlers may suggest new content for inclusion in the CiteSeerX system. Any content acquisition service may suggest content using this interface (i.e., it is not restricted to crawlers per se). This interface describes how to pass content to the CiteSeerX ingestion system, which is responsible for deciding which content will be accepted into the repository and for extracting information from, transforming, and archiving the content that is acquired. This important point bears repeating: just because new content is submitted for ingestion over this interface does not mean that the content will be accepted for inclusion in the CiteSeerX repository. Please see the documentation regarding the ingestion system for further details. As long as the following guidelines are followed, any arbitrary content source may be bridged into the CiteSeerX ingestion system.

Crawl Data

Content to be submitted for ingestion will be handled at the level of individual files. That is, all files that are obtained will be downloaded and submitted individually for ingestion. In addition, all files must be accompanied by a separate XML file containing metadata regarding each file to be ingested. This metadata file must have a root element of "CrawlData" and include the following child elements:

- **crawlDate:** the time that the file was acquired, in a format parsable by java.util.Date
- **url:** the URL of the acquired file
- **parentUrl** (Optional): the url of the page that linked to the file that was acquired
- **contentType** (Optional): the type of content from HTTP headers
- **SHA1:** the SHA1 hash (in hexadecimal form) of the acquired file

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<CrawlData>
<crawlDate>Wed Aug 22 19:54:32 GMT 2007</crawlDate>
<url>http://www.personal.psu.edu/igc2/papers/petinot04api.pdf</url>
<parentUrl>http://www.personal.psu.edu/igc2/pubs.html</parentUrl>
<contentType>application/pdf</contentType>
<SHA1>cf3267bc68816891c2fe5f438e35e12c9375bf13</SHA1>
</CrawlData>
```

Exposing the Crawler Repository

A conforming crawler should download acquired content into a file system repository that can be exported to other machines via some networked file system. Examples may include exporting the crawler repository over NFS or the Global File System. This is required for the ingestion system to pull content from the crawler into a central repository for archiving, and for exposing the files for subsequent processing such as file filtration and metadata extraction. Each crawler should be configured with a unique crawler ID that allows the ingestion system to map messages from the crawler to the local mount point of its unique file system repository.

JMS Interface

The Java Messaging Service API is used to provide a decoupled interface that is both reliable and customizable for arbitrary crawler applications. Crawlers (or other content acquisition services) must be enabled to send messages over a JMS channel to the ingestion service, and may optionally be enabled to receive JMS messages regarding job submissions and send JMS reports regarding errors encountered during processing.

Channels

- **csx.ingestion.documentsToIngest** (Required): channel to be used for submitting new content (file references) to the CiteSeerX ingestion system.
- **csx.ingestion.newSubmissions** (Optional): channel to be used by other application components for submitting new jobs to the crawler.
- **csx.ingestion.statusUpdates** (Optional): channel to be used for sending messages to other application components regarding any errors encountered during the crawl process.

Messages

All messages in CiteSeerX are of type MapMessage. The key/value pairs that must be present in messages adhering to the crawler interface are discussed below.

New Content Message

This type of message is sent over the documentsToIngest channel and is targeted for the ingestion system. Messages of this type must include the following fields:

- **JID** - the job ID for this submission. This should be a unique string.
- **REPID** - the repository ID for the sending crawler. This should uniquely identify the crawler (and be mappable to the file system the crawler stores content on).
- **FILEPATH** - the path, relative to the root of the crawler repository, where the acquired file resides.
- **METAPATH** - the path, relative to the root of the crawler repository, where the metadata for the acquired file resides.
- **RESOURCETYPE** - a string representing the type of resource that has been acquired (e.g., "article" for a research paper or "hub" for a page that links to potentially interesting content.

URL Submission Message

This type of message is sent over the newSubmissions channel and is targeted for the crawler. If this message channel is used, a mechanism for creating a new crawl job based on the submission message must be present in the crawler implementation. Messages of this type must include the following fields:

- **JID** - the job ID for this submission. This should be a unique string.
- **URL** - One or more URLs to use as seeds for the crawl job. If multiple URLs are present, they must be separated by newlines.
- **DESC** - A textual description of this crawl job.

URL Status Message

This type of message is sent over the statusUpdates channel is targeted to external application components that are interested in such updates (e.g., listeners in MyCiteSeer web application). The purpose is to inform users of any errors that are encountered during the crawl process. Messages of this type must include the following fields:

- **TYPE** - should be set to 1 for this type of message.
- **JID** - the job ID for this submission. This should be a unique string specific to a crawl job (not necessarily the ID of the specific request that caused this content to be crawled) and should map to a submission job ID so that the update can be matched to an initial job.
- **URL** - the URL that this update refers to.
- **STATUS** - and integer that represents the status code of the request (e.g., 401, 404, etc.).

It is recommended to only use this message type for notification of error conditions in order to minimize the volume of messages created. Codes recommended to skip include 100, 101, 300, 302, 303, 304, and 307.

Conforming Crawler Implementations

Heritrix

Heritrix (http://crawler.archive.org/) is the Internet Archive's open-source web crawler project. Please see the Heritrix web site and documentation for a full description of the capabilities of this crawler. Heritrix has been developed with a modular framework than can be modified and extended to provide custom functionality without altering the Heritrix source code. This framework has been used to adapt the Heritrix crawler to conform to the CiteSeerX Crawler Interface specification. The instructions provided here are based on customizations to Heritrix 1.12.1, and may not be applicable to future major versions of the Heritrix software.

Customizations

JMS Interface

This is the core component for bridging Heritrix to other applications using JMS message channels. A JMSInterface class should be loaded as a servlet that initializes on startup in the Heritrix web.xml, which will initialize and manage connections to JMS resources.

Access to a Heritrix instance is necessary for submitting new crawl jobs via JMS. Although no Heritrix instance will be available at startup, the JMSInterface class polls the environment until a Heritrix is created. At that point, the messaging channels will be started and new submissions are possible.

Bridging the JMS Interface into Heritrix Management

A JobSubmitter class is provided that bridges JMS submissions into the Heritrix job submission system. This class creates CrawlJob instances based on SubmissionData objects and specified Heritrix job profile information, then passes the CrawlJobs into the Heritrix CrawlJobHandler for regular processing. In this way, jobs can be managed through the Heritrix web application just in the same manner as if the jobs where created using traditional methods.

Parent URL Annotation

A ParentURLAnnotationProcessor class is provided in order to track link relationships during the crawl process. After links are discovered in a CrawlURI and promoted to CandidateURIs by a LinkScoper, this class is used to augment all child CandidateURIs with the URL of the parent CrawlURI. This data is placed in the children with the key "parent-url", and is carried through the system as the children pass through their own processing cycles.

This class is quite useful if you want to know where content came from without having to do batch post-processing on the link graph after crawls.

In configuration, this processor must be placed after a LinkScoper and before the FrontierScheduler.

Crawl Status Updates

A JMSCrawlStatusUpdateProcessor is provided, which calls JMSInterface to create JMS messages indicating any problems encountered during the crawl. Messages will only be sent for jobs whose names start with CSXConstants.USER_SUBMISSION_PREFIX. There are several failure codes that will be ignored, specified in the static final ignoreCodes array within this class. These codes are -50, 1, 100, 101, 300, 301, 302, 303, 304, 307. All other will be reported.

This processor should be configured as a post processor so that it is called after an attempt is made to download CrawlURI resources.

Download Status Updates

A JMSDownloadNotificationProcessor is provided for notifying the ingestion system that new content is ready for processing. This class serves dual purposes: first, this class is responsible for creating metadata files for downloaded content. Second, this class will call JMSInterface to create a message indicating that the files are ready for ingestion, along with file path information.

Important: this class assumes that MirrorWriterProcessor is used to download files. This processor should be placed directly after the MirrorWriterProcessor in profile configuration.

Downloading Hub Documents

A CSXHubFilter module is provided for identifying pages that link to potentially interesting content, or "hub" URIs, where a hub is defined as any document containing links to documents of interest (such as PDF, PS, etc.). CSXHubFilter will respond with ACCEPT if any of the out links discovered from the URI being filtered match a regular expression specified in configuration. The pattern match should be set in CSX ConfigurationManager style, with the path edu.psu.citeseerx.heritrix.jms.hubLinkIndicator. This should be used as a condition for downloading content in the MirrorWriterProcessor.

Quick Install from CiteSeerX Distribution

Copy the provided Heritrix distribution to the location where you would like it to reside. Edit conf/heritrix.properties if needed to specify the port on which the admin web application should run (heritrix.cmdline.port property). Then edit conf/csx-heritrix.xml. Make sure that you specify a unique repositoryID in the heritrix.jms configuration path. Each crawler that interfaces with CiteSeerX should have a unique ID, not just Heritrix instances. It will also be necessary to change the messaging.jmsProvider.url property to point to your ActiveMQ deployment (e.g., <url>tcp://activemq.loc.net:61616</url>).

The crawler repository root for the customized Heritrix is the jobs/ directory under the Heritrix installation directory. Make sure this directory is available to ingestion services as described the CSX Crawler Interface specification above and then configure ingestion services with the appropriate location, mapped from the value you specified as this Heritrix instance's repository ID.

Now you are ready to begin using Heritrix with CiteSeerX.

Installation from Scratch

A full copy of the customized Heritrix installation should be provided along with the CiteSeerX distribution. If you are using the provided package, please use the above Quick Install instructions. However, if you are installing from scratch (for instance, if you are upgrading your Heritrix base application), the instructions below should enable you to fully customize a Heritrix installation for use with CiteSeerX.

See the Heritrix documentation for initial setup procedures. Follow the instructions there before moving on to complete modifications for the CiteSeerX crawling environment.

Required Libraries

In addition to the libraries that come with Heritrix by default, the following jar files should be included in the Heritrix lib/ directory:

From CiteSeerX:
csx-heritrix (include messaging, messaging.messages, heritrix.jms, and utility packages)

From the ActiveMQ library:
activemq-core
backport-util-concurrent

geronimo-j2ee-management_XX_spec
geronimo-jms_XX_spec

Other dependencies:
javax.jms
commons-configuration
commons-io
jdom

Extra Heritrix Configuration

Registering the New Modules

Once the required libraries have been added, the new functionality provided in csx-heritrix needs to be made visible to Heritrix. First, edit the module configuration files to register new modules with the system. Unfortunately, this configuration files reside within the main heritrix jar file in the Heritrix installation directory. Copy this jar into an empty directory in a safe location, then extract the jar contents with the command:

```
jar xf heritrix-XX.jar
```

Edit modules/Processor.options, adding the following lines to the end of the file:

```
edu.psu.citeseerx.heritrix.jms.JMSCrawlStatusProcessor|JMSCrawlStatusPr
ocessor
edu.psu.citeseerx.heritrix.jms.JMSDownloadNotificationProcessor|JMSDown
loadNotificationProcessor
edu.psu.citeseerx.heritrix.jms.ParentURLAnnotationProcessor|ParentURLAn
notationProcessor
```

Then edit modules/DecideRule.options, adding the following line to the end of the file:

```
edu.psu.citeseerx.heritrix.jms.CSXHubFilter
```

Now, change back to the directory in which you extracted the contents of the original jar, delete the original jar, and then create a new one with the same name with the command:

```
jar cf heritrix-XX.jar *
```

Backup the original jar in the Heritrix installation directory, then replace it with the jar that you just created.

Creating a Submission Profile

See the Heritrix documentation for creating new job profiles. For convenience, instructions are provided here for copying the provided CSX_User_Submission profile into your Heritrix directory. In the conf/profiles directory under the Heritrix installation directory:

```
mkdir CSX_User_Submission
cp /path/to/provided/submission_profile.xml
CSX_User_Submission/order.xml
touch CSX_User_Submission/seeds.txt
```

A reference submission profile order.xml file is provided at the end of the Heritrix section of this document. The intent of the user submission crawl profile is to create jobs that will crawl only to depth 1 from seeds.

Adding the JMS Message Listener

One final bit of Heritrix configuration that is needed requires editing the web.xml file in the admin webapp. From the webapps directory in the Heritrix installation directory, extract admin.war:

```
mkdir admin
cp admin.war admin/
cd admin
jar xf admin.war
```

Then edit WEB-INF/web.xml, adding the following servlet definition to the list of servlets already defined by Heritrix:

```
<servlet>
    <servlet-name>edu.psu.citeseerx.heritrix.jms.JMSInterface</servlet-
name>
    <servlet-
class>edu.psu.citeseerx.heritrix.jms.JMSInterface</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
```

Now, either create a new war file using the modified admin directory and replace the original admin.war, or simply delete the original admin.war.

CSX Configuration

Before the crawler can be started with these modifications, an extra conf file will be needed for the custom components that we have added. This file will follow the CSX ConfigurationManager style. The name and location of this file is arbitrary, but generally a file called csx-heritrix.xml is placed in the Heritrix conf/ directory. This configuration file will be used to manage the JMS message interface as well as to provide additional configuration for discovering hub documents.

The following XML illustrates a properly formatted csx-heritrix.xml file. Under the heritrix.jms path, configuration is needed for identifying the names of the JMS message channels that are needed. A "repositoryID" label is needed that uniquely identifies this crawler instance. The repository ID is used by ingestion services to locate files that are ready for ingestion. The "submissionProfile" attribute specifies the name of the Heritrix job profile to use when creating new jobs based on JMS submissions. This is the profile that is supplied with the CiteSeerX distribution (or that you have previously configured).

Additionally, a "hubLinkIndicator" attribute is needed that specifies a regular expression to match against out links of crawled pages. This regular expression should match URL strings that point to potentially interesting content (such as PDF documents, in the example supplied here). For all crawled pages, if any out links match this expression, the linking page is marked as a hub and will be acquired by ingestion.

The csx-heritrix.xml file must also provide explicit configuration for the JMS message service, as described in CSX Messaging documentation. At the minimum, you will need to change the value of jmsProvider.url to point to your ActiveMQ deployment. The message channels configured here must match the channel names under the heritrix.jms configuration path. newSubmissionChannel should be conofigured with the role of "consumer" and the other channels should be configured with the role of "producer". All channels should be queues.

Finally, to make the csx-heritrix.xml configuration visible to the application, the Java runtime should be started with the usual ConfigurationManager definitions: "-DCSX_HOME=/path/to/heritrix -DCSX_CONF=conf/csx-heritrix.xml". This can be achieved by adding these configuration directives to your JAVA_OPTS environment variable by either setting the JAVA_OPTS variable in your shell environment or by explicitly appending the directives to JAVA_OPTS at the beginning of the heritrix startup script. If you are using the heritrix supplied in the CiteSeerX distribution, the heritrix startup script should already be appropriately modified.

```
<?xml version="1.0"?>
<serviceConfiguration>
<edu><psu><citeseerx>

<utility>
  <ConfigurationManager>
    <autoSave>false</autoSave>
  </ConfigurationManager>
</utility>

<heritrix>

  <jms>
```

```
<newSubmissionsChannel>csx.ingestion.newSubmissions</newSubmissionsChan
nel>

<statusUpdateChannel>csx.ingestion.statusUpdates</statusUpdateChannel>

<ingestionChannel>csx.ingestion.documentsToIngest</ingestionChannel>
    <repositoryID>heritrix1</repositoryID>
    <submissionProfile>CSX_User_Submission</submissionProfile>
    <hubLinkIndicator>.*(\.(pdf))(\.(g?z))?$</hubLinkIndicator>
  </jms>

</heritrix>

<messaging>

  <jmsProvider>
    <url>tcp://ACTIVEMQ_HOST:61616</url>
    <clientID>heritrix1</clientID>

    <queue>
      <name>csx.ingestion.newSubmissions</name>
      <role>consumer</role>
      <acknowledgeMode>CLIENT_ACKNOWLEDGE</acknowledgeMode>
    </queue>

    <queue>
      <name>csx.ingestion.statusUpdates</name>
      <role>producer</role>
      <acknowledgeMode>CLIENT_ACKNOWLEDGE</acknowledgeMode>
    </queue>

    <queue>
      <name>csx.ingestion.documentsToIngest</name>
      <role>producer</role>
      <acknowledgeMode>CLIENT_ACKNOWLEDGE</acknowledgeMode>
    </queue>

  </jmsProvider>

</messaging>

</citeseerx></psu></edu>
</serviceConfiguration>
```

Usage

After configuring your customized Heritrix application, the crawler repository must be made available to ingestion services according to the CSX Crawler Interface Specification. The root of the customized Heritrix repository is the jobs/ directory under the Heritrix installation directory. Ingestion services should be configured to map this repository location from the repository ID of this Heritrix instance.

At this point, to start Heritrix you simply need to execute the Heritix startup script. You may use Heritrix as prescribed in the Heritrix user manual (from the web application) or through the CiteSeerX JMS interface. Messaging will commence in adherence to the CSX Crawler Interface specification described at the beginning of this document.

To create custom jobs and profiles that will still adhere to the CSX Crawler Interface, new jobs and profiles can be created based on the CSX_User_Submission profile. Any modifications to the crawl scope (such as using different scope modules or increasing crawl depth) should be safe, but care should be taken not to modify writers or post-processors without a firm understanding of the consequences.

User Submission Profile order.xml file

```
<?xml version="1.0" encoding="UTF-8"?><crawl-order
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="heritrix_settings.xsd">
  <meta>
    <name>CSX_User_Submission</name>
    <description>Standard profile for indexing user
submissions</description>
    <operator>Admin</operator>
    <organization></organization>
    <audience></audience>
    <date>20070822182732</date>
  </meta>
  <controller>
    <string name="settings-directory">settings</string>
    <string name="disk-path"></string>
    <string name="logs-path">logs</string>
    <string name="checkpoints-path">checkpoints</string>
    <string name="state-path">state</string>
    <string name="scratch-path">scratch</string>
    <long name="max-bytes-download">0</long>
    <long name="max-document-download">0</long>
    <long name="max-time-sec">0</long>
    <integer name="max-toe-threads">50</integer>
    <integer name="recorder-out-buffer-bytes">4096</integer>
    <integer name="recorder-in-buffer-bytes">65536</integer>
    <integer name="bdb-cache-percent">0</integer>
    <newObject name="scope"
class="org.archive.crawler.deciderules.DecidingScope">
      <boolean name="enabled">true</boolean>
      <string name="seedsfile">seeds.txt</string>
      <boolean name="reread-seeds-on-config">true</boolean>
      <newObject name="decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
        <map name="rules">
          <newObject name="AcceptByDefault"
class="org.archive.crawler.deciderules.AcceptDecideRule">
          </newObject>
          <newObject name="RejectIfTooManyHops"
class="org.archive.crawler.deciderules.TooManyHopsDecideRule">
            <integer name="max-hops">1</integer>
          </newObject>
          <newObject name="AcceptIfPrerequisite"
class="org.archive.crawler.deciderules.PrerequisiteAcceptDecideRule">
          </newObject>
          <newObject name="AcceptIfTranscluded"
class="org.archive.crawler.deciderules.TransclusionDecideRule">
            <integer name="max-trans-hops">3</integer>
            <integer name="max-speculative-hops">0</integer>
          </newObject>
        </map>
      </newObject>
    </newObject>
    <map name="http-headers">
      <string name="user-agent">Mozilla/5.0 (compatible;
CiteSeerBot/1.12.1 +http://citeseerx.ist.psu.edu/bot.html)</string>
      <string name="from">citeseerx-bot@ist.psu.edu</string>
    </map>
    <newObject name="robots-honoring-policy"
class="org.archive.crawler.datamodel.RobotsHonoringPolicy">
      <string name="type">classic</string>
      <boolean name="masquerade">false</boolean>
      <text name="custom-robots"></text>
      <stringList name="user-agents">
      </stringList>
    </newObject>
    <newObject name="frontier"
class="org.archive.crawler.frontier.BdbFrontier">
      <float name="delay-factor">4.0</float>
      <integer name="max-delay-ms">20000</integer>
      <integer name="min-delay-ms">2000</integer>
      <integer name="max-retries">3</integer>
      <long name="retry-delay-seconds">10</long>
      <integer name="preference-embed-hops">1</integer>
      <integer name="total-bandwidth-usage-KB-sec">0</integer>
      <integer name="max-per-host-bandwidth-usage-KB-sec">0</integer>
      <string name="queue-assignment-
policy">org.archive.crawler.frontier.HostnameQueueAssignmentPolicy</str
ing>
      <string name="force-queue-assignment"></string>
      <boolean name="pause-at-start">false</boolean>
      <boolean name="pause-at-finish">false</boolean>
      <boolean name="source-tag-seeds">false</boolean>
      <boolean name="recovery-log-enabled">true</boolean>
      <boolean name="hold-queues">true</boolean>
      <integer name="balance-replenish-amount">3000</integer>
      <integer name="error-penalty-amount">100</integer>
      <long name="queue-total-budget">-1</long>
      <string name="cost-
policy">org.archive.crawler.frontier.ZeroCostAssignmentPolicy</string>
      <long name="snooze-deactivate-ms">300000</long>
      <integer name="target-ready-backlog">50</integer>
      <string name="uri-included-
structure">org.archive.crawler.util.BdbUriUniqFilter</string>
    </newObject>
    <map name="uri-canonicalization-rules">
      <newObject name="Lowercase"
class="org.archive.crawler.url.canonicalize.LowercaseRule">
        <boolean name="enabled">true</boolean>
      </newObject>
```

```
   <newObject name="Userinfo"
class="org.archive.crawler.url.canonicalize.StripUserinfoRule">
    <boolean name="enabled">true</boolean>
   </newObject>
   <newObject name="WWW[0-9]*"
class="org.archive.crawler.url.canonicalize.StripWWWNRule">
    <boolean name="enabled">true</boolean>
   </newObject>
   <newObject name="SessionIDs"
class="org.archive.crawler.url.canonicalize.StripSessionIDs">
    <boolean name="enabled">true</boolean>
   </newObject>
   <newObject name="SessionCFIDs"
class="org.archive.crawler.url.canonicalize.StripSessionCFIDs">
    <boolean name="enabled">true</boolean>
   </newObject>
   <newObject name="QueryStrPrefix"
class="org.archive.crawler.url.canonicalize.FixupQueryStr">
    <boolean name="enabled">true</boolean>
   </newObject>
  </map>
  <map name="pre-fetch-processors">
   <newObject name="Preselector"
class="org.archive.crawler.prefetch.Preselector">
    <boolean name="enabled">true</boolean>
    <newObject name="Preselector#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
     <map name="rules">
     </map>
    </newObject>
    <boolean name="override-logger">false</boolean>
    <boolean name="recheck-scope">true</boolean>
    <boolean name="block-all">false</boolean>
    <string name="block-by-regexp"></string>
    <string name="allow-by-regexp"></string>
   </newObject>
   <newObject name="Preprocessor"
class="org.archive.crawler.prefetch.PreconditionEnforcer">
    <boolean name="enabled">true</boolean>
    <newObject name="Preprocessor#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
     <map name="rules">
     </map>
    </newObject>
    <integer name="ip-validity-duration-seconds">21600</integer>
    <integer name="robot-validity-duration-seconds">86400</integer>
    <boolean name="calculate-robots-only">false</boolean>
   </newObject>
  </map>
  <map name="fetch-processors">
   <newObject name="DNS"
class="org.archive.crawler.fetcher.FetchDNS">
    <boolean name="enabled">true</boolean>
    <newObject name="DNS#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
     <map name="rules">
     </map>
    </newObject>
```

```
        <boolean name="accept-non-dns-resolves">false</boolean>
        <boolean name="digest-content">true</boolean>
        <string name="digest-algorithm">sha1</string>
      </newObject>
      <newObject name="HTTP"
class="org.archive.crawler.fetcher.FetchHTTP">
        <boolean name="enabled">true</boolean>
        <newObject name="HTTP#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
        <newObject name="midfetch-decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
        <integer name="timeout-seconds">60</integer>
        <integer name="sotimeout-ms">20000</integer>
        <integer name="fetch-bandwidth">0</integer>
        <long name="max-length-bytes">0</long>
        <boolean name="ignore-cookies">false</boolean>
        <boolean name="use-bdb-for-cookies">true</boolean>
        <string name="load-cookies-from-file"></string>
        <string name="save-cookies-to-file"></string>
        <string name="trust-level">open</string>
        <stringList name="accept-headers">
        </stringList>
        <string name="http-proxy-host"></string>
        <string name="http-proxy-port"></string>
        <string name="default-encoding">ISO-8859-1</string>
        <boolean name="digest-content">true</boolean>
        <string name="digest-algorithm">sha1</string>
        <boolean name="send-if-modified-since">true</boolean>
        <boolean name="send-if-none-match">true</boolean>
        <boolean name="send-connection-close">true</boolean>
        <boolean name="send-referer">true</boolean>
        <boolean name="send-range">false</boolean>
        <string name="bind-address"></string>
      </newObject>
    </map>
    <map name="extract-processors">
      <newObject name="ExtractorHTTP"
class="org.archive.crawler.extractor.ExtractorHTTP">
        <boolean name="enabled">true</boolean>
        <newObject name="ExtractorHTTP#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
      </newObject>
      <newObject name="ExtractorHTML"
class="org.archive.crawler.extractor.ExtractorHTML">
        <boolean name="enabled">true</boolean>
        <newObject name="ExtractorHTML#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
        <boolean name="extract-javascript">true</boolean>
        <boolean name="treat-frames-as-embed-links">true</boolean>
        <boolean name="ignore-form-action-urls">false</boolean>
        <boolean name="overly-eager-link-detection">true</boolean>
        <boolean name="ignore-unexpected-html">true</boolean>
      </newObject>
      <newObject name="ExtractorCSS"
class="org.archive.crawler.extractor.ExtractorCSS">
        <boolean name="enabled">true</boolean>
        <newObject name="ExtractorCSS#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
      </newObject>
      <newObject name="ExtractorJS"
class="org.archive.crawler.extractor.ExtractorJS">
        <boolean name="enabled">true</boolean>
        <newObject name="ExtractorJS#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
      </newObject>
      <newObject name="ExtractorSWF"
class="org.archive.crawler.extractor.ExtractorSWF">
        <boolean name="enabled">true</boolean>
        <newObject name="ExtractorSWF#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
      </newObject>
    </map>
    <map name="write-processors">
      <newObject name="MirrorWriter"
class="org.archive.crawler.writer.MirrorWriterProcessor">
        <boolean name="enabled">true</boolean>
        <newObject name="MirrorWriter#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
            <newObject name="rejectByDefault"
class="org.archive.crawler.deciderules.RejectDecideRule">
            </newObject>
            <newObject name="acceptByFileType"
class="org.archive.crawler.deciderules.MatchesFilePatternDecideRule">
              <string name="decision">ACCEPT</string>
              <string name="use-preset-pattern">Miscellaneous</string>
              <string name="regexp">.*(\.(pdf))(\.(g?z))?</string>
            </newObject>
            <newObject name="acceptIfHub"
class="edu.psu.citeseerx.heritrix.jms.CSXHubFilter">
            </newObject>
          </map>
        </newObject>
        <boolean name="case-sensitive">false</boolean>
        <stringList name="character-map">
        </stringList>
        <stringList name="content-type-map">
        </stringList>
        <string name="directory-file">index.html</string>
        <string name="dot-begin">%2E</string>
        <string name="dot-end">.</string>
        <stringList name="host-map">
        </stringList>
        <boolean name="host-directory">true</boolean>
        <string name="path">mirror</string>
        <integer name="max-path-length">1023</integer>
        <integer name="max-segment-length">255</integer>
        <boolean name="port-directory">false</boolean>
        <boolean name="suffix-at-end">true</boolean>
        <string name="too-long-directory">LONG</string>
        <stringList name="underscore-set">
        </stringList>
      </newObject>
      <newObject name="JMSDownloadNotificationProcessor"
class="edu.psu.citeseerx.heritrix.jms.JMSDownloadNotificationProcessor"
>
        <boolean name="enabled">true</boolean>
        <newObject name="JMSDownloadNotificationProcessor#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
      </newObject>
    </map>
    <map name="post-processors">
      <newObject name="Updater"
class="org.archive.crawler.postprocessor.CrawlStateUpdater">
        <boolean name="enabled">true</boolean>
        <newObject name="Updater#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
      </newObject>
      <newObject name="LinksScoper"
class="org.archive.crawler.postprocessor.LinksScoper">
        <boolean name="enabled">true</boolean>
        <newObject name="LinksScoper#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
        <boolean name="override-logger">false</boolean>
        <boolean name="seed-redirects-new-seed">true</boolean>
        <integer name="preference-depth-hops">-1</integer>
        <newObject name="scope-rejected-url-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
      </newObject>
      <newObject name="ParentURLAnnotationProcessor"
class="edu.psu.citeseerx.heritrix.jms.ParentURLAnnotationProcessor">
```

```
        <boolean name="enabled">true</boolean>
        <newObject name="ParentURLAnnotationProcessor#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
      </newObject>
      <newObject name="Scheduler"
class="org.archive.crawler.postprocessor.FrontierScheduler">
        <boolean name="enabled">true</boolean>
        <newObject name="Scheduler#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
      </newObject>
      <newObject name="JMSCrawlStatusProcessor"
class="edu.psu.citeseerx.heritrix.jms.JMSCrawlStatusProcessor">
        <boolean name="enabled">true</boolean>
        <newObject name="JMSCrawlStatusProcessor#decide-rules"
class="org.archive.crawler.deciderules.DecideRuleSequence">
          <map name="rules">
          </map>
        </newObject>
      </newObject>
    </map>
    <map name="loggers">
      <newObject name="crawl-statistics"
class="org.archive.crawler.admin.StatisticsTracker">
        <integer name="interval-seconds">20</integer>
      </newObject>
    </map>
    <string name="recover-path"></string>
    <boolean name="checkpoint-copy-bdbje-logs">true</boolean>
    <boolean name="recover-retain-failures">false</boolean>
    <newObject name="credential-store"
class="org.archive.crawler.datamodel.CredentialStore">
      <map name="credentials">
      </map>
    </newObject>
  </controller>
</crawl-order>
```