NetarchiveSuite 7.x Release Notes

Download Links for NetarchiveSuite 7.3:

- Download NetarchiveSuite
- Download Heritrix 3 Bundle (required)
- Javadoc
- docker-compose assembly
- Manuals
- OpenWayback Overlay warfile

7.0 Release Date: 2021-03-19

7.1 Release Date: 2021-07-06

7.2 Release Date: 2021-08-19

7.3 Release Date: 2022-01-31

Highlights in 7.4

The CrawlRSS module has been updated to be compatible with the current version of heritrix. See documentation - RSS Harvests .

Highlights in 7.3

1. Fixed a bug in the bitmagasin logic used by WaybackIndexer to fetch all filenames
2. Made fetching of hadoop results from hdfs pipe directly to disk, thereby avoiding potential OutOfMemory issue
3. Refactored the hadoop version of the CDX-indexing workflow
4. Added a number of upstream fixed to heritrix, including one to fix unwanted behaviour when a url redirects to a top-level-domain
5. Added two new settings parameters to make FileResolver more robust in the event of server instability.

```
    /**
     * Number of retries for fileresolver if an empty result is obtained
(0 = try only once). default 3.
     */
    public static String FILE_RESOLVER_RETRIES = "settings.common.
fileResolver.retries";

    /**
     * Seconds to wait between retries. default 5.
     */
    public static String FILE_RESOLVER_RETRY_WAIT = "settings.common.
fileResolver.retrywaitSeconds";
```

Highlights in 7.2

1. Fixed **NAS-2868** - Getting issue details... STATUS and **NAS-2864** - Getting issue details... STATUS so that the version of Heritrix reported in all archive and metadata files is correct and consistent.
2. Included all Heritrix patches up to the 2021-08-03 Interim Release , as well as a number of even more recent minor bugfixes . This upgrade includes as a major new feature the ExtractorChrome module which enables browser-based harvesting from directly within the Heritrix extractor chain. To enable browser-based harvesting, add a bean like this

```
  <bean id="extractorChrome" class="org.archive.modules.extractor.
ExtractorChrome">
      <property name="executable" value="/usr/bin/google-chrome"/>
  </bean>
```

to the FetchChain of your crawler-beans before the ExtractorHTTP element. Then make sure your harvest job runs on a machine where chrome (or chromium) is available at the specified executable path. Here you can use NetarchiveSuite's existing harvest-channel mappings functionality if only some of your harvesting machines are to be used for browser-based harvesting. Content harvested by the browser can be identified in the crawl log as they will be annotated "browser".

3. ExtractorSitemap has been modified with two optional properties:

```
 <bean id="extractorSitemap" class="org.archive.modules.extractor.
ExtractorSitemap">
      <property name="urlPattern" value=".*sitemap.*\.xml.*"/>
      <property name="enableLenientExtraction" value="true" />
 </bean>
```

if "urlPattern" is set then any url matching this pattern is assumed to be a sitemap. Otherwise ExtractorSitemap reverts to its default functionality whereby it checks the mime-type of every url and then sniffs the start of any xml url to see if it looks like a sitemap. If "enableLenientExtraction" is set to true then every url found in the sitemap will be extracted. Otherwise the extractor will omit any urls which do not obey the scoping rules defined in the site map specification.

Highlights in 7.1

1. Fixed (after many years) **NAS-2870** - Getting issue details... STATUS whereby all generated revisit-records had badly formatted WARC-Payload-Digest fields and were therefore invalid according to the Warc standard.
2. Added 3 new link extractors (from the British Library) to heritrix :
   - org.archive.modules.extractor.ExtractorRobotsTxt
   - org.archive.modules.extractor.ExtractorSitemap
   - org.archive.modules.extractor.ExtractorJson
     Note that ExtractorSitemap deviates slightly in functionality from the British Library version in that it is considerably more lenient in both what it identifies as a sitemap and what Urls it accepts in sitemaps.
3. Added caching of crawl logs and metadata-indexes when hadoop is used for processing
   a. The new caching functionality for crawl logs and metadata indexes stores data in a directory specified by the setting

      ```
      settings.common.webinterface.metadata_cache_dir
      ```

      whose default value is "metadata_cache" (relative to the current working directory where the GUIApplication is started). At present there is no automatic cleaning of this directory.
4. Added retry functionality to improve the robustness of the WarcRecordClient
5. Fixed a bug whereby files uploaded from a harvester were not being deleted when the Bitrepository backend is in use
6. Added retry-handling to Bitrepository uploads via two new settings keys under settings.common.arcrepositoryClient.bitrepository

   ```
   <store_retries>3</store_retries>
   <retryWaitSeconds>1800</retryWaitSeconds>
   ```

7. Added parameters to manage memory and core usage in hadoop mapper-only jobs

   ```
   settings.common.hadoop.mapred.mapMemoryMb
   settings.common.hadoop.mapred.mapMemoryCores
   ```

8. Added support for uberized jobs, optimised for small tasks in hadoop, via

```
settings.common.hadoop.mapred.enableUbertask
```

9. Added hdfs-caching functionality to hadoop jobs. When this feature is enabled, any local files passed as input to the hadoop job are first copied into hdfs and cached for future use. This should create savings when the same file is processed multiple times, as is often the case for metadata files. This functionality is controlled by the following parameters

   ```
   settings.common.hadoop.mapred.hdfsCacheEnabled
   settings.common.hadoop.mapred.hdfsCacheDir
   settings.common.hadoop.mapred.hdfsCacheDays
   ```

   Note that if the cache is enabled but the "hdsfCacheDays" parameter is set to zero then files are still copied into hdfs before processing but are deleted and recopied each time they are used. This can be useful for benchmarking.
10. Added parameters to determine which hadoop mapreduce job queue is used for different jobs. Currently two possibilities are allowed for:

    ```
    settings.common.hadoop.mapred.queue.batch
    settings.common.hadoop.mapred.queue.interactive
    ```

    "Interactive" is used for jobs started by GUI operations and "batch" for all other jobs. By assigning these to different hadoop queues, each with a non-zero minimum quota, one can ensure that interactive jobs do not have to wait indefinitely while batch jobs are being processed.
11. Improved the performance of the GUI functionality associated with the button "Browse only relevant crawl-log lines for this domain".

Highlights in 7.0

NetarchiveSuite 7.0 introduces an entirely new backend storage and mass-processing implementation based on software from bitrepository.org and hadoop. The new functionality is enabled by defining the following key in the settings file for all applications:

```
<settings>
   <common>
      <arcrepositoryClient>
         <class>dk.netarkivet.archive.arcrepository.distribute.
BitmagArcRepositoryClient</class>
```

and additionally

```
<settings>
   <common>
      <useBitmagHadoopBackend>true</useBitmagHadoopBackend>
```

The older arcrepositoryClient implementation `dk.netarkivet.archive.arcrepository.distribute.JMSArcRepositoryClient` will be deprecated in future releases. (The developers are unaware of any other organisations currently using the older client, but please contact us if you still rely on it.)

The new architecture introduces many new keys and external configuration files. There is therefore a separate Guide To Configuring the NetarchiveSuite 7.0 Backend.

Upgrading From Previous NetarchiveSuite Releases

For those using either JMSArcRepositoryClient or LocalArcRepositoryClient there should be no special requirements to upgrade.

Issues Resolved in Release 7.0