

## JHoNas final report

Foster WARC usage in scalable Web Archiving workflows using Jhove2 and  
NetarchiveSuite

### Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Milestones</b>	<b>1</b>
<b>3</b>	<b>Released software</b>	<b>4</b>
3.1	Expected . . . . .	4
3.2	Additional . . . . .	4
3.3	Other projects using JWAT . . . . .	5
<b>4</b>	<b>JHOVE2 development</b>	<b>6</b>
<b>5</b>	<b>NetarchiveSuite(NAS) development</b>	<b>7</b>
<b>6</b>	<b>ARC to WARC Migration</b>	<b>8</b>
6.1	Harvesting . . . . .	8
6.2	The Archive . . . . .	8
<b>7</b>	<b>Experience learned</b>	<b>9</b>
7.1	WARC . . . . .	9
7.2	Development 'ecosystem' . . . . .	10
<b>8</b>	<b>Final remarks</b>	<b>11</b>
<b>A</b>	<b>Proposal: Foster WARC usage in scalable Web Archiving workflows using Jhove2 and NetarchiveSuite</b>	<b>12</b>
<b>B</b>	<b>Status update 11 Apr. 2012</b>	<b>16</b>
<b>C</b>	<b>Status update 21 Apr. 2012</b>	<b>19</b>
<b>D</b>	<b>Status update 26 Jun. 2012</b>	<b>24</b>
<b>E</b>	<b>Status update 1 Augr. 2012</b>	<b>30</b>
<b>F</b>	<b>Status update 13 Sep. 2012</b>	<b>36</b>
<b>G</b>	<b>Status update 27 Sep. 2012</b>	<b>41</b>
<b>H</b>	<b>Status update 17 Apr. 2013</b>	<b>47</b>
<b>I</b>	<b>NAS workshop agenda and outcome (2012-04-02)</b>	<b>50</b>
<b>J</b>	<b>NAS 3.21.0 release notes (Developer release)</b>	<b>56</b>

<b>K</b>	<b>JHOVE2 2.1.0 release notes</b>	<b>59</b>
<b>L</b>	<b>JHOVE2 WARC module specifications</b>	<b>67</b>
<b>M</b>	<b>JHOVE2 ARC module specifications</b>	<b>82</b>
<b>N</b>	<b>JHOVE2 GZip module specifications</b>	<b>94</b>

## 1 Introduction

This report is the documented result of the work done in connection with the JHoNas project. The original title of the project proposal being: **Foster WARC usage in scalable Web Archiving workflows using Jhove2 and NetarchiveSuite**. The original document can be found in appendix A.

The overall goal of the project was to enhance existing tools in order to ease the adaptation of WARC as the preferred archiving format for digital preservation.

In order to accomplish this, two applications were chosen which would cover the entire digital preservation workflow.

The two applications chosen were:

- JHove2<sup>1</sup>
- NetarchiveSuite<sup>2</sup>

## 2 Milestones

Each milestone includes a description of relevant activities and outcomes. Project status updates can be found in appendix B to H

### M1 - Technical specification of WARC module for JHOVE2 (jan-12)

The technical specification for WARC was more or less based on the specifications that had been done earlier for ARC/GZip by BnF. The initial work on the specification was done in Paris at the yearly NetarchiveSuite workshop (held late November 2011). However the specifications could not be submitted for approval before the WARC module was stable enough for all properties to have been defined.

The first draft was submitted but not approved since it was lacking a description of how validation was performed. An amended version including additional information was drafted and approved the following month. The specification was also delayed by the completion of the WARC validation implementation.

The technical specifications for ARC/GZip were also updated according to their new properties and also to include information about how the validation was performed.

The specifications can be downloaded from the JHove2 website<sup>3</sup>.

### M2 - Prototype Code release of JHOVE2-modules (mar-12)

In addition to the implementation of a new WARC module it was also expected that the existing ARC/GZip modules could be migrated to run on the latest JHove2 code base. However

---

<sup>1</sup><https://bitbucket.org/jhove2/main/wiki/Home>

<sup>2</sup><https://sourceforge.net/projects/netarchive-suite/>

<sup>3</sup><https://bitbucket.org/jhove2/main/wiki/Modules>

the existing GZip/ARC modules were not compatible with the new JHove2 code base after some significant changes to the JHove2 internals. Instead the modules were constructed from scratch based on existing JHove2 modules and the old ARC/GZip modules.

Since debugging and unit testing was not the fastest thing in an application like JHove2, we decided that it would be easier to develop the WARC, ARC and GZip code as a separate project. The JHove2 modules would then use the third party library JWAT<sup>4</sup> for actual validation.

The GZip and ARC code was partially rewritten because it contained buggy validation.

### **M3 - Workshop in Copenhagen on WARC/NAS specifications (apr-12)**

A small workshop was held in Copenhagen to discuss how to use WARC in NetarchiveSuite and follow up on JHove2 development.

The main focus of the discussions was which records and headers should be written in the metadata produced by NetarchiveSuite and data files produced by Heritrix.

Changing how Heritrix writes its data files, later proved to be too complicated to accomplish in the scope of the JHoNas project.

See appendix I for agenda and outcome. Also available from the NAS wiki<sup>5</sup>.

### **M4 - Progress report at IIPC GA 2012, Washington D.C (maj-12)**

A short update was given on the first day. Also a working prototype of the WARC, ARC and GZip modules were demonstrated in a PWG workshop. Both presentations should be available on the IIPC website.

### **M5 - Developer Release of NetarchiveSuite with WARC-support (aug-12)**

WARC support in NAS was started in the beginning of 2012 but not prioritized before mid 2012 because of increasing JHove2 module work.

However work on NAS was coordinated to fit the general release schedule.

NetarchiveSuite 3.21.0<sup>6</sup> was released on 5.9.2012 after a standard release test. Release notes can be found in appendix J.

### **M6 - Final Code release of JHOVE2-modules (sep-12)**

A release of JHOVE2 including the GZip, ARC and WARC modules was scheduled near the end of the one year project period. Attempts were made to initiate the preparation of the release some months earlier and a tentative release date was set for early September. Funding for the JHOVE2 project itself ended with the 2.0.0 release and since funding is generally tight

---

<sup>4</sup><http://jwat.org/>

<sup>5</sup><https://sbforge.org/display/NAS/NAS+Warc+workshop>

<sup>6</sup><https://sbforge.org/display/NAS/NetarchiveSuite+3.21.0+Release+Notes>

everywhere, maintenance of the project is subject to when ever the involved partner have some free time from their otherwise normal schedule. In the end people found time in their busy schedule to formalize rules for new contributors, fix some outstanding bugs which would be nice to include in the same release and also write down the process of preparing a release.

The official release and releases note are available from the JHOVE2 website<sup>7</sup>. Release notes can also be found in appendix K.

### **M7 - Workshop in Aarhus on WARC/NAS tests (sep-12)**

Prior to the stable release of NAS with WARC-support a workshop<sup>8</sup> was held in Aarhus to follow up on JHove2 test results and to discuss the last WARC changes for NAS.

Besides thorough testing of JHOVE2 at KB, BnF also ran it through their own comprehensive tests.

### **M8 - Stable Release of NetarchiveSuite with WARC-support (nov-12)**

The stable release of NAS followed the general released schedule, though slightly delayed. NetarchiveSuite 4.0<sup>9</sup> was released on 28.01.2013 after a thorough release test.

### **M9 - Final project report (nov-12)**

The final report was postponed until the completion of the previous milestones, ie. releases of NAS and JHove2.

---

<sup>7</sup><https://bitbucket.org/jhove2/main/wiki/JHOVE2-Downloads>

<sup>8</sup><https://sbforge.org/display/NAS/2012-October+workshop+at+SB>

<sup>9</sup><https://sbforge.org/display/NAS/NetarchiveSuite+4.0+Release+Notes>

## 3 Released software

### 3.1 Expected

As part of the project proposal the following software releases were expected:

- JHove2 with GZip, ARC and WARC modules

JHOVE 2.1.0 binary and release notes can be found here:

<https://bitbucket.org/jhove2/main/wiki/JHOVE2-Downloads>

<https://bitbucket.org/jhove2/main/downloads>

The source code is also on Bitbucket as a Mercurial repository and can be found here:

<https://bitbucket.org/jhove2/main/src>

- NetarchiveSuite with WARC support

NetarchiveSuite 4.0.x binary and release notes can be found here:

<https://sbforge.org/display/NAS/NetarchiveSuite+4.0.X+Release+Notes>

The source code is hosted by the Danish State Library in SVN here:

<https://sbforge.org/svn/netarchivesuite>

### 3.2 Additional

In the process of developing the JHOVE2 modules some additional software was created to ease the overall development.

- Java Web Archiving Toolkit(JWAT)

JWAT is a standalone library for GZip, ARC and WARC manipulation. Basically it is classes for reading, writing and validating GZip, ARC and WARC files.

The JWAT homepage is location here:

<https://sbforge.org/display/JWAT/JWAT>

The source code is hosted on bitbucket as a Mercurial repository.

<https://bitbucket.org/nclarkekb/jwat>

- JWAT-Tools

A small commandline utility which can be use for different GZip, ARC and/or WARC related tasks. Among these tasks are GZip/ARC/WARC validation and ARC2WARC conversion. This tool can also easily be extend or reused in other projects, see below.

Documentation for JWAT-Tools is available from the JWAT homepage.

The source code is hosted on bitbucket as a Mercurial repository.

<https://bitbucket.org/nclarkekb/jwat-tools>

- JWAT-Tools-GUI

A small GUI application which extends the testing ability of the commandline utility. It also displays the results in a more manageable way. Each file in turn can then have its records listed including the number of errors/warnings found. Finally each record can be viewed for the exact error/warning messages, the ARC/WARC header, optional HTTP header and in some cases the payload. <https://bitbucket.org/nclarkekb/jwat-tools-gui>

### 3.3 Other projects using JWAT

After the stable release of JWAT it was also reused in the following small projects:

- **JWAT-Tools-SOLR**  
A small project that iterates through ARC/WARC records, runs the payload through TIKA and prepares to upload the result to SOLR.  
<https://bitbucket.org/nclarkekb/jwat-tools-solr>
- **JWAT Wayback ResourceStore**  
Various 1.7.1 snapshot versions of the Wayback machine have problems with non compressed ARC/WARC files so this small experimental ResourceStore was implemented to use the JWAT readers instead of the Heritrix ones.  
<https://bitbucket.org/nclarkekb/jwat-wayback-resourcestore>
- **LAP-Writer-WARC**  
The WARC writer for INA's LiveArchivingProxy depends on JWAT for writing.  
<https://bitbucket.org/nclarkekb/lap-writer-warcc>
- **Retro**  
At Netarkivet.dk we have used JWAT to validate and build HTML indexes of data converted from older formats to WARC files.

## 4 JHOVE2 development

Initially the tasks related to the JHOVE2 milestones were to take the existing GZip and ARC modules and modify them to work with the current JHOVE2 code base. After this a new WARC module was to be implemented and a new release of JHOVE2 was to be built at some point before the deadline. Also included in these milestones was the writing of technical specifications for the WARC module for publishing on the JHOVE2 wiki alongside the existing modules specifications.

Two problems arose shortly into that plan. The JHOVE2 architecture had change to a degree, that the GZip and ARC modules could not easily be modified to the new JHOVE2 code base. Secondly continuous testing of JHOVE2 modules would have been much more time consuming than just testing the modules separately or as a separate project.

It was quickly decided that it would be best to have the GZip, ARC and WARC code as a separate project, which could then be used by the JHOVE2 modules. The new GZip, ARC and WARC modules were implemented by looking at existing JHove2 modules and the old GZip and ARC modules.

At first the GZip/ARC code was moved to the separate project and modified to improve the structure and overall code quality. The WARC package was implemented gradually while reimplementing the GZip/ARC packages here and there. Eventually the GZip and ARC packages were more or less completely rewritten. Mostly because they were structured badly and did not have sufficient validation to cover all possible cases.

The first draft of the WARC technical specifications did not include a description of the validation process nor a description of which types of warnings/errors could be expected to be reported. As the WARC technical specification was based on the GZip/ARC ones these were also amended to include the same level of information about validation and warnings/errors reported.

The following tasks were also undertaken even though they were not mentioned in the final proposal.

- GZip/ARC JHOVE2 modules more or less rewritten from scratch.
- GZip reader/validator completely rewritten.
- GZip writer implemented.
- ARC reader/validator more or less completely rewritten.
- ARC writer implemented.
- WARC writer implemented.
- Improved GZip/ARC technical specifications, including description of validation process and warnings/errors reported.



The GZip, ARC and WARC writers were not required to complete the JHOVE2 milestones. However it made subsequent unit testing easier as test files could be made automatically. As a third party library JWAT and tools also benefited greatly from having all the required functionality in one package.

## 5 NetarchiveSuite(NAS) development

The NAS milestones consisted of various sub tasks designed to add WARC support to the project.

The tasks could be summarized as the following:

- Establish the WARC metadata format to be used
- Write WARC metadata files
- Manage Heritrix harvesting in WARC
- Read WARC metadata files
- Run batch jobs on WARC files
- Enable Wayback to access WARC files in the archive.

Since NAS already uses Heritrix's ARC reader/writer it more or less prevented the use of JWAT for WARC support. However JWAT could still be utilized as minor helper classes.

The most obvious place to start was in the metadata code. The old ARC metadata code had to be moved into separate classes leaving only generic interfaces exposed to the rest of NAS. After these changes were implemented and tested, work on implementing the WARC metadata classes was started. At the same time the batch system was expanded with two new types of batch jobs. One for running batch jobs on WARC files and another for running batch jobs on both ARC and WARC files.

Managing Heritrix from NAS with WARC also required the order.xml files managed by NAS and sent to Heritrix to be updated to include additional configuration for a WARC writer processor. Besides this some overall configuration was also required to tell NAS which format to use at startup since only one format at a time can be used for all harvests.

Adding Wayback access to the WARC files in the NAS bit archive was solved by returning the WARC record as an ARC record. When Wayback actually needs the WARC header for additional features this will need to be changed.

Between the developer and stable release there were some communication with BnF about the metadata format used. This was based on discussions at one of our workshops. For the result see appendix I.

Although development should have been split between JHOVE2 and NAS, most of the actual time was used on JHOVE2. As a consequence final touches on the NAS WARC support implementation was done by Søren Vejrup Carlsen as part of development on the stable 4.0 release.

## 6 ARC to WARC Migration

### 6.1 Harvesting

Early on it was decided to migrate to harvesting in WARC instead of ARC when NAS WARC support was deemed stable enough. The first version of NAS with WARC support was run in ARC mode, primarily because it was a developer release which had not been thoroughly tested and secondarily to see if the changes had corrupted the rewritten ARC functionality. With no serious problem found in the developer release (3.21), the stable release (v4.0) with WARC support was on track. After the stable version of NAS with WARC support was ready, Netarkivet.dk upgraded to this version prior to it's next planned broadcrawl. Besides a difference in the ARC/WARC writer API causing too many opens files, no other serious bugs emerged after the switch from ARC to WARC.

### 6.2 The Archive

At some point Netarkivet.dk will migrate the archive from ARC to WARC, but there are no specific plans yet. Since the archive is made up of uncompressed ARC files it is unfeasable to keep the original and converted files. In the process of writing JWAT-Tools an experimental ARC to WARC converter was also implemented.

Two issues are relevant when migrating. One is the amount of time required to migrate the data and the other is validating that all data has in fact been migrated correctly.

Another thing that emerged while writing the ARC to WARC converter was the fact that older ARC records can be difficult to migrate since the records can have a 'no-type' content type in which case the payload has to be run through TIKa, File, Droid or similar identification tools. And in many cases the payload does not even include a valid HTTP header. In a lot of cases a semi-valid HTTP header is present which can be repair. In others they include an ICE-Cast streaming header. Most headers can be repair fairly easily. However each case has to be handle programatically in the converter.

Migration was tested on a machine with 2 CPUs with each 12 cores, 99GB ram and local RAID storage. If memory serves, 1TB ARC files could be migrated to WARC in approximately 4 hours. Migrating pre 2005 ARC files could not be done without a repair function unless it is acceptable that some data ends up unbrowsable through Wayback. Implementing a repair function results in a lot of re-runs to verify the correctness which will of course increase the total time required to migrate data.

JWAT-Tools does not have a comparison command as of yet so verifying the migration can instead be done by building a CDX of the original data and one of the migrated data and comparing the two.

## 7 Experience learned

### 7.1 WARC

Working with ARC and WARC some difference are obvious. WARC is an official ISO with all that this entails. ARC on the other hand is just some semi organized words written down. Even though the document is fairly straightforward, there is a discrepancy in that the document describes a line feed after each record which in real life is not actually the case if you examine how Heritrix writes ARC files. On a side note looking at the 'official' description of the CDX format, most of the possible columns are probably only known to the original implementers of CDX in Wayback. Not so long ago I tried re-creating an URL normalizer to be able to lookup data in Wayback created CDX files. Using a lots of hours has only shown that the scheme used to normalize URLs does not seem fairly logical. This only proves the point that tools/formats used widely by a community must be based on official standards.

Reading ARC records is fairly straightforward until records are corrupt. In those cases the only way to look for more records is to look for lines with a specific number of space separated strings. This number is different for V1 and V2 ARC records. To make matters worse, some of the items in an ARC header can also be corrupt making it a bigger challenge to detect the beginning of a real record.

Using WARC it is easier to detect a record in a damaged stream. You just look for 'WARC/x.x' and a CR-LF pair.

One important thing to notice is the references to RFC's in the standard. A lot of these references point to different header related standards. The most complex part of writting a compliant WARC reader is supporting all the these different references. A WARC header value can utilize Encoded-Words, Quoted-Printable, LeadingWhiteSpace and/or UTF-8. UTF-8 is an addition to WARC. However reading the WARC standard it is ambiguous whether it is permissible to use UTF-8 encoded characters >255 directly in the header. It is permissible when using Quoted-Printables.

The standard also allows for the creation of additional record types and custom headers. This is only a problem when the WARC records have to validated. Ideally a WARC validator should be custamizable in that record types and headers are configurable leaving the validator generic and thus expandable. Some general guidelines would also be useful to help people decide whether new record types and headers are really necessary. Personally I would prefer to use content-types including parameters as much as possible instead of inventing new record types.

Given the recent polemic about the identical payload truncation header, wording in the standard could probably be a bit more precise regard this header value and generally which header can appear in which record type.

WARC is however still a great improvement to ARC.

## 7.2 Development 'ecosystem'

For lack of a better word, 'ecosystem' supposed to cover all aspects of a project from planning, releasing, testing to actual development.

The big difference between JHOVE2 and NAS is funding. JHOVE2 is not currently being funded and development is almost non existing except in cases when one of the partners has a little bit of extra time. NAS has funding but not enough for all the development it requires.

JHOVE2 uses github.com for the source repository, Wiki and bug tracking. NAS in turn uses SVN, a bunch of different Wikis(Confluence, etc.), JIRA for bug tracking, Fisheye/Crucible for Code review and Jenkins for automatic build/test. Although Confluence, JIRA, Crucible, Jenkins are not perfect they are however a huge improvement compared to github. On the other hand they required a bit more maintainance which there has to be allocated time for.

JHOVE2 does not have an official release strategy and until recently did not have guidelines on how to build a release. In conjunction with the latest release of JHOVE2 a document was assembled with all the relevant information required to develop and build JHOVE2. This is a huge improvement. NAS on the other hand aim at 4 releases per years, 2 stable and 2 development. The difference between the two types of releases is the amount of testing performed. Testing for a stable release can take anywhere from 1-3 weeks while the time required for a development release is closer to 1. Personally I'm not quite sure why only two releases are stable, but presumably it is because of the amount of time required for testing. For some reason integration testing of NAS has not been automated yet.

Both NAS and JHOVE2 could benefit from refactoring. JHOVE2 would benefit from a refactoring into a modularized maven project and with atleast some refactoring of the persistence layer. Another problem for JHOVE2 is run time, it is god awful slow running in recursive mode which is a problem for large scale use.

In my opinion NAS would also benefit from major refactoring since development in recent years has mainly focused on fixing old problems and adding new features. The main problem with refactoring NAS is funding.

JHOVE2 would benefit greatly by planning regular releases, using Jenkins for code review and of course getting a bit of funding for maintainance. Testing is a bit ad hoc and up to each partner according to which data is available locally. An dedicated Wiki/bug tracking outside of github would also be an improvement but is not crucial to the project.

NAS seems to have several generations of Wikis running, they are slowly being cleaned and migrated, but it is not an ideal situation until this is completed. Besides this NAS would also benefit from converting to maven and git.

On a side note Heritrix and Wayback would benefit greatly by improving their 'ecosystem'. Regular releases, an open bug tracking system, integration testing and of course a lot of refactoring of the code.

## 8 Final remarks

I must admit to having a compulsive urge for code perfection, unfortunately this does not always fare so well with strict deadlines. That said I hope people will find the outcome of this project useful and that it will benefit the IIPC community. Thanks must go to the JHOVE2 developers, people at BnF and all the rest that have been helping in completing the project.

## **A Proposal: Foster WARC usage in scalable Web Archiving workflows using Jhove2 and NetarchiveSuite**



## Foster WARC usage in scalable Web Archiving workflows using Jhove2 and NetarchiveSuite

*A project proposal from the NetarchiveSuite Community to IIPC Program Officer and Steering Committee*

### **Stakeholders and contacts:**

*Netarchive.dk: Birgit Nordsmark Henriksen & Bjarne Andersen*

*Bibliothèque nationale de France: Sara Aubry & Clément Oury*

*Österreichische Nationalbibliothek: Michaela Mayr*

### **Context and baseline**

Since May 2009, memory institutions and other digital archiving organizations can use the WARC (Web ARChive) file format, which was officially released as an international standard (ISO 28500:2009) to store and preserve documents harvested on the web. WARC is an extension of the ARC format, which has been extensively used since 1996 by the Internet Archive and by most members of the IIPC. These institutions recognized the need to extend the ARC format to add new capabilities, notably the recording of HTTP requests, the recording of local metadata, allocation of a unique identifier for every contained file, management of duplicates and migrated records, and the segmentation of records.

International standardization was a critical step towards the wide adoption of the WARC format. As part of this effort, IIPC also set up in November 2009 a “WARC usage task force” to write implementation guidelines, which were delivered and approved by the Preservation Working Group the following year. However today, because production and preservation workflows have recently been settled and are extensively used, many members are still using the ARC format for production purposes while acknowledging the need to transition to WARC. Difficulties related to the progress of the WARC tools project haven’t helped bringing the required confidence to organize this transition. IIPC members seem to expect some pilot institutions to do the first move and to report on real-life, large scale in-house implementation tests of WARC in their production and preservation workflows in order to gain confidence in the format, learn from pioneer experiences and ultimately envisage their own transition from ARC to WARC.

It should be noted that this project does not overlap with the requirements or expected outcomes of the WARC Tools project lead by Hanzo. It should also be noted that there may be interesting interaction or continuation of this project with the recently launched 3,5 years European project SCAPE<sup>1</sup>. The University Library of Aarhus being lead of the SCAPE Characterisation Workpackage and the National Library of Austria being involved in both projects as well, close coordination would be guaranteed.

### **The NetarchiveSuite community now proposes to develop the usage of the WARC format working into two directions:**

- 1) give the ability to ingest WARC files into digital preservation workflows using JHOVE2,
- 2) study and implement WARC in a scalable production workflow using NetarchiveSuite as an example.

### **Part 1: WARC files into digital preservation workflows: the JHOVE2 solution**

JHOVE2 is an open source software for format-aware characterization of digital objects. JHOVE2 enables format identification, feature extraction, validation and assessment. The JHOVE2 project is a collaborative undertaking of the California Digital Library, Portico, and Stanford University. JHOVE2 is made freely available under the terms of the BSD open source license. This part of the proposal aims at providing JHOVE2 support for the following functions in order to make it a more useful tool for web archiving:

- *Module for the WARC format:* Characterization performed at the record level, including both record headers and blocks: Warcinfo, response, resource, request, metadata, revisit, conversion, continuation. The proposal includes a significant amount of resources for developing this module. This will leave enough time to develop both the baseline WARC-module but also do advanced functionality based on input from the IIPC community
- *Integration of the ARC and GZIP modules developed by BnF into the core of JHOVE2.*

<sup>1</sup> SCALable Preservation Environments : <http://internetmemory.org/en/index.php/projects/scape>



## NetarchiveSuite

This project is to complement and continue the effort launched in 2010 to develop modules to the JHOVE2 project and software lead by the California Digital Library. BnF, one of the stakeholders of the present proposal, has been actively involved in this project for which it has spent a dedicated budget outsourced to a private company, ATOS, which is in charge of building BnF's digital repository archiving and preservation system. ATOS and BnF have developed ARC and GZIP modules to Jhove2. This development took place in cooperation with CDL and with the support of IIPC Program Officer.

### **Part 2: Study and implement WARC in a scalable production workflow: the NetarchiveSuite environment**

The NetarchiveSuite is a complete web archiving open source software package. It gives the ability to prepare, schedule, run and monitor harvests of websites. It also enables to perform quality assurance and preserve harvested content. NetarchiveSuite is used for production purposes, developed and maintained by the NetarchiveSuite community which currently includes the State and University Library, Aarhus, Denmark, The Royal Library, Copenhagen, Denmark, the National Library of France and the National Library of Austria. The community hopes to extend to new partners in the future.

This part of the proposal aims at:

- studying the implementation of the WARC format into the Heritrix web crawler in the light of the WARC standard and IIPC WARC implementation guidelines written by the WARC Usage task force,
- as the format may be revised within the ISO in May 2012, gathering possible fixes or evolutions needed by IIPC members and updating the guidelines if necessary (BnF, as convenor of the *ad hoc* standardization group at ISO and co-lead of the PWG, could help with this),
- studying and documenting the impact of WARC in harvesting and post-harvesting processes (such as indexing and feeding metadata into a curator tool), which would benefit all local curator tools,
- implementing WARC into NetarchiveSuite, while keeping ARC compatibility alive,
- delivering a report based on the experience of the 3 partner institutions.

### **Budget, Management, Timeline**

- **Development**

- 1man-year (ca. 1400 hours in Denmark) based on the recruitment of a developer for 12 months responsible to achieve Jhove2 developments, implementing with NetarchiveSuite, along with other testing and evaluation tasks.
- technical project management would be located in Denmark and closely connected to NetarchiveSuite management.
- IIPC funded developer would also be based in Denmark, at Netarchive.dk, who would allocate a work station / office.
- **Costs: 92,400 euros (see detailed task description and estimation)**

- **IIPC related project management & coordination**

- Distribution of Roles: BnF: specifications lead; Netarchive.dk: development lead; ÖNB: testing (Part 2 of the project), all partners to implement and report.
- Specification, testing, reporting and overall project management will be done with internal resources within the three partners as a collaborative self-financing contribution to the project. This part is estimated to 4MM.
- Coordination requires 2 to 3 project team meetings between partners over a 12 months period in Europe.

**Costs: (travelling expenses): 12,000 euros**

- **Support for editing, translation and dissemination work**

- Delivery of a report and presentation at IIPC events of transition testing and experience from ARC to WARC
- **Costs: 5,000 euros**

### **Total project costs: 114,350 euros**





## NetarchiveSuite

### Timeline & project milestones

Project launch (after recruitment of developer by Netarchive.dk): between October & November 2011

#### Milestones / Deliverables

jan-2012: Technical specification of WARC module for JHOVE2

apr-2012: Prototype Code release of JHOVE2-modules

may-2012: Progress report at IIPC GA 2012, Washington D.C

Jul-2012: Developer Release of NetarchiveSuite with WARC-support

sep-2012: Final Code release of JHOVE2-modules

nov-2012: Stable Release of NetarchiveSuite with WARC-support

nov-2012: Final project report (and possible presentation/workshop attached to an IIPC event or workshop in the Fall)

### Detailed development task descriptions

#### Tasks for NetarchiveSuite WARC-implementation:

1. harvesting (configuration of NetarchiveSuite and heritrix): 50 hours
2. indexing of warc (creation of CDX-files and warc-indexing): 75 hours
3. Generic batch-job for warc: 50 hours
4. metadata-generation (creation of post-crawl WARC metadata-files): 75 hours
5. User-interface adjustments: 20 hours
6. Support for ARC/WARC switch in various NetarchiveSuite modules: 125 hours
7. Code-reviews: 50 hours
8. Unit-Testing: 75 hours

Total: **520 hours**

#### Tasks for JHOVE2 implementation:

1. Developer training in JHOVE2 architecture and APIs: 40 hours
2. Analysis of format requirements WARC: 60 hours
3. Technical specifications for WARC: 60 hours
4. Stakeholder review and final specs: 30 hours
5. Coding of WARC-module: 120 hours
6. Coding of advanced features for the WARC-module: 120 hours
7. Integration of BnF ARC/GZIP-module into JHOVE2-core: 50 hours
8. Prototype Code release: 40 hours
9. Code reviews: 50 hours
10. Functional and Performance testing: 50 hours
11. Refactoring: 40 hours
12. Final Code release: 20 hours
13. Documentation of new components: 30 hours
14. Documentation of changes to core JHOVE2 APIs: 20 hours

Total: **730 hours**

Technical project management: **150 hours**

#### **Project Total: 1400 hours**

Senior developer salary hourly rate: 55 euros

Total cost: 77,000 euros

Overhead rate: 20% = 15,400 euros

**Total salary cost: 92,400 euros**

## **B   Status update 11 Apr. 2012**

## Project update

### 1 JHove2 WARC technical specification (Part 1)

[https://bitbucket.org/nclarkekb/jhove2-iipc/downloads/JHOVE2-WARC-module-spec-2\\_0\\_0RC1.doc](https://bitbucket.org/nclarkekb/jhove2-iipc/downloads/JHOVE2-WARC-module-spec-2_0_0RC1.doc)

- Submittet to **Aaron Binns**, who had some questions and comments.
- Issues to be amended
  - Describe how the module validates against the *ISO* standard.
  - Include a list of generated errors/warnings.
  - Rephrase description of temporary file creation.

### 2 JHove2 module implementations (Part 1)

<https://sbforge.org/display/NAS/WARC+support+in+JHove2>

<https://bitbucket.org/nclarkekb/jhove2-iipc/downloads>

#### 2.1 ARC/WARC format modules

- The *ARC* and *WARC* modules are more or less complete.
- Stable release on hold until **completion** of the *JWAT* libraries.

#### 2.2 GZip format module

- The *GZip* module is almost complete.
- **Cleanup** to remove old code and use *JWAT GZip* functionality instead.

#### 2.3 File identification module

- Imported from *JHove2-Bnf* branch and modified to compile with trunk.
- Correctly identifies *WARC* and *GZip* files.
- *File* identifies *ARC* files but not when used from *JHove2*. (**Debugging required**)

#### 2.4 XSLDisplayer display module

- Imported from *JHove2-BnF* branch and modified to compile with trunk.
- *BnF containerMD* XSL wrapper compiled and included in trunk.
- *containerMD.xsl* untested but most likely requires **modifying** to work with new modules.

The JHoNas Project

April 11, 2012

---

### 3 JWAT (Part 1.5)

Implements all the actual *ARC*, *WARC* and *GZip* functionality.

#### 3.1 Library (reusable!)

<https://sbforge.org/display/JWAT/JWAT>

<https://bitbucket.org/nclarkekb/jwat/>

Features:

- GZip reader/validator/writer more or less complete.
- ARC reader/validator almost complete.
- WARC reader/validator more or less complete.
- WARC writer almost complete.
- Common classes almost complete.

#### 3.2 Tools (recyclable)

<https://bitbucket.org/nclarkekb/jwat-tools/downloads/>

Handy command line utility which currently

- Validates GZip, ARC and WARC archives.
- Decompresses \*.arc.gz, \*.warc.gz and \*.gz files.
- Compresses \*.warc and single files.
- Parallelized validation (threads configurable)

### 4 WARC support in NetarchiveSuite (Part 2)

- BnF and Netarkivet.dk has a meeting in Copenhagen.
  - To work on the WARC metadata structure for NAS.
  - Refine the already defined tasks for WARC in NAS.
  - Plan for the GA.
- Started work on the WARC support in NAS tasks.

---

April 11, 2012

2

## C Status update 21 Apr. 2012

## Project update

### 1 JHove2 WARC technical specification (Part 1)

[https://bitbucket.org/nclarkekb/jhove2-iipc/downloads/JHOVE2-WARC-module-spec-2\\_0\\_0RC1.doc](https://bitbucket.org/nclarkekb/jhove2-iipc/downloads/JHOVE2-WARC-module-spec-2_0_0RC1.doc)

- Submittet to **Aaron Binns**, who had some questions and comments.
- Issues to be amended
  - Describe how the module validates against the *ISO* standard.
  - Include a list of generated errors/warnings.
  - Rephrase description of temporary file creation.

### 2 JHove2 module implementations (Part 1)

<https://sbforge.org/display/NAS/WARC+support+in+JHove2>

<https://bitbucket.org/nclarkekb/jhove2-iipc/downloads>

#### 2.1 ARC/WARC format modules

- The *ARC* and *WARC* modules are more or less complete.
- Stable release on hold until **completion** of the *JWAT* libraries.

#### 2.2 GZip format module

- The *GZip* module is almost complete.
- **Cleanup** to remove old code and use *JWAT GZip* functionality instead.

#### 2.3 File identification module

- Imported from *JHove2-BnF* branch and modified to compile with local fork.
- Correctly identifies *WARC* and *GZip* files.
- *File* identifies *ARC* files but not when used from *JHove2*. (**Debugging required**)

#### 2.4 XSL display module

- Imported from *JHove2-BnF* branch and modified to compile with local fork.
- *BnF containerMD* XSL wrapper compiled and included in local fork.
- *containerMD.xsl* untested but most likely requires **modifying** to work with current JHove2 module output.

### 3 JWAT (Part 1.5)

Implements all the actual *ARC*, *WARC* and *GZip* functionality.

#### 3.1 Library (reusable!)

<https://sbforge.org/display/JWAT/JWAT>

<https://bitbucket.org/nclarkekb/jwat/>

Features:

- GZip reader/validator/writer more or less complete.
- ARC reader/validator almost complete.
- WARC reader/validator more or less complete.
- WARC writer almost complete.
- Common classes almost complete.
- 100kb jars and no external dependencies.

Substitute for the readers/writers in Heritrix.

#### 3.2 Tools (recyclable)

<https://bitbucket.org/nclarkekb/jwat-tools/downloads/>

Handy command line utility which currently

- Validates GZip, ARC and WARC archives.
- Decompresses \*.arc.gz, \*.warc.gz and \*.gz files.
- Compresses \*.warc and single files.
- Parallelized validation (threads configurable)

Application with a simple graphical user interface

- Add WARC, ARC and GZip files to the work queue.
- Validate WARC, ARC and GZip files.
- Overview of queue with progress and result in a table.

## 4 WARC support in NetarchiveSuite (Part 2)

- BnF and Netarkivet.dk had a meeting in Copenhagen.
  - To work on the WARC metadata structure for NAS.
  - Refine the already defined tasks for WARC in NAS.
  - Plan for the GA.
- Started work on the WARC support in NAS tasks.
  - Prototype for handling WARC files in batch jobs.
  - Added some functionality for using WARC instead of ARC for NAS metadata.

## 5 Milestones

### 5.1 M1: Technical spec. of WARC module for JHOVE2 (Jan/Feb-2012)

Progress: **98%**

Has no significant impact on the overall module implementation.

Tasks:

- Minor changes and resubmission.

### 5.2 M2: Prototype Code release of JHOVE2-modules (Mar-2012)

Progress: **95%**

Since the modules are almost complete(v1.0) the prototype milestone should be a formality. Three Release Candidates are available through the link in section 2, earliest from 10-feb-2012.

Tasks:

- BnF will review the JHove2 output on WARC files. (Apr/May-2012)
- The prototype will be submitted to Aaron for review after the specification have been accepted.

### 5.3 M3: Dev. Release of NetarchiveSuite with WARC-support (Aug-2012)

Progress: **15%**

<https://sbforge.org/jira/browse/NAS-1720>

Implementation has begun, the specific tasks and their progress can be browsed in the link above.

Only issue currently is the suitability of the Heritrix readers vs. JWAT and issues relating to this.

---

April 21, 2012

3 of 4



**5.4 M4: JHove2 WARC, ARC and GZip modules v1.0 (Sep-2012)**

Progress: **90%**

The implementation part of this milestone is almost complete.  
Remaining tasks fall into the following categories.

- Cleanup GZip modules.
- Complete remaining issues on the JWAT library.
- Testing of JHove2 modules at BnF. (May, if possible)
- Approval of program officer (Aaron)

There are however some administrative tasks which must be overcome.

- Integration with JHove2 trunc (**Still no word from the JHove2 partners!**)

**5.5 M5: Final project report (Nov-2012)**

Progress: **1%**

Tasks

- Establish extent of report.
- Author report, possibly rehashing available materials at that time.

## **D   Status update 26 Jun. 2012**

## Project update

### 1 Milestones

Includes actions from last project update and open tasks.

Appendix A contains an overview of the JWAT sub-project.

Appendix B contains an overview of the JHove2 project.

#### 1.1 M1: Technical spec. of WARC module for JHOVE2 (Jan/Feb-2012)

Progress: **99.9%**

[https://bitbucket.org/nclarkekb/jhove2-iipc/downloads/JHOVE2-WARC-module-spec-2\\_0\\_0RC2.doc](https://bitbucket.org/nclarkekb/jhove2-iipc/downloads/JHOVE2-WARC-module-spec-2_0_0RC2.doc)

Actions:

- An amended version of the technical specifications was resubmitted to **Aaron Binns**.
- The updated version now includes:
  - Description of how the module validates against the *ISO* standard.
  - Includes a list of generated errors/warnings.
  - Rephrased description of temporary file creation.
- Technical specifications and milestone approved.

Tasks:

- Send invoice to Clément/BnF and receive payment for M1.

#### 1.2 M2: Prototype Code release of JHOVE2-modules (Mar-2012)

Progress: **97.5%**

<https://sourceforge.net/projects/nas/warc-support-in-jhove2/>

<https://bitbucket.org/nclarkekb/jhove2-iipc/downloads>

Actions:

- All remaining development has been moved from Milestone 4 to Milestone 2.
- Jhove2 IIPC RC4 was released 2012-05-12.
- BnF has reviewed some initial JHove2 output from some WARC file tests.

Tasks:

- Complete remaining issues on the JWAT library.
- Cleanup GZip modules.
- Minor changes reported by BnF after initial review.

- Consolidate all modules and commit the extra BnF modules to the repository.
- The prototype can be submitted to Aaron for review after some minor issues reported by BnF have been fixed.

Estimated work on JWAT: 2-3 days.

Estimated work on JHove2: 2-3 days.

### 1.3 M3: Dev. Release of NetarchiveSuite with WARC-support (Aug-2012)

Progress: **25+**%

<https://sfborge.org/jira/browse/NAS-1720>

Actions:

- At the BnF and Netarkivet.dk meeting in Copenhagen it was decided to extend the metadata structure step by step..
- NAS supports harvesting in ARC or WARC.
- NAS supports metadata generation in ARC or WARC.
- NAS almost supports CDX generation from WARC files in the Harvest documentation phase.
- NAS can now run WARC batch jobs.
- A WARC CDX extractor batch job has been implemented.
- NAS can now run archive batch jobs (ARC and/or WARC) files.
- An archive CDX extractor batch job has been implemented.

Tasks:

- Unit test the new features implemented.
- Review the new features implemented.
- Complete work on NAS issues as they appear on JIRA.

Estimated progress is likely a bit concervative.

**1.4 M4: JHove2 WARC, ARC and GZip modules v1.0 (Sep-2012)**Progress: **90%**

Actions:

- Any implementation work planned for this milestone has been moved to M2.
- Requested and got approved as a JHove2 submitter.
- Suggested that JHove2 should use Crucible/Fisheye/Jenkins.

Tasks:

- Create Crucible/Jenkins project at SBForge.org (Mikis).
- Testing of JHove2 modules at BnF by Thomas Ledoux. (Beginning of July)
- Merge with JHove2 main codebase.
- Approval of program officer (Aaron)

Uncertainties:

- Planning, merging and releasing of JHove2 with JHoNAS components.

**1.5 M5: Final project report (Nov-2012)**Progress: **1%**

Tasks

- Establish extent of report.
- Author report, possibly rehashing available materials at that time.

## A JWAT

### A.1 JWAT packages (reusable!)

<https://sbforge.org/display/JWAT/JWAT>

<https://bitbucket.org/nclarkekb/jwat/>

Implements all the actual *ARC*, *WARC* and *GZip* functionality.

Features:

- Common classes complete.
- GZip reader/validator/writer complete.
- ARC reader/validator almost complete.
- WARC reader/validator/writer complete.
- 100kb jars and no external dependencies.

Alternative to the readers/writers in Heritrix.

### A.2 JWAT-Tools (recyclable)

<https://bitbucket.org/nclarkekb/jwat-tools/downloads/>

Handy command line utility which currently

- Validates GZip, ARC and WARC archives.
- Decompresses \*.arc,gz, \*.warc,gz and \*.gz files.
- Compresses \*.warc and single files.
- Parallelized validation (threads configurable)
- ARC to WARC converter.

### A.3 JWAT-Tools-GUI

Application with a simple graphical user interface.

- Add WARC, ARC and GZip files to the work queue.
- Validate WARC, ARC and GZip files.
- Overview of queue with progress and result in a table.

## B JHove2 modules

### B.1 ARC/WARC format modules

- The *ARC* and *WARC* modules are more or less complete.
- Stable release on hold until **completion** of the *JWAT* libraries.

### B.2 GZip format module

- The *GZip* module is almost complete.
- **Cleanup** to remove old code and use *JWAT GZip* functionality instead.

### B.3 File identification module

- Imported from *JHove2-BnF* branch and modified to compile with local fork.
- Correctly identifies *WARC* and *GZip* files.
- *File* identifies *ARC* files but not when used from *JHove2*. (**Debugging required**)

### B.4 XSL display module

- Imported from *JHove2-BnF* branch and modified to compile with local fork.
- *BnF containerMD* XSL wrapper compiled and included in local fork.
- *containerMD.xsl* untested but most likely requires **modifying** to work with current JHove2 module output.

## **E   Status update 1 Augr. 2012**



## Project update

### 1 Milestones

Includes actions from last project update and open tasks.

Appendix A contains an overview of the JWAT sub-project.

Appendix B contains an overview of the JHove2 project.

#### 1.1 M1: Technical spec. of WARC module for JHOVE2 (Jan/Feb-2012)

Progress: **100.0%**

[https://bitbucket.org/nclarkekb/jhove2-iiipc/downloads/JHOVE2-WARC-module-spec-2\\_0\\_0RC2.doc](https://bitbucket.org/nclarkekb/jhove2-iiipc/downloads/JHOVE2-WARC-module-spec-2_0_0RC2.doc)

Actions:

- Milestone payment should be complete by now.

#### 1.2 M2: Prototype Code release of JHOVE2-modules (Mar-2012)

Progress: **98.5%**

<https://sbforge.org/display/NAS/WARC+support+in+JHove2>

<https://bitbucket.org/nclarkekb/jhove2-iiipc/downloads>

Actions:

- Updated to use JWAT-1.0.0-SNAPSHOT.
- ARC Module uses the new ARC reader/validator.
- GZip Module uses the new GZip reader/validator.
- Changed output issues reported by BnF after initial review.
- File Module ARC issue debugged and located. File Module integration complete.
- Jhove2 IIPC RC5 and RC6 have been released.
- Requested Aaron Binns to start the approval process of the prototype milestone. Further material might be required prior to approval.

Tasks:

- JWAT/JHove2 reviews in progress at kb.dk.
- Provide extra material if required.

### 1.3 M3: Dev. Release of NetarchiveSuite with WARC-support (Aug-2012)

Progress: **25** + %

<https://sbforge.org/jira/browse/NAS-1720>

Working:

- NAS supports harvesting in ARC or WARC.
- NAS supports metadata generation in ARC or WARC.
- NAS can now run WARC batch jobs.
- A WARC CDX extractor batch job has been implemented.
- NAS can now run archive batch jobs (ARC and/or WARC) files.
- An archive CDX extractor batch job has been implemented.

Tasks:

- Debug NAS CDX generation from WARC files in the Harvest documentation phase.
- Unit test the new features implemented.
- Review the new features implemented.
- Complete work on NAS issues as they appear on JIRA.
- August will mostly be used to prepare a development release of NAS with WARC support.

Estimated progress is likely a bit conservative.

**1.4 M4: JHove2 WARC, ARC and GZip modules v1.0 (Sep-2012)**Progress: **95%**

Actions:

- JHove2 Crucible/Jenkins project created at SBForge.org.
- Some testing of JHove2 modules at BnF by Thomas Ledoux. (Outcome uncertain)

Tasks:

- Complete ARC refactoring in the JWAT library.
- Cleanup JHove2 code removing unused code etc.
- Integrate working XLS Display Module with IIPC repository.
- Merge with JHove2 main codebase.
- Approval of program officer (Aaron)

Uncertainties:

- Planning, merging and releasing of JHove2 with JHoNAS components.

**1.5 M5: Final project report (Nov-2012)**Progress: **1%**

Tasks

- Establish extent of report.
- Author report, possibly rehashing available materials at that time.

## A JWAT

### A.1 JWAT packages (reusable!)

<https://sourceforge.net/projects/jwat/>

<https://bitbucket.org/nclarkekb/jwat/>

Implements all the actual *ARC*, *WARC* and *GZip* functionality.

Features:

- Common classes complete.
- GZip reader/validator/writer complete.
- ARC reader/validator/writer almost complete.
- WARC reader/validator/writer complete.
- 100kb jars and no external dependencies.

Alternative to the readers/writers in Heritrix.

### A.2 JWAT-Tools (recyclable)

<https://bitbucket.org/nclarkekb/jwat-tools/downloads/>

Handy command line utility which currently

- Validates GZip, ARC and WARC archives.
- Decompresses \*.arc.gz, \*.warc.gz and \*.gz files.
- Compresses \*.warc and single files.
- Parallelized validation (threads configurable)
- ARC to WARC converter.

### A.3 JWAT-Tools-GUI

Application with a simple graphical user interface.

- Add WARC, ARC and GZip files to the work queue.
- Validate WARC, ARC and GZip files.
- Overview of queue with progress and result in a table.

## B JHove2 modules

Stable release on hold until **completion** of the *JWAT* libraries.

### B.1 ARC/WARC format modules

- The *ARC* and *WARC* Format Modules are more or less complete.

### B.2 GZip format module

- The *GZip* Format Module is more or less complete.

### B.3 File identification module

- *File* Identification Module should be complete.

### B.4 XSL display module

- XSL Display Module is more or less complete.
- *containerMD.xsl* requires modification to work with the current JHove2 output format.

## **F   Status update 13 Sep. 2012**

## Project update

### 1 Milestones

Includes actions from last project update and open tasks.

Appendix A contains an overview of the JWAT sub-project.

Appendix B contains an overview of the JHove2 project.

#### 1.1 M1: Technical spec. of WARC module for JHOVE2 (Jan/Feb-2012)

Progress: **100.0%**

[https://bitbucket.org/nclarkekb/jhove2-iiipc/downloads/JHOVE2-WARC-module-spec-2\\_0\\_0RC2.doc](https://bitbucket.org/nclarkekb/jhove2-iiipc/downloads/JHOVE2-WARC-module-spec-2_0_0RC2.doc)

No further work to be done.

#### 1.2 M2: Prototype Code release of JHOVE2-modules (Mar-2012)

Progress: **99.0%**

<https://sbforge.org/display/NAS/WARC+support+in+JHove2>

<https://bitbucket.org/nclarkekb/jhove2-iiipc/downloads>

Actions:

- Still waiting for an answer from Aaron Binns concerning the approval of this milestone.

Tasks:

- Provide extra material if required.

#### 1.3 M3: Dev. Release of NetarchiveSuite with WARC-support (Aug-2012)

Progress: **100%**

<https://sbforge.org/jira/browse/NAS-1720>

<https://sbforge.org/display/NAS/NetarchiveSuite+3.21.0+Release+Notes>

Actions:

- NAS with WARC support was released on 5.9.2012.

Actions:

- Testing of NAS with WARC by Bnf and/or ONB.
- Approval of program officer (Aaron Binns)

Additional work on WARC in NAS is now subject to normal issue management and prioritization by netarkivet.dk.

**1.4 M4: JHove2 WARC, ARC and GZip modules v1.0 (Sep-2012)**Progress: **98.5%**

Actions:

- Thomas Ledoux/BnF has tested the different modules and almost all issues should have been fixed now.

Tasks:

- Complete a more relaxed URI validation class.
- Copy unit tests from JWAT to JHove2.
- A JHove2 meeting has been set up for 9/13/2012.
- Prepare for release.
- Merge with JHove2 main codebase.
- Release JHove2 2.1.0.
- Approval of program officer (Aaron Binns)

**1.5 M5: Final project report (Nov-2012)**Progress: **5%**

Tasks

- Establish extent of report.
- Author report, possibly rehashing available materials at that time.

This should not be a very time consuming task.



## A JWAT

### A.1 JWAT packages (reusable!)

<https://sbforge.org/display/JWAT/JWAT>

<https://bitbucket.org/nclarkekb/jwat/>

Implements all the actual *ARC*, *WARC* and *GZip* functionality.

Features:

- Common classes complete.
- GZip reader/validator/writer complete.
- ARC reader/validator/writer complete.
- WARC reader/validator/writer complete.
- Approx. 150kb jars and no external dependencies.

Alternative to the readers/writers in Heritrix.

### A.2 JWAT-Tools (recyclable)

<https://bitbucket.org/nclarkekb/jwat-tools/downloads/>

Handy command line utility which currently

- Validates GZip, ARC and WARC archives.
- Validates XML payload against DTD or XSD declarations (mets, etc.).
- Simple plugin system in progress.
- Decompresses \*.arc.gz, \*.warc.gz and \*.gz files.
- Compresses \*.warc and single files.
- Parallelized validation (threads configurable)
- ARC to WARC converter.

### A.3 JWAT-Tools-GUI

Application with a simple graphical user interface.

- Add WARC, ARC and GZip files to the work queue.
- Validate WARC, ARC and GZip files.
- Overview of queue with progress and result in a table.

## B JHove2 modules

Stable release on hold until **completion** of the *JWAT* libraries.

### B.1 ARC/WARC format modules

- The *ARC* and *WARC* Format Modules should be complete.

### B.2 GZip format module

- The *GZip* Format Module should be complete.

### B.3 File identification module

- *File* Identification Module should be complete.

### B.4 XSL display module

- XSL Display Module should be complete.
- *containerMD.xsl* requires modification to work with the current JHove2 output format.

## **G   Status update 27 Sep. 2012**

## Project update

### 1 Milestones

Includes actions from last project update and open tasks.

For completeness this document now includes all milestones defined by the project and not only the major deliverables.

The following table lists each milestone and its overall status.

M	Date	Description	Status
M1	jan-12	Technical specification of WARC module for JHOVE2	100%
M2	mar-12	Prototype Code release of JHOVE2-modules	99.9%
M3	apr-12	Workshop in Copenhagen on WARC/NAS specifications	99.9%
M4	maj-12	Progress report at IIPC GA 2012, Washington D.C	99.9%
M5	aug-12	Developer Release of NetarchiveSuite with WARC-support	99.9%
M6	sep-12	Final Code release of JHOVE2-modules	99.0%
M7	sep-12	Workshop in Copenhagen/Aarhus on WARC/NAS tests	N/A
M8	nov-12	Stable Release of NetarchiveSuite with WARC-support	N/A
M9	nov-12	Final project report (and possible presentation/workshop attached to an IIPC event or workshop in the Fall)	5%

Appendix A contains an overview of the JWAT sub-project.

Appendix B contains an overview of the JHove2 project.

#### 1.1 M1: Technical spec. of WARC module for JHOVE2 (Jan/Feb-2012)

Progress: **100.0%**

[https://bitbucket.org/nclarkekb/jhove2-iipc/downloads/JHOVE2-WARC-module-spec-2\\_0\\_0RC2.doc](https://bitbucket.org/nclarkekb/jhove2-iipc/downloads/JHOVE2-WARC-module-spec-2_0_0RC2.doc)

No further work to be done.

#### 1.2 M2: Prototype Code release of JHOVE2-modules (Mar-2012)

Progress: **99.9%**

<https://sf Forge.org/display/NAS/WARC+support+in+JHove2>

<https://bitbucket.org/nclarkekb/jhove2-iipc/downloads>

The prototype was completed around August.

Tasks:

- Provide documentation so this milestone can be closed administratively.
- Send invoice.

### 1.3 M3 Workshop in Copenhagen on WARC/NAS specifications (Apr-2012)

Progress: **99.9%**

<https://sbforge.org/display/NAS/NAS+Warc+workshop>

The workshop was held as planned and the outcome is visible on the wiki above.

Tasks:

- Provide documentation so this milestone can be closed administratively.

### 1.4 M4 Progress report at IIPC GA 2012, Washington D.C (May-12)

Progress: **99.9%**

<https://netpreserve.org/>

A short presentation was held at the GA and a demonstration was shown on the PWG workshop.

Tasks:

- Provide presentations so this milestone can be closed administratively.

### 1.5 M5: Dev. Release of NetarchiveSuite with WARC-support (Aug-2012)

Progress: **99.9%**

<https://sbforge.org/jira/browse/NAS-1720>

<https://sbforge.org/display/NAS/NetarchiveSuite+3.21.0+Release+Notes>

NAS with WARC support was released on 5.9.2012.

Tasks:

- Provide documentation so this milestone can be closed administratively.
- Send invoice.

Additional work on WARC in NAS is now subject to normal issue management and prioritization by netarkivet.dk.

Testing of NAS with WARC by Bnf and/or ONB will be scheduled for the final release of NAS with WARC.

**1.6 M6: JHove2 WARC, ARC and GZip modules v1.0 (Sep-2012)**Progress: **99.0%**

Actions:

- Thomas Ledoux/BnF has tested the different modules and only an issue with temporary files remains,
- Codefreeze around 5.10.2012

Tasks:

- Complete a more relaxed URI validation class.
- Copy unit tests from JWAT to JHove2.
- Prepare for codefreeze.
- Send invoice when JHove2 2.1.0 is released.

**1.7 M7 Workshop in Copenhagen/Aarhus on WARC/NAS tests (Sep-12)**Progress: **N/A**<https://sbforge.org/display/NAS/2012-October+workshop+at+SB>

Planning is underway and a date for the workshop has been chosen. (29-30 October)

**1.8 M8 Stable Release of NetarchiveSuite with WARC-support (Nov-12)**Progress: **N/A**<https://sbforge.org/jira/browse/NAS/fixforversion/10746>

NAS 4.0 - Prod release with WARC support.

**1.9 M9: Final project report (Nov-2012)**Progress: **5%**

Actions:

- Clément and Aaron had some ideas to what could be included.

Tasks

- Make an outline of the report content.
- Author report, included material from the wikis that will not be included on the NAS or JHove2 webpages.

## A JWAT

### A.1 JWAT packages (reusable!)

<https://sbforge.org/display/JWAT/JWAT>

<https://bitbucket.org/nclarkekb/jwat/>

Version 1.0.0 planned for the JHove2 codefreeze.

Implements all the actual *ARC*, *WARC* and *GZip* functionality.

Features:

- Common classes complete.
- GZip reader/validator/writer complete.
- ARC reader/validator/writer complete.
- WARC reader/validator/writer complete.
- Approx. 150kb jars and no external dependencies.

Alternative to the readers/writers in Heritrix.

### A.2 JWAT-Tools (recyclable)

<https://bitbucket.org/nclarkekb/jwat-tools/downloads/>

Handy command line utility which currently

- Validates GZip, ARC and WARC archives.
- Validates XML payload against DTD or XSD declarations (mets, etc.).
- Simple plugin system in progress.
- Decompresses \*.arc.gz, \*.warc.gz and \*.gz files.
- Compresses \*.warc and single files.
- Parallelized validation (threads configurable)
- ARC to WARC converter.

### A.3 JWAT-Tools-GUI

Application with a simple graphical user interface.

- Add WARC, ARC and GZip files to the work queue.
- Validate WARC, ARC and GZip files.
- Overview of queue with progress and result in a table.

## B JHove2 modules

Codefreeze planned for 5.10.2012.

### B.1 ARC/WARC format modules

- The *ARC* and *WARC* Format Modules should be complete.

### B.2 GZip format module

- The *GZip* Format Module should be complete.

### B.3 File identification module

- *File* Identification Module should be complete.

### B.4 XSL display module

- XSL Display Module should be complete.
- *containerMD.xsl* requires modification to work with the current JHove2 output format.



## **H   Status update 17 Apr. 2013**

## Project update

### 1 Milestones

Includes actions from last project update and open tasks.

For completeness this document now includes all milestones defined by the project and not only the major deliverables.

The following table lists each milestone and its overall status.

M	Date	Description	Status
M1	jan-12	Technical specification of WARC module for JHOVE2	100%
M2	mar-12	Prototype Code release of JHOVE2-modules	100%
M3	apr-12	Workshop in Copenhagen on WARC/NAS specifications	100%
M4	maj-12	Progress report at IIPC GA 2012, Washington D.C	100%
M5	aug-12	Developer Release of NetarchiveSuite with WARC-support	100%
M6	sep-12	Final Code release of JHOVE2-modules	100%
M7	sep-12	Workshop in Copenhagen/Aarhus on WARC/NAS tests	100%
M8	nov-12	Stable Release of NetarchiveSuite with WARC-support	100%
M9	nov-12	Final project report (and possible presentation/workshop attached to an IIPC event or workshop in the Fall)	20.0%

#### 1.1 Milestone 1, 2, 3, 4, 5, 7

Progress: **100.0%**

Should all have been approved by Program Officer and payments should have been made.

**1.2 M6: JHove2 WARC, ARC and GZip modules v1.0 (Sep-2012)**Progress: **100.0%**

Actions:

- JHove2.1.0 released: 2013-03-11

Approval and payment: Unknown.

**1.3 M8 Stable Release of NetarchiveSuite with WARC-support (Nov-12)**Progress: **100.0%**

Actions:

- NAS 4.0 released: 2013-01-28

Approval: Unknown.

**1.4 M9: Final project report (Nov-2012)**Progress: **20.0%**

Actions:

- Got input from Clément.

Tasks:

- Finish report ASAP!

## **I NAS workshop agenda and outcome (2012-04-02)**

4/17/13

NAS Warc workshop - NetarchiveSuite - SBForge



## NAS Warc workshop

Added by [Mikis Seth Sørensen](#), last edited by [Søren Veirup Carlsen](#) on Sep 26, 2012

Information on the BnF-Netarkivet.dk workshop at [KB](#) with the purpose of defining the WARC implementation work in NAS.

- Place: [KB](#)
- Time: April 2 09:15 to 13:00??.
- Participants:
  - BnF: Clément, Sara and Sophie
  - KB: Nicholas & Søren
  - SB: Mikis

### Agenda

- (1 hour) Recap on JHove2 module status.
  - Status for merge to HEAD of Nicholas's code.
- Martha is aware of the problems with merging 3rd party code to HEAD, and as the Jhove2 is a high priority from IIPC will hope this will be addressed before or under the GA in Washington.
- Status for JHove2 milestone, including demo.
  - A proposal for criterias for a validation of the prototype release, is that only the output of Jhove2 modules should be used (the code itself will be tested as the part of the road to the final release).
  - Clement will mail Aaron regarding payment for the initial technical specification.
  - BnF will test the JHove2 release in May, so we can get the first milestone validated.
  - As Nicholas has removed the Jhove2 ability to run in parallel, the performance aspects of Nicholas's code need to be tested and perhaps discussed at the GA.
  - Nicholas and Clement should have a technical discussion regarding Nicholas's code during the GA. Subjects here would be code merge to HEAD, parallelization. Perhaps Monday at 17:00.
  - KB will be using the Jhove2 WARC for digital document characterization as part of the preservation.
  - As WARC is currently used more for none-web archiving, Clement is very interested in input to extensions to the WARC ISO standard.
  - We talked a bit about the possibility of a PDF module which will be needed by BnF. Perhaps a job for Nicholas?
  - Nicholas will look at what it means to propose it as defaults extraction.
  - BnF will look at specific WARC extensions.
  - We should plan a NAs workshop in late august/early september. We should have finished the Jhove2 testing here and the NAS WARC functionality should be nearing completion. Nicholas current contract ends in mid september, so it shouldn't be any later (unless the contract is extended).
  - WARC module validation:
    1. BnF will send sample WARC files Nicholas can generate Jhove2 output for inspection by BnF. The WARC's should both be analyzed with 'File' and 'Droid'.
    2. Nicholas will then mail the basic code release to Aaron for testing (Tomas(BnF) and also by Steve?), so we can be prepared for input to the final release.
    3. BnF will merge Nicholas's code and test it, including performance test (parallelism).
    4. Any PWG feedback.
- (10 minutes) Discuss Jhonas presentation at GA : project update (10-15 presentation on Tuesday) + half day presentation at the PWG

4/17/13

NAS Warc workshop - NetarchiveSuite - SBForge

1. Short JhoNAS presentation
  - a. General presentation of the project (WARC, Jhove2, NAS), why, who.
  - b. Summary of current status.
  - c. Refere people to detailed sessions.
1. PWG workshop
  - \*# Nicholas will prepare an agenda in cooperation with Clément. The agenda should be sent to IIPC as soon as possible so it can be posted to the GA web.
    1. #More detailed breakdown of what is extracted by the Jhove2 WARC module.
      - More detailed walkthrough of the metadata model which will be used in NetarchiveSuite (including ARC-WARC mapping) and the metadata is handled in general in NAS.
      - Demo of the module.
      - A priority here is to expose the value of this project for 3rd parties and listen to ideas for additional features.
    2. Nicholas will prepare an presentation in cooperation with Clément.
- (30 minutes) Discussion about NetarchiveSuite workshop at IIPC GA.
  - We should consider breaking the last part into a discussion track and a demo/handons. Annick and Nicholas might handle the demo/handons part.
  - We should consider sending a mail to participants with a update on the agenda and request information regarding the expectations for the workshop (and confirm their participation). Sara will request a list of participants from Abbie.
- (1 hour) Review of the currently defined tasks: [NAS-1720@jira](mailto:NAS-1720@jira).
  - Comments added to issues.
  - BnF would like to be able to define custom identifiers for WARC.
- (Afternoon) Leveraging the WARC formats possibility for adding metadata.
  - Define initial metadata model.
- Mapping of NetarchiveSuite metadata with WARC warcinfo, metadata and named files.
  - Initial mapping defined. Can be found at the bottom of the page (attachement).
  - Clement (and Sophie and Sara) will write up the proposed WARC format specification and send it to the participants.
  - Nicholas will create a specification wiki page based on this. It will be recommended to the participants to subscribe to changes to this page.
  - The additional NAS functionality need to support the extended format (harvest info metadata, configurable file name format, etc.) will be defined by Nicholas (assisted by Søren and Mikis).

## WARC in NAS format draft

### Implementation of the WARC format into NetarchiveSuite: mapping and requirements

Updated draft after the Jhonas Workshop in Copenhagen , April 2-3 2012

#### WARC data files

##### *WARC warcinfo record*

Do not change Heritrix output, but insert the following records:

##### *WARC metadata record containing harvestinfo.xml*

WARC/1.0

WARC-Type: metadata

<WARC-Target-URI: none, written just to clarify>

WARC-Date: [Same as the warcinfo record]

WARC-Record-ID: [UID of the record]

WARC-Concurrent-To: [UID of the warcinfo record]

WARC-Warcinfo-ID: [UID of the warcinfo record]

Content-Type: application/xml (or text/xml?)

WARC-Block-Digest: [digest]

Content-Length: [content size of the block]

4/17/13

NAS Warc workshop - NetarchiveSuite - SBForge

[Insert here harvestinfo.xml]

*WARC metadata record containing ONB extended fields (if any configured at the harvest level)*

WARC/1.0

WARC-Type: metadata

<WARC-Target-URI: none, written just to clarify>

WARC-Date: [Same as the warcinfo record]

WARC-Record-ID: [UID of the record]

WARC-Concurrent-To: [UID of the warcinfo record]

WARC-Warcinfo-ID: [UID of the warcinfo record]

Content-Type: application/xml (or text/xml?)

WARC-Block-Digest: [digest]

Content-Length: [content size of the block]

[Insert here the extended fields – decide wich format?]

*WARC response records*

Do not change Heritrix output.

Note: there are some issues related to Heritrix development:

- absence of WARC-warcinfo-ID field for each response record,
- check the possibility of changing the UID system.

WARC metadata file

*WARC Warcinfo record*

WARC/1.0

WARC-Type: warcinfo

WARC-Date: [Date of creation of the record]

WARC-Record-ID: [UID of the record]

WARC-Filename: [Name of the WARC metadata file]

Content-Type: application/warc-fields

WARC-Block-Digest: [digest]

Content-Length: [content size of the block]

software: NetarchiveSuite/[version]/http://netarchive.dk/suite/

ip: [ip of the harvest server]

hostname: [full name of the harvest server]

conformsTo: [declare a specific metadata profile]

isPartOf: [name or number of the job]

*WARC metadata record containing harvestinfo.xml*

WARC/1.0

WARC-Type: metadata

<WARC-Target-URI: none, written just to clarify>

WARC-Date: [Same as the warcinfo record]

WARC-Record-ID: [UID of the record]

WARC-Concurrent-To: [UID of the warcinfo record]

WARC-Warcinfo-ID: [UID of the warcinfo record]

Content-Type: application/xml (or text/xml?)

WARC-Block-Digest: [digest]

Content-Length: [content size of the block]

[Insert here harvestinfo.xml]

*WARC metadata record containing ONB extended fields (if any configured at the harvest level)*

WARC/1.0

WARC-Type: metadata

<WARC-Target-URI: none, written just to clarify>

WARC-Date: [Same as the warcinfo record]

WARC-Record-ID: [UID of the record]

WARC-Concurrent-To: [UID of the warcinfo record]

<https://sbforge.org/display/NAS/NAS+Warc+workshop>

3/5

4/17/13

NAS Warc workshop - NetarchiveSuite - SBForge

WARC-Warcinfo-ID: [UID of the warcinfo record]  
Content-Type: application/xml (or text/xml?)  
WARC-Block-Digest: [digest]  
Content-Length: [content size of the block]

[Insert here the extended fields]

*WARC resource records (for configurations, logs and reports)*

WARC/1.0  
WARC-Type: resource  
WARC-Target-URI: [file://hostname/HarvestID/JobID/crawl/logs| setup |reports/filename  
Example: file://netarchivesuite.bnf.fr/HarvestID\_23/JobID\_345/crawl/logs/crawl.log]  
WARC-Date: [Date of creation of the record]  
WARC-Block-Digest: [digest]  
WARC-IP-Address: 207.241.229.39 [ip of the harvest server]  
WARC-Record-ID: [UID of the record]  
WARC-Warcinfo-ID: [UID of the warcinfo record]  
Content-Type: [text/plain or application/xml|text/xml or anything else if needed?]  
Content-Length: [content size of the block]

[The file recorded here]

*WARC metadata record describing the source of each resource record*

WARC/1.0  
WARC-Type: metadata  
WARC-Target-URI: [URI of the related record]  
WARC-Date: [Same as the related record]  
WARC-Record-ID: [UID of the record]  
WARC-Concurrent-To: [UID of the related record]  
WARC-Warcinfo-ID: [UID of the warcinfo record]  
Content-Type: text/plain  
WARC-Block-Digest: [digest]  
Content-Length: [content size of the block]

[Insert here the information about the creation tool of the related resource record. Example: CreatedBy: Heritrix 1.14.2]

#### Fields to be added in harvestinfo.xml and corresponding XSD schema

jobSubmitDate: Launch date of the job (as calculated by NAS)  
performer: exact term needs to be checked - Harvesting organization. Depends on the installation/configuration.  
Audience: organization or person for which the harvest is intended for. Depends on the harvest. Needs a specific field to be added at the harvest level.  
template: name of the harvest template.

#### Note on WARC file naming

According to the standard, annex C (informative):  
Naming should follow the scheme: Prefix-Timestamp-Serial-Crawlhost.warc.gz

Examples given in the guidelines:  
BNF-CRAWL-003-20071013163428-02639-crawling04.us.archive.org.arc.gz  
NLNZ-TI1179651-20091006011055-00000-kaiwae-z11.arc

Proposal for NAS data files:

[Name of harvesting organization, depends on the NAS installation]-[configurable prefix containing no dash]-[Harvest definition number]-[job number]-[Timestamp]-[Serial]-[Crawlhost].warc.gz

Proposal for NAS, metadata files:

[Name of harvesting organization, depends on the NAS installation]-[configurable prefix containing no dash]-metadata-[Harvest definition number]-[job number]-[Timestamp]-[Serial]-[Crawlhost].warc.gz

Example of data file : BnF-elec2012-24-345- 20120331123243-001234-gulliver101.warc.gz



4/17/13

NAS Warc workshop - NetarchiveSuite - SBForge

Example of metadata file : BnF-elec2012-24-345-metadata-20120331154452-001234-gulliver101.warc.gz

Question: should we just give the code of harvesting organization in the name of the file, constraining ourselves to use one dash only?

---

None

---

## **J NAS 3.21.0 release notes (Developer release)**



## NetarchiveSuite 3.21.0 Release Notes

Added by [Mikis Seth Sørensen](#), last edited by [Søren Vejrup Carlsen](#) on Oct 16, 2012

Planned release 5.9.2012.

- [Highlights](#)
- [Upgrade-notes](#)
  - [Harvester DB](#)
- [Full list of issues resolved in this release.](#)

[Known issues](#)

### Highlights

Support for WARC harvesting, processing and access. To enable WARC-writing in NAS, which is by default disabled, you need to do the following:

- Override in your deployment configuration "settings.settings.harvester.harvesting.metadata.metadataFormat" with "warc",
- Override in your deployment configuration "settings.harvester.harvesting.heritrix.archiveFormat" with "warc".
- Make sure that the templates, you are using, contains the WARCWriterProcessor. On how to do this, see [NAS-1958](#). You can just add the WARCWriterProcessor. You don't need the remove ARCWriterProcessor. If the ARCWriterProcessor exists in the template, this processor will just be disabled when Netarchivesuite runs Heritrix in warc-mode.

### Upgrade-notes

#### Harvester DB

- The `creationdate` field has been added to the Job table.

To update the databases use the `dk.netarkivet.harvester.tools.HarvestdatabaseUpdateApplication` (See Additional Tools Manual).

### Full list of issues resolved in this release.







### [JIRA Issues \(15 issues\)](#)

Type	Key	Priority	Summary
	<a href="#">NAS-2110</a>		<a href="#">Wayback Indexer fails to catch Unchecked Exception</a>
	<a href="#">NAS-1720</a>		<a href="#">Enable WARC file writing and handling in the NetarchiveSuite</a>
	<a href="#">NAS-1965</a>		<a href="#">Make it possible to use either ARC or WARC as the harvesting format.</a>
	<a href="#">NAS-1959</a>		<a href="#">Implement CDX-generating code, that also works for WARC-files</a>
	<a href="#">NAS-1960</a>		<a href="#">Extend our BatchJob framework to handle WARC-files on record level</a>
	<a href="#">NAS-1964</a>		<a href="#">Upgrade of Indexserver system</a>
	<a href="#">NAS-1351</a>		<a href="#">it-conf-example.xml may be slightly confusing</a>
	<a href="#">NAS-2103</a>		<a href="#">QA-scripts doesn't work with WARC metadatafile</a>
	<a href="#">NAS-1962</a>		<a href="#">Store the contents of the metadata-1.arc files as WARC-records</a>
	<a href="#">NAS-2061</a>		<a href="#">Define the layout of the metadata warc file</a>
	<a href="#">NAS-2033</a>		<a href="#">Introduce Scheduling Time attribute on HarvestJobs</a>
	<a href="#">NAS-2018</a>		<a href="#">Twitter extractor module for Heritrix</a>





[Download](#)

[Manuals](#)

[Javadoc](#)

	<a href="#">NAS-2063</a>		<a href="#">No system state information about Starting to create jobs in harvestjobManager</a>
	<a href="#">NAS-2094</a>		<a href="#">deploy should remove old libs before install</a>
	<a href="#">NAS-2087</a>		<a href="#">Wrong text in Danish I18N key on bitpreservation page</a>

#### Known issues

<a href="#">JIRA Issues</a> <input type="checkbox"/>			
Type	Key	Priority	Summary
	<a href="#">NAS-2116</a>		<a href="#">Wayback index handling of archive doublets</a>
	<a href="#">NAS-2109</a>		<a href="#">metadata://netarkivet.dk/crawl/reports/arcfiles-report.txt is empty when Heritrix set to WARC writing</a>
Page 1 of 1      Displaying 1 to 2 of 2 items			

None

## K JHOVE2 2.1.0 release notes

Stanford University LIBRARIES &  
ACADEMIC INFORMATION RESOURCES

## JHOVE2 – Next-Generation Framework and Application for Format-Aware Characterization

Version: 2.1.0

Issued: 2013-02-14

Status: Final

### Release Notes

#### Version 2.1.0

Version 2.1.0 of JHOVE2 includes 3 new format modules, 1 new Identifier module, and several bug fixes and enhancements from the Issues page on the JHOVE2 wiki (<https://bitbucket.org/jhove2/main/issues>).

New format modules included in this release:

- ARC
- GZIP
- WARC

This release includes a new identifier module, based on the Unix "file" utility. The downloadable release is configured to run the DROID identifier that was released in version 2.0.0.

For information on how to install the "file" utility on Windows, MAC, and Unix machines, and for information on how to update the JHOVE2 Spring configuration files to employ the new Identifier module, please see the "Specification and Installation/Configuration Guide" for the File Identifier Module on the JHOVE2 wiki modules page (<https://bitbucket.org/jhove2/main/wiki/documents/JHOVE2-File-module-spec-2.1.0RC2.pdf>).

Resolved issues included in this release:

- #56: Review Laurent Bihanic's Gzip code
- #125: opensp tests fail on ubuntu
- #126: 0 tag IFD error message
- #128: jargs jar has moved to a different Maven Repository -- pom.xml must be updated
- #130: Have BerkeleyDB je persistence database use user home directory by default
- #132: Tool to confirm that all Messages are represented in jhove2\_messages.properties file
- #134: duplication of the Formatmodule output takes place when using the in-Memory Persistence Manager.
- #136: Windows driver script doesn't work outside of home directory
- #140: Incorrect "PostScript" name for "PDF" in "OtherFormats-config.xml"
- #143: Error message for org.jhove2.module.format.tiff.IFDEntry.InvalidCountValueMessage is missing in jhove2\_messages.properties file

Release Notes

Page 1 of 4

Stanford University LIBRARIES &  
ACADEMIC INFORMATION RESOURCES

- #146: Typo in droid signaturefile
- #147: WARC Droid Signature definition
- #148: Bug in InMemorySourceAccessor/InMemoryBaseModuleAccessor/...
- #153: Tiff Module never reports Validity.True
- #155: Problems with spaces and hyphens in file paths
- #156: Create GZip format module
- #157: Create ARC format module
- #158: Create WARC format module
- #160: org.jhove2.module.format.wave.bwf.LinkChunk missing zero-arg constructor
- #161: org.jhove2.config.spring.SpringConfigInfo must make CLASSPATH for Spring context configurable
- #162: Message  
org.jhove2.module.format.sgml.OpenSpWrapper.IOExceptionForSGMLStdErrFile2 in Java code is not in messages properties files
- #163: spring-test-2.5.6.jar is not included in the download zip file
- #165: TiffTagTest and ICCModuleTestBase need setUpBeforeClass() overrides
- #166: Update MessagesChecker tool to read in more than one .properties file
- #167: Wrong URL for OPenSp windows binary download in User Guide
- #168: Need documentation for new GZIP module
- #169: Need documentation for new ARC module
- #170: Need documentation for new WARC module
- #171: Document new File identifier
- #172: New BSD File -based identifier
- #173: create displayer properties file for Arc module
- #174: Create displayer properties file for gzip module
- #175: Create displayer properties file for WARC module
- #176: Update user's guide to refer to configuration info for File-based identifier

For information about issues resolved in this release, known bugs, open issues, and enhancement requests, please refer to

- ***JHOVE2 Issues page***

<https://bitbucket.org/jhove2/main/issues?sort=version>

For detailed installation and configuration instructions please refer to:

- ***JHOVE2 User's Guide***

[http://bitbucket.org/jhove2/main/wiki/documents/JHOVE2-Users-Guide\\_20110222.pdf](http://bitbucket.org/jhove2/main/wiki/documents/JHOVE2-Users-Guide_20110222.pdf).

For detailed guidance on developing additional format modules please refer to:

- ***JHOVE2 Architectural Overview***

<http://bitbucket.org/jhove2/main/wiki/documents/JHOVE2-Architecture-v2-0-0.pdf>

Release Notes

Page 2 of 4



- **JHOVE2 Programmer's Guide**

<http://bitbucket.org/jhove2/main/wiki/documents/JHOVE2Programmer2-0-0.pdf>

Questions concerning the use of JHOVE2 and module development should be addressed to [JHOVE2-TechTalk-l@listserv.ucop.edu](mailto:JHOVE2-TechTalk-l@listserv.ucop.edu).

Specific errors or suggestions may be reported to the JHOVE2 issue tracker at <https://bitbucket.org/jhove2/main/issues?sort=id>.

#### CALIFORNIA DIGITAL LIBRARY

Stephen Abrams  
Patricia Cruse  
John Kunze  
Marisa Strong  
Perry Willett

#### PORTICO

John Meyer  
Sheila Morrissey

#### STANFORD UNIVERSITY

Richard Anderson  
Tom Cramer  
Hannah Frost

#### BIBLIOTHÈQUE NATIONALE DE FRANCE

Laurent Bihanic

#### NETARKIVET.DK

Nicholas Clarke



Stanford University LIBRARIES &  
ACADEMIC INFORMATION RESOURCES

## Version 2.0.0

JHOVE2 is a next-generation framework and application for format-aware characterization. Characterization is the process of deriving *representation information* about a formatted digital object that is indicative of its significant nature and is useful for purposes of classification, analysis, and use. Effective and efficient means of characterization is a key component of any digital preservation program.

JHOVE2 supports four specific aspects of characterization:

- *Identification*. The process of determining the *presumptive* format of a digital object on the basis of suggestive extrinsic hints and intrinsic signatures, both internal (e.g. magic number) and external (e.g. file extension).
- *Validation*. The process of determining the level of *conformance* to the normative syntactic and semantic rules defined by the authoritative specification of the object's format.
- *Feature extraction*. The process of reporting the *intrinsic properties* of a digital object significant for purposes of classification, analysis, and use.
- *Assessment*. The process of determining the level of *acceptability* of a digital object for a specific purpose on the basis of locally-defined policy rules.

The object of JHOVE2 characterization can be a file, a subset of a file, or an aggregation of an arbitrary number of files that collectively represent a single coherent digital object. JHOVE2 can automatically process objects that are arbitrarily nested in containers, such as file system directories or Zip files.

The JHOVE2 project seeks to build on the success of the original JHOVE characterization tool (<http://hul.harvard.edu/jhove>) by addressing known limitations and offering significant new functions. These enhancements include:

- Streamlined APIs incorporating increased modularization and uniform design patterns.
- Object-focused, rather than file-focused, characterization, with support for arbitrarily-nested container formats and formats instantiated across multiple files.
- Signature-based identification using DROID (<http://sourceforge.net/projects/droid>).
- Rules-based assessment to support determinations of object *acceptability* in addition to validation of format *conformity*.
- Extensive user configuration of modules, characterization strategies, localized messages, and formatted results.
- Performance improvements using Java buffered I/O (java.nio).
- Persistence manager to support the characterization of an arbitrary number of objects with a fixed memory footprint.

Release Notes

Page 4 of 4



The JHOVE2 project is a collaborative undertaking of the University of California Curation Center at the California Digital Library, Portico, and Stanford University, with generous funding from the Library of Congress as part of its National Digital Information Infrastructure and Preservation Program (NDIIPP).

JHOVE2 is made freely available under the terms of the BSD open source license for all project-developed code; some third-party libraries may be covered by other open source licenses.

<http://jhove2.org/>

[JHOVE2-Announce-l@listserv.ucop.edu](mailto:JHOVE2-Announce-l@listserv.ucop.edu)

[JHOVE2-Techtalk-l@listserv.ucop.edu](mailto:JHOVE2-Techtalk-l@listserv.ucop.edu)

Version 2.0.0 of JHOVE2 supports all the major technical objectives of the project, including a more sophisticated modular architecture; signature-based file identification; policy-based assessment of objects; recursive characterization of objects comprising aggregate files and files arbitrarily-nested in containers; and extensive configuration and reporting options. It provides a stable interface against which developers can create additional format modules.

Format modules, and profiles, included in this release are:

- ICC color profile
- SGML
- Shapefile *Main, Index, dBASE*
- TIFF *4 – 6, Class B, G, R, P, Y, TIFF/IT, TIFF/EP, Exif, GeoTIFF, DNG*
- UTF-8 *ASCII*
- WAVE *Broadcast Wave Format*
- XML
- Zip

Please note that the Zip module comprises a non-validating partial module, which accomplishes recursive JHOVE2 descent on the contents of the Zip file, but does not yet validate the Zip file itself against the standard.

Stanford University LIBRARIES &  
ACADEMIC INFORMATION RESOURCES

Version 2.0.0 of JHOVE2 can be downloaded from <https://bitbucket.org/jhove2/main/downloads>.  
Download packages are available in Zip and tar.gz form.

For information about issues resolved in this release, known bugs, open issues, and enhancement requests, please refer to

- **JHOVE2 Issues page**

<https://bitbucket.org/jhove2/main/issues?sort=version>

For detailed installation and configuration instructions please refer to:

- **JHOVE2 User's Guide**

[http://bitbucket.org/jhove2/main/wiki/documents/JHOVE2-Users-Guide\\_20110222.pdf](http://bitbucket.org/jhove2/main/wiki/documents/JHOVE2-Users-Guide_20110222.pdf).

For detailed guidance on developing additional format modules please refer to:

- **JHOVE2 Architectural Overview**

<http://bitbucket.org/jhove2/main/wiki/documents/JHOVE2-Architecture-v2-0-0.pdf>

- **JHOVE2 Programmer's Guide**

<http://bitbucket.org/jhove2/main/wiki/documents/JHOVE2Programmer2-0-0.pdf>

Questions concerning the use of JHOVE2 and module development should be addressed to [JHOVE2-TechTalk-l@listserv.ucop.edu](mailto:JHOVE2-TechTalk-l@listserv.ucop.edu).

Specific errors or suggestions may be reported to the JHOVE2 issue tracker at <https://bitbucket.org/jhove2/main/issues?sort=id>.

## Development planning

Additional JHOVE2 functionality is scheduled for inclusion in subsequent releases:

- Version 2.1.0
  - ARC and Gzip modules  
(integration of third-party development by Bibliothèque nationale de France / Atos)
  - Grid and NetCDF modules  
(integration of third-party development by Wegener Institute for Polar and Marine Research)
  - JPEG 2000 module
- Version 2.2.0
  - PDF module

Stanford University LIBRARIES &  
ACADEMIC INFORMATION RESOURCES**JHOVE2 project team***California Digital Library*

Stephen Abrams  
Patricia Cruse  
John Kunze  
Isaac Rabinovitch  
Marisa Strong  
Perry Willett

*Portico*

John Meyer  
Sheila Morrissey

*Stanford University*

Richard Anderson  
Tom Cramer  
Hannah Frost

*Library of Congress*

Martha Anderson  
Justin Littman

*With help from*

Walter Henry  
Nancy Hoebelheinrich  
Keith Johnson  
Evan Owens

## **L JHOVE2 WARC module specifications**



## JHOVE2: Next-Generation Architecture for Format-Aware Characterization WARC Module

Version 2.1.0

Issued 2012-12-03

Status Draft

### 1 Introduction

JHOVE2 is a framework and application for next-generation format-aware characterization of digital objects. The function of JHOVE2 is encapsulated in a series of modules that can be configured for use within the framework's plug-in architecture. The WARC module provides characterization services for the WARC format.

#### **Important information for users of the JHOVE2 WARC module**

The authoritative specification for WARC [WARC] is *unambiguous*.

Validation of WARC instances by this module is *comprehensive*.

**NOTE** A format specification is considered *unambiguous* if there is broad community consensus regarding the intention of *all* normative requirements of the format's authoritative specification; otherwise it is considered *ambiguous*, and areas of potential ambiguity will be documented below.

Module validation is considered *comprehensive* if *all* normative requirements defined by that specification are validated by the module; otherwise it is considered *selective*, and non-validated features will be documented below.

### 2 Identification

<b>Primary format or format family</b>	
<b>Canonical format name:</b>	warc
<b>Alias format name(s):</b>	warc
<b>Canonical format identifier:</b>	JHOVE2 <a href="http://jhove2.org/terms/format/warc">http://jhove2.org/terms/format/warc</a>
<b>Alias format identifier(s):</b>	PRONOM PUID: fmt/289
	MIME application/warc

<b>JHOVE2 WARC module</b>	
<b>JHOVE2 module name:</b>	WarcModule
<b>JHOVE2 module identifier:</b>	JHOVE2 <a href="http://jhove2.org/terms/reportable/org/jhove2/module/format/warc/WarcModule">http://jhove2.org/terms/reportable/org/jhove2/module/format/warc/WarcModule</a>
<b>JHOVE2 module class</b>	org.jhove2.module.format.warc.WarcModule.java org.jhove2.module.format.warc.WarcModule.class
<b>JHOVE2 module jar</b>	



WARC File or stream Signature		
File format	Jhove2 Profile	File Header(s) Signature(s)
warc	warc	WARC /

### 3 References

For the purposes of the JHOVE2 WARC module the authoritative format specifications are:

[WARC] ISO 28500:2009  
[http://www.iso.org/iso/catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=44717](http://www.iso.org/iso/catalogue/catalogue_tc/catalogue_detail.htm?csnumber=44717)

Draft version: <http://bibnum.bnf.fr/WARC/index.html>

For IIPC members only:  
<http://www.netpreserve.org/forum/viewtopic.php?f=70&t=386>

#### Other Useful References:

[Heritrix] Internet Archive's web crawler  
<http://crawler.archive.org/>

[WARCWriter] Java module that writes WARC files  
<http://crawler.archive.org/apidocs/org/archive/io/warc/WARCWriter.html>

[Include specification of ARC, GZIP module + File identification when they will be available online]

[RFC2616] Hypertext Transfer Protocol -- HTTP/1.1  
<http://tools.ietf.org/id/draft-ietf-http-v11-spec-rev-06.txt>

[RFC1945] Hypertext Transfer Protocol -- HTTP/1.0  
[http://datatracker.ietf.org/doc/rfc1945/?include\\_text=1](http://datatracker.ietf.org/doc/rfc1945/?include_text=1)

### 4 Validity

#### 4.1 General

A WARC file consists of one or more WARC records. To be considered a valid WARC file every record in the file must be valid. To adhere to the standard a valid WARC record shall contain all mandatory headers, shall not contain any invalid headers and may or may not contain any recommended and/or optional headers. These requirements are defined in the standard for each type of record.

Please refer to the standard for a complete definition of WARC validity.



## 4.2 Format versions

JHOVE2 treats the WARC format as a family having several versions.

Current valid versions are 0.17, 0.18 and 1.0. This list may evolve through the ISO periodical revision process (next one will occur in 2012).

A WARC file is still considered valid even if the version differs across the records.

## 4.3 Validation implemented

In order to ensure the validity of a WARC file the module reads the whole file sequentially from beginning to end looking for records to validate. The module will only report a valid WARC file if this process does not encounter any problems warranting errors or warnings.

Should the module be unable to read the entire file because of a problem (runtime exception), the validity of the WARC file is undetermined until the module is corrected or the WARC file validated by other means. Problems with the underlying file system can result in the reader not being able to validate the whole file.

Errors/warnings are reported on a file or record level. Normally errors/warnings are reported in the offending record. In case there is no current record to attach errors/warnings to, they are reported in the reader.

So if the module is reading a non WARC file it will most likely not report any records, instead errors/warnings will be reported in the reader and the file will be considered invalid. Similarly any garbage after WARC records will not return a record but will still report errors/warnings in the reader and the WARC file will be considered invalid. Any garbage in front of a record will not be reported in the reader but will trigger errors/warnings on the record level since it is in the beginning of the record.

Records with incorrect content-length values will be handled according to the situation. A content-length which is different from the actual payload will result in garbage errors/warnings in the preceding record or the reader. If the length is lesser the next record will be read and returned. On the other hand, if the length is greater, the next record header will not be read and instead result in errors/warnings being reported in the next record found or the reader if no more records are found.

In cases where there is a known payload, like an http response/request, errors/warnings are reported if the expected length of the payload is different from the one stated in the WARC header.

### 4.3.1 Reported version errors

No errors are reported in case the version is either "0.17", "0.18" or "1.0".

The following situations report errors.

- Any other string will report "**Unknown magic version number**".
- A version string consisting of integers delimited by a "." but with more or less than 2 parts will report "**Invalid magic version string**". (fx. <x> or <x>.<x>.<x>.<x>)
- Error "**Invalid data before WARC version**": if there is data in front of the version line.
- Error "**Invalid empty lines before WARC version**": if there are empty lines in front of the record.

### 4.3.2 Reported WARC record errors

The following errors can be reported for a WARC record.



## JHOVE2

- Error “**Unable for parse HTTPHeader**”: if for some reason the HTTP Header is malformed.
- Error “**Payload length mismatch**”: in case the payload was truncated.
- Error “**Invalid expected 2 trailing newlines**”: if there are more or less than 2 newlines.

### 4.3.3 HeaderLine reader errors

The (header)line reader used to read WARC and HTTP headers reads and validates lines according to the ISO standard. The ISO standard includes references to several RFCs which the (header)line reader also adheres to. The reader will validate raw, us-ascii, iso8859-1 and utf-8 characters. It will also validate “quoted strings” and “encoded words”.

The following errors/warnings can be returned from the (header)line reader.

- Error “**Unexpected EOF**”: if a (header)line is not completed by a LF but an EOF.
- Error “**Misplaced CR**”: if a CR is not preceded by a LF.
- Error “**Missing CR**”: if the reader is expecting CRLFs and only encounters a LF.
- Error “**Excessive CR**”: if the reader is expecting LFs and encounters a CRLF.
- Error “**Invalid UTF-8 encoding**”: in case an encoded character can not be decoded according to the UTF-8 specification.
- Error “**Invalid US-ASCII character**”: basically if the character is not 7-bit.
- Error “**Invalid control character**”: if the character is between 0-31, with some exceptions.
- Error “**Invalid separator character**”: if a separator character appears where it is not supposed to be, like in the header name.
- Error “**Missing quote**”: if a quoted string ends without a quotation mark.
- Error “**Missing quoted pair character**”: if a quoted string ends with a backslash but with no quoted pair character.

### 4.3.4 Reported field parsing errors/warnings

While reading the WARC header the following errors/warnings can be reported before the values are validated.

- Warning “**Empty header line**”: if the header name is empty.
- Warning “**Unknown header line**”: if there is no colon separator.
- Warning “**Invalid header line**”: if there are no new lines or the line is mangled.
- Error: “**Duplicate header**”: in case the same WARC header occurs more than once, with the exception of the “Warc-Concurrent-To” header.

### 4.3.5 Reported field format errors/warnings

The following header value formats are validated.

- String
- Integer
- Long
- WARC date
- IP address
- URI
- Content-Type
- WARC Digest

The following errors/warnings can be reported.

- Warning “**Empty field**”: if a value is empty.
- Error “**Invalid numeric format**”: if the value is not a valid integer or long.
- Error “**Invalid WARC date format**”: if the date format does not conform to the standard.
- Error “**Invalid IPv4 or IPv6 format**”: if the value is not a valid IP.



- Error **“Invalid URI format”**: if the value is not a valid URI according to the URI profile used to validate.
- Error **“Invalid relative URI”**: if the URI is not absolute.
- Error **“Invalid Content-Type format”**: if the value is valid according to rfc2616.
- Error **“Invalid WARC digest format”**: if the digest format does not conform to the standard.

#### 4.3.6 Reported WARC header errors/warnings

After the WARC header has been read all the fields are validated according to the “Warc-Type”.

The following errors/warnings can be reported.

- Warning **“Unknown Warc-Type”**: in case of an undefined Warc-Type.
- Warning **“Unknown Warc-Profile”**: in case of an undefined Warc-Profile.
- Error **“Required Warc-Type missing”**: mandatory Warc-Type missing.
- Error **“Required Warc-Record-Id missing”**: mandatory Warc-Record-Id missing
- Error **“Required Warc-Date missing”**: mandatory Warc-Date missing
- Error **“Required Content-Length missing”**: mandatory Content-Length missing.
- Warning **“Recommended Content-Type missing”**: recommended if content-length <> 0.
- Warning **“Recommended Content-Type ‘application/warc-fields’”**: recommended content-type of a “WarcInfo” record.
- Error **“Invalid Warc-Segment-Number”**: if the segment number is not 1 and the record is not of the “Continuation” type or if the segment number is 1 and the record is of the “Continuation” type.

Furthermore each record type validates which fields are mandatory and which are not. If case of fields which presence is required or undesired the following errors are reported.

- Error **“Required <field> missing”**: in case a field is not present which must be.
- Error **“Undesired <field> present”**: in case a field is not suitable for the record type.

#### 4.3.7 Reported digest errors

- Error **“Unknown block/payload digest encoding scheme”**: in case the encoding used in the “Warc-Block/Payload-Digest” cannot be auto-detected.
- Error **“Invalid block/payload digest does not match”**: in case the computed digest does not match the on in the header.
- Error **“Invalid block/payload digest encoding scheme”**: in case an unknown encoding scheme was requested and none is present in the WARC header.

## 5 Reportable properties

### 5.1 WarcModule properties

The following table lists some of the properties which are reported by the WarcModule.

Each WarcModule present in the output is the result of processing a single WARC file.

WarcModule	
<b>Property</b>	WarcRecordNumber
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/warc/WarcModule/WarcRecordNumber
<b>Type</b>	Integer
<b>Description</b>	Number of WARC records



<b>Property</b>	WarcFileName
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/warc/WarcModule/WarcFileName
<b>Type</b>	String
<b>Description</b>	WARC file name
<b>Property</b>	WarcFileSize
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/warc/WarcModule/WarcFileSize
<b>Type</b>	Long
<b>Description</b>	WARC file size
<b>Property</b>	Errors
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/warc/WarcModule/Errors
<b>Type</b>	Map<String,Integer>
<b>Description</b>	The number of errors by error type
<b>Property</b>	File version
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/warc/WarcModule/ WarcFileVersion
<b>Type</b>	String
<b>Description</b>	There is a file version if all records of the same file share the same WARC version. Otherwise the version is undefined.

Validator (WarcModule)	
<b>Property</b>	isValid
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/Validator/isValid
<b>Type</b>	Validator\$Validity
<b>Description</b>	Validation status.
<b>Property</b>	Coverage
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/Validator/Coverage
<b>Type</b>	Validator\$Coverage
<b>Description</b>	Format module validation coverage.

## 5.2 WarcRecordSource properties

In JHove2 terms each record found inside a WARC file is represented as a **WarcRecordSource**. A **WarcRecordSource** always contains a set of **WarcRecordBaseProperties** which represents all the common properties irrespective of the record type. In addition to the base properties each **WarcRecordSource** also include properties which are relative to the record type. These are defined here as **Warc<WarcType>Properties**.

WarcRecordBaseProperties	
<b>Property</b>	WARC-Date
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcDate
<b>Type</b>	String (w3c-iso8601)
<b>Description</b>	WARC record archive date
<b>Property</b>	WARC-Record-ID
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcRecordID
<b>Type</b>	String
<b>Description</b>	ID of the WARC record

# JHOVE2

<b>Property</b>	Content-Type
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/ContentType">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/ContentType</a>
<b>Type</b>	String
<b>Description</b>	WARC record content type (not payload content type)
<b>Property</b>	Content-Length
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/ContentLength">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/ContentLength</a>
<b>Type</b>	Long
<b>Description</b>	Number of octets in the block
<b>Property</b>	WARC-Type
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcType">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcType</a>
<b>Type</b>	String
<b>Description</b>	Type of the WARC record
<b>Property</b>	WARC-Block-Digest
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcBlockDigest">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcBlockDigest</a>
<b>Type</b>	String
<b>Description</b>	WARC record block checksum
<b>Property</b>	Block-Digest-Algorithm
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcBlockDigestAlgorithm">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcBlockDigestAlgorithm</a>
<b>Type</b>	String
<b>Description</b>	Digest algorithm for the block
<b>Property</b>	Block-Digest-Encoding
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcBlockDigestEncoding">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcBlockDigestEncoding</a>
<b>Type</b>	String
<b>Description</b>	Digest encoding for the block (for example base 32, base 64...)
<b>Property</b>	isBlockDigestValid
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/isBlockDigestValid">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/isBlockDigestValid</a>
<b>Type</b>	Boolean
<b>Description</b>	If present, indicates whether the block digest was tested successfully or not
<b>Property</b>	WARC-Payload-Digest
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcPayloadDigest">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcPayloadDigest</a>
<b>Type</b>	String
<b>Description</b>	WARC record payload checksum (if any)
<b>Property</b>	Payload-Digest-Algorithm
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcPayloadDigestAlgorithm">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcPayloadDigestAlgorithm</a>
<b>Type</b>	String
<b>Description</b>	Digest algorithm for the payload
<b>Property</b>	Payload-Digest-Encoding
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcPayloadDigestEncoding">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcPayloadDigestEncoding</a>
<b>Type</b>	String
<b>Description</b>	Digest encoding for the payload (for example base 32, base 64...)
<b>Property</b>	isPayloadDigestValid
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/isPayloadDigestValid">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/isPayloadDigestValid</a>
<b>Type</b>	Boolean
<b>Description</b>	If present, indicates whether the payload digest was tested successfully or not



<b>Property</b>	WARC-Truncated
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcTruncated">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcTruncated</a>
<b>Type</b>	String
<b>Description</b>	Reason for the truncation of the WARC record
<b>Property</b>	hasPayload
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/hasPayload">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/hasPayload</a>
<b>Type</b>	Boolean
<b>Description</b>	Specifies whether the WARC record contains a payload
<b>Property</b>	PayloadLength
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/PayloadLength">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/PayloadLength</a>
<b>Type</b>	Long
<b>Description</b>	WARC record payload length (if any), in bytes
<b>Property</b>	WARC-Identified-Payload-Type
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcIdentifiedPayloadType">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcIdentifiedPayloadType</a>
<b>Type</b>	String
<b>Description</b>	Reliable Content type of the payload of the WARC record (if any)
<b>Property</b>	WARC-Segment-Number
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcSegmentNumber">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/WarcSegmentNumber</a>
<b>Type</b>	Integer
<b>Description</b>	Sequence number of the current segment in the logical whole record
<b>Property</b>	isNonCompliant
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/isNonCompliant">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRecordBaseProperty/isNonCompliant</a>
<b>Type</b>	Boolean
<b>Description</b>	WARC record non-valid member status
<b>Property</b>	Computed-Block-Digest
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedBlockDigest">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedBlockDigest</a>
<b>Type</b>	String
<b>Description</b>	Computed block checksum
<b>Property</b>	Computed-Digest-Algorithm
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedBlockDigestAlgorithm">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedBlockDigestAlgorithm</a>
<b>Type</b>	String
<b>Description</b>	Digest algorithm for the block
<b>Property</b>	Computed-Digest-Encoding
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedBlockDigestEncoding">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedBlockDigestEncoding</a>
<b>Type</b>	String
<b>Description</b>	Digest encoding for the block (for example base 32, base 64...)
<b>Property</b>	Computed-Payload-Digest
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedPayloadDigest">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedPayloadDigest</a>
<b>Type</b>	String
<b>Description</b>	Computed payload checksum (if any)
<b>Property</b>	Computed-Digest-Algorithm
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedPayloadDigestAlgorithm">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedPayloadDigestAlgorithm</a>
<b>Type</b>	String
<b>Description</b>	Digest algorithm for the payload

# JHOVE2

<b>Property</b>	Computed-Digest-Encoding
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedPayloadDigestEncoding">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedPayloadDigestEncoding</a>
<b>Type</b>	String
<b>Description</b>	Digest encoding for the payload (for example base 32, base 64...)

## WarcWarcinfoProperties

<b>Property</b>	WARC-Filename
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcWarcinfoProperty/WarcFileName">http://jhove2.org/terms/property/org/jhove2/core/source/WarcWarcinfoProperty/WarcFileName</a>
<b>Type</b>	String
<b>Description</b>	Name of the original WARC file

## WarcResponseProperties

<b>Property</b>	WARC-Target-URI
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/WarcTargetUri">http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/WarcTargetUri</a>
<b>Type</b>	String
<b>Description</b>	Original URI whose capture gave rise to the information content in the record
<b>Property</b>	WARC-Concurrent-To
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/WarcConcurrentTo">http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/WarcConcurrentTo</a>
<b>Type</b>	String
<b>Description</b>	Expresses a relationship between different records Note that as an exception to the general rule, it is possible to have several WARC-Concurrent-To fields
<b>Property</b>	WARC-IP-Address
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/WarcIPAddress">http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/WarcIPAddress</a>
<b>Type</b>	String
<b>Description</b>	Numeric Internet address contacted to retrieve any included content
<b>Property</b>	IP-Address-Version
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/IPAddressVersion">http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/IPAddressVersion</a>
<b>Type</b>	Integer
<b>Description</b>	Version of the IP Address (4 or 6)
<b>Property</b>	WARC-Warcinfo-ID
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/WarcWarcinfoID">http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/WarcWarcinfoID</a>
<b>Type</b>	String
<b>Description</b>	Record ID of the associated Warcinfo
<b>Property</b>	ProtocolResultCode
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/ProtocolResultCode">http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/ProtocolResultCode</a>
<b>Type</b>	Integer
<b>Description</b>	Protocol response result code (ex: 200) [RFC1945] and [RFC2616]
<b>Property</b>	ProtocolVersion
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/ProtocolVersion">http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/ProtocolVersion</a>
<b>Type</b>	String
<b>Description</b>	Protocol version (example: HTTP/1.0) [RFC1945] and [RFC2616]
<b>Property</b>	ProtocolContentType
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/ProtocolContentType">http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/ProtocolContentType</a>
<b>Type</b>	String
<b>Description</b>	Protocol content type (= payload content type) [RFC1945] and [RFC2616]



<b>Property</b>	ServerName
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/WarcResponseProperty/ServerName
<b>Type</b>	Integer
<b>Description</b>	Name of the server/software that delivered the answer [RFC1945] and [RFC2616]

#### WarcResourceProperties

<b>Property</b>	WARC-Target-URI
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/WarcResourceProperty/WarcTargetUri
<b>Type</b>	String
<b>Description</b>	Original URI whose capture gave rise to the information content in the record
<b>Property</b>	WARC-Concurrent-To
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/WarcResourceProperty/WarcConcurrentTo
<b>Type</b>	String
<b>Description</b>	Expresses a relationship between different records Note that as an exception to the general rule, it is possible to have several WARC-Concurrent-To fields
<b>Property</b>	WARC-IP-Address
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/WarcResourceProperty/WarcIPAddress
<b>Type</b>	String
<b>Description</b>	Numeric Internet address contacted to retrieve any included content
<b>Property</b>	IP-Address-Version
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/WarcResourceProperty/IPAddressVersion
<b>Type</b>	Integer
<b>Description</b>	Version of the IP Address (4 or 6)
<b>Property</b>	WARC-Warcinfo-ID
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/WarcResourceProperty/WarcWarcinfoID
<b>Type</b>	String
<b>Description</b>	Record ID of the associated Warcinfo

#### WarcRequestProperties

<b>Property</b>	WARC-Target-URI
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/WarcRequestProperty/WarcTargetUri
<b>Type</b>	String
<b>Description</b>	Original URI whose capture gave rise to the information content in the record
<b>Property</b>	WARC-Concurrent-To
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/WarcRequestProperty/WarcConcurrentTo
<b>Type</b>	String
<b>Description</b>	Expresses a relationship between different records Note that as an exception to the general rule, it is possible to have several WARC-Concurrent-To fields
<b>Property</b>	WARC-IP-Address
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/WarcRequestProperty/WarcIPAddress
<b>Type</b>	String
<b>Description</b>	Numeric Internet address contacted to retrieve any included content
<b>Property</b>	IP-Address-Version
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/WarcRequestProperty/IPAddressVersion
<b>Type</b>	Integer
<b>Description</b>	Version of the IP Address (4 or 6)

## JHOVE2

<b>Property</b>	WARC-Warcinfo-ID
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRequestProperty/WarcWarcinfoID">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRequestProperty/WarcWarcinfoID</a>
<b>Type</b>	String
<b>Description</b>	Record ID of the associated Warcinfo
<b>Property</b>	ProtocolVersion
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRequestProperty/ProtocolVersion">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRequestProperty/ProtocolVersion</a>
<b>Type</b>	String
<b>Description</b>	Protocol version (example: HTTP/1.0) [RFC1945] and [RFC2616]
<b>Property</b>	ProtocolUserAgent
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRequestProperty/ProtocolUserAgent">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRequestProperty/ProtocolUserAgent</a>
<b>Type</b>	String
<b>Description</b>	User Agent expressed in the HTTP request (if any) [RFC1945] and [RFC2616]

### WarcMetadataProperties

<b>Property</b>	WARC-Target-URI
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcMetadataProperty/WarcTargetUri">http://jhove2.org/terms/property/org/jhove2/core/source/WarcMetadataProperty/WarcTargetUri</a>
<b>Type</b>	String
<b>Description</b>	Original URI whose capture gave rise to the information content in the record
<b>Property</b>	WARC-Concurrent-To
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcMetadataProperty/WarcConcurrentTo">http://jhove2.org/terms/property/org/jhove2/core/source/WarcMetadataProperty/WarcConcurrentTo</a>
<b>Type</b>	String
<b>Description</b>	Expresses a relationship between different records Note that as an exception to the general rule, it is possible to have several WARC-Concurrent-To fields
<b>Property</b>	WARC-Refers-To
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcMetadataProperty/WarcRefersTo">http://jhove2.org/terms/property/org/jhove2/core/source/WarcMetadataProperty/WarcRefersTo</a>
<b>Type</b>	String
<b>Description</b>	Expresses a relationship between different records
<b>Property</b>	WARC-IP-Address
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcMetadataProperty/WarcIPAddress">http://jhove2.org/terms/property/org/jhove2/core/source/WarcMetadataProperty/WarcIPAddress</a>
<b>Type</b>	String
<b>Description</b>	Numeric Internet address contacted to retrieve any included content
<b>Property</b>	IP-Address-Version
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcMetadataProperty/IPAddressVersion">http://jhove2.org/terms/property/org/jhove2/core/source/WarcMetadataProperty/IPAddressVersion</a>
<b>Type</b>	Integer
<b>Description</b>	Version of the IP Address (4 or 6)
<b>Property</b>	WARC-Warcinfo-ID
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcMetadataProperty/WarcWarcinfoID">http://jhove2.org/terms/property/org/jhove2/core/source/WarcMetadataProperty/WarcWarcinfoID</a>
<b>Type</b>	String
<b>Description</b>	Record ID of the associated Warcinfo

### WarcRevisitProperties

<b>Property</b>	WARC-Target-URI
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRevisitProperty/WarcTargetUri">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRevisitProperty/WarcTargetUri</a>
<b>Type</b>	String
<b>Description</b>	Original URI whose capture gave rise to the information content in the record
<b>Property</b>	WARC-Profile
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRevisitProperty/WarcProfile">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRevisitProperty/WarcProfile</a>
<b>Type</b>	String
<b>Description</b>	URI signifying the kind of analysis and handling applied in a revisit record





<b>Property</b>	WARC-Refers-To
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRevisitProperty/WarcRefersTo">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRevisitProperty/WarcRefersTo</a>
<b>Type</b>	String
<b>Description</b>	Expresses a relationship between different records
<b>Property</b>	WARC-IP-Address
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRevisitProperty/WarcIPAddress">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRevisitProperty/WarcIPAddress</a>
<b>Type</b>	String
<b>Description</b>	Numeric Internet address contacted to retrieve any included content
<b>Property</b>	IP-Address-Version
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRevisitProperty/IPAddressVersion">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRevisitProperty/IPAddressVersion</a>
<b>Type</b>	Integer
<b>Description</b>	Version of the IP Address (4 or 6)
<b>Property</b>	WARC-Warcinfo-ID
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcRevisitProperty/WarcWarcinfoID">http://jhove2.org/terms/property/org/jhove2/core/source/WarcRevisitProperty/WarcWarcinfoID</a>
<b>Type</b>	String
<b>Description</b>	Record ID of the associated Warcinfo

#### WarcConversionProperties

<b>Property</b>	WARC-Target-URI
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcConversionProperty/WarcTargetUri">http://jhove2.org/terms/property/org/jhove2/core/source/WarcConversionProperty/WarcTargetUri</a>
<b>Type</b>	String
<b>Description</b>	Original URI whose capture gave rise to the information content in the record
<b>Property</b>	WARC-Refers-To
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcConversionProperty/WarcRefersTo">http://jhove2.org/terms/property/org/jhove2/core/source/WarcConversionProperty/WarcRefersTo</a>
<b>Type</b>	String
<b>Description</b>	Expresses a relationship between different records
<b>Property</b>	WARC-Warcinfo-ID
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcConversionProperty/WarcWarcinfoID">http://jhove2.org/terms/property/org/jhove2/core/source/WarcConversionProperty/WarcWarcinfoID</a>
<b>Type</b>	String
<b>Description</b>	Record ID of the associated Warcinfo

#### WarcContinuationProperties

<b>Property</b>	WARC-Target-URI
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcContinuationProperty/WarcTargetUri">http://jhove2.org/terms/property/org/jhove2/core/source/WarcContinuationProperty/WarcTargetUri</a>
<b>Type</b>	String
<b>Description</b>	Original URI whose capture gave rise to the information content in the record
<b>Property</b>	WARC-Segment-Origin-ID
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcContinuationProperty/WarcSegmentOriginID">http://jhove2.org/terms/property/org/jhove2/core/source/WarcContinuationProperty/WarcSegmentOriginID</a>
<b>Type</b>	String
<b>Description</b>	Identifies the starting record in a series of segmented records whose content blocks are reassembled to obtain a logically complete content block
<b>Property</b>	WARC-Segment-Total-Length
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcContinuationProperty/WarcSegmentTotalLength">http://jhove2.org/terms/property/org/jhove2/core/source/WarcContinuationProperty/WarcSegmentTotalLength</a>
<b>Type</b>	String
<b>Description</b>	In the final record of a segmented series, reports the total length of all segment content blocks when concatenated together



<b>Property</b>	WARC-Warcinfo-ID
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/WarcContinuationProperty/WarcWarcinfoID">http://jhove2.org/terms/property/org/jhove2/core/source/WarcContinuationProperty/WarcWarcinfoID</a>
<b>Type</b>	String
<b>Description</b>	Record ID of the associated Warcinfo

## 6 Configuration

As stated in the JHOVE2 *Architectural Overview*:

JHOVE2 uses the Spring Java enterprise platform ([www.springsource.org](http://www.springsource.org)) to manage module instantiation. JHOVE2 is implemented in terms of the Spring philosophy of *dependency injection*: all relationships between JHOVE2 components are defined in external configuration files, not the Java source code, so they can be modified easily by a JHOVE2 user at the time of installation or invocation.

### Spring XML configuration files

The Spring configuration file for the WarcModule and the WARC Format is:  
`config/spring/module/format/warc/jhove2-warc-config.xml`

The Spring bean with id="WarcModule" is used to create instances of the module and, among other things, sets the values for following properties:

The property "format" is set to the WARC Format bean.

The boolean property "recurse" specifies whether or not the WARC module should recursively characterize the payload of a WARC record.

### Parallelization

The WARC module does not employ parallelization. Since all records in a WARC file are processed sequentially, it makes little sense to process them in parallel. At most a slight increase in performance could be accomplished at the expense of increased I/O and the potential for out of order reporting of records.

### Java properties files

The Java property file for the WarcModule is `config/message/warc_messages.properties`. It defines the module format validation error message templates.

Each line in a Java properties files defines a name/value pair and has the general form: `name = value`. Colons (":"), meaningful spaces (" "), and non-printing characters in property names must be escaped in order for the file to be parsed correctly, e.g. "\:", "\ ", "\000" (the NUL character, 0x00).

### Temporary files

Since the JHove2 characterization process must be able to work recursively with potentially large files, which in turn can be container formats, JHove2 employs a strategy of automatically creating temporary files when necessary. Temporary files can to some extent be avoided by using a large enough BufferSize, but this unfortunately more often than not results in OutOfMemory errors.

## JHOVE2

Working on data files only creates temporary files in cases where any contained data requires a transformation in the unwrapping. The most common transformation being data decompression. As an example working on a WARC file will not create any temporary files for the records since the data is available directly in the WARC file. On the other hand working on GZip compressed WARC records will create temporary files for the GZip entries but not for the WARC records since they are available directly in the temporary GZip entry files. Zip and tar.gz files would also create temporary files, where tar, html, gif, jpeg and pdf files would not.

The maximum size of the data chunk held in memory is defined by the property `BufferSize` of the JHOVE2 Invocation object. If no such object is defined in the JHOVE2 configuration, the default value applies: 131072 bytes. To adjust the value, one needs to modify the main JHOVE2 configuration file (`config/spring/jhove2-config.xml`) to define an Invocation object and set the property value. Alas, the JHOVE2 command line tool (`jhove2.sh/bat/cmd`) ignores the Invocation objects defined in the configuration and creates its own, populated from the command line arguments. The “-b” option corresponds to the `BufferSize` property.

As temporary file creation, writing and reading rely on slow disk I/Os, it is recommended, if characterization performance (in CPU time) is important, to have the temporary file directory pointing to a RAM disk. See your operating system documentation for the procedure to create a RAM disk.

## 7 Implementation Notes

The compressed WARC format (`warc.gz`) is an interweaved file format: while a GZIP-compressed TAR file is made of a single GZIP member, the tarball itself, a compressed WARC file is made of a series of individually compressed WARC records.

In order to support the characterization of GZIP compressed WARC files there is a co-dependence between the GZIP and WARC modules. Since it is a requirement that one can validate a WARC file as a single entity, a single `reader` must be used to read the individually compressed records. In order to circumvent certain restrictions imposed by the current persistence model, a solution was found by manipulating the `Source` hierarchy and persisting the reader in a static field inside the GZIP module.

The JHOVE2 WARC module is an original implementation of the WARC specification ([WARC]). It does not reuse code from any other existing implementation (namely [Heritrix]).

## **M JHOVE2 ARC module specifications**



## JHOVE2: Next-Generation Architecture for Format-Aware Characterization ARC Module

Version 2.1.0

Issued 2012-12-04

Status Draft

### 1 Introduction

JHOVE2 is a framework and application for next-generation format-aware characterization of digital objects. The function of JHOVE2 is encapsulated in a series of modules that can be configured for use within the framework's plug-in architecture. The ARC module provides characterization services for the ARC format.

#### **Important information for users of the JHOVE2 ARC module**

The authoritative specification for ARC [ARC] is *ambiguous*.

Validation of ARC instances by this module is *selective*.

**NOTE** A format specification is considered *unambiguous* if there is broad community consensus regarding the intention of *all* normative requirements of the format's authoritative specification; otherwise it is considered *ambiguous*, and areas of potential ambiguity will be documented below.

Module validation is considered *comprehensive* if *all* normative requirements defined by that specification are validated by the module; otherwise it is considered *selective*, and non-validated features will be documented below.

### 2 Identification

<b>Primary format or format family</b>	
<b>Canonical format name:</b>	arc
<b>Alias format name(s):</b>	arc
<b>Canonical format identifier:</b>	JHOVE2 <a href="http://jhove2.org/terms/format/arc">http://jhove2.org/terms/format/arc</a>
<b>Alias format identifier(s):</b>	PRONOM PUID: x-fmt/219
	MIME application/x-ia-arc

<b>JHOVE2 ARC module</b>	
<b>JHOVE2 module name:</b>	ArcModule
<b>JHOVE2 module identifier:</b>	JHOVE2 <a href="http://jhove2.org/terms/reportable/org/jhove2/module/format/arc/ArcModule">http://jhove2.org/terms/reportable/org/jhove2/module/format/arc/ArcModule</a>
<b>JHOVE2 module class</b>	org.jhove2.module.format.arc.ArcModule.java org.jhove2.module.format.arc.ArcModule.class
<b>JHOVE2 module jar</b>	



<i>ARC File or stream Signature</i>		
File format	Jhove2 Profile	File Header(s) Signature(s)
<b>arc</b>	arc	filedesc://

### 3 References

For the purposes of the JHOVE2 ARC module the authoritative format specifications are:

- [ARC] ARC File Format Reference, Mike Burner & Brewster Kahle (Internet Archive), 15 september 1996, version 1.0.  
<http://www.archive.org/web/researcher/ArcFileFormat.php>
- [ARC\_Heritrix] It was developed as an extension to ARC File Format version 1.0, to allow writing of extra metadata into first record of an ARC file  
<http://www.archive.org/arc/1.0/arc.html>

#### Other Useful References:

- [Heritrix] Internet Archive's web crawler  
<http://crawler.archive.org/>
- [ARCWriter] Java module that writes ARC files  
<http://crawler.archive.org/apidocs/org/archive/io/arc/ARCWriter.html>
- [ARC\_IA] Internet Archive ARC file format  
<http://www.digitalpreservation.gov/formats/fdd/fdd000235.shtml>
- [dk.netarkivet.ArcUtils] Java ARC utilities  
<http://netarchive.dk/kildetekster/index-en.php>
- [RFC2616] Hypertext Transfer Protocol -- HTTP/1.1  
<http://tools.ietf.org/id/draft-ietf-http-v11-spec-rev-06.txt>
- [RFC1945] Hypertext Transfer Protocol -- HTTP/1.0  
[http://datatracker.ietf.org/doc/rfc1945/?include\\_text=1](http://datatracker.ietf.org/doc/rfc1945/?include_text=1)

### 4 Validity

#### 4.1 General

ARC files contain one version block and possibly several trailing ARC records. To be considered a valid ARC file the version block and every ARC record in the file must be valid. An ARC record is valid if its URL record is compliant with the version block URL record definition, and composed of valid fields (all mandatory fields are defined and their format is valid).

Note that the ARC record is valid even if the characterization of its payload fails.

# JHOVE2

Please refer to the following reference for a definition:

<http://www.archive.org/web/researcher/ArcFileFormat.php>

## 4.2 Format versions

JHOVE2 treats the ARC format as a family having several versions.

The ARC module checks that:

- The ARC file is either compliant with version 1 block or version 2 block definition.
- ARC file version format is either '1.0' ('<reserved>' field of the version block is '0'), '1.1' ('<reserved>' field is '1') or '2.0' ('<reserved>' field of the version block is '0').
- Each ARC Record contains a protocol response and an object.
- The version block contains a payload, if the ARC file is version '1.1'.

The version block definition must be the same for the version block and all records present in the ARC file.

## 4.3 Version block

The version block identifies the original filename, file version, and URL record fields of the archive file.

The ARC file format supports version 1 block and version 2 block definitions:

### Version 1 block definition

```
filedesc://<path><sp><ip-address><sp><date><sp>text/plain<sp><length><nl>
1<sp><reserved><sp><origin-code><nl>
URL<sp>IP-address<sp>Archive-date<sp>Content-type<sp>Archive-length<nl>
<network_doc><nl>
```

### Version 2 block definition

```
filedesc://<path><sp><ip-address><sp><date><sp>text/plain<sp>200<sp>-<sp>-
<sp>0<sp><filename><sp><length><nl>
2<sp><reserved><sp><origin-code><nl>
URL<sp>IP-address<sp>Archive-date<sp>Content-type<sp>Result-
code<sp>Checksum<sp>Location<sp>Offset<sp>Filename<sp>Archive-length<nl>
<network_doc><nl>
```

A version block is valid, if it is either compliant with version 1 block or version 2 block definition, and contains valid fields (all mandatory fields are defined and their formats are valid).

## 4.4 Validation implemented

In order to ensure the validity of an ARC file the module reads the whole file sequentially from beginning to end looking for records to validate. The module will only report a valid ARC file if this process does not encounter any problems warranting errors or warnings.

Should the module be unable to read the entire file because of a problem (runtime exception), the validity of the ARC file is undetermined until the module is corrected or the ARC file validated by other means. Problems with the underlying file system can result in the reader not being able to validate the whole file.

Errors/warnings are reported on a file or record level. Normally errors/warnings are reported in the offending record. In case there is no current record to attach errors/warnings to, they are reported in the reader.



So if the module is reading a non ARC file it will most likely not report any records, instead errors/warnings will be reported in the reader and the file will be considered invalid. Similarly any garbage after ARC records will not return a record but will still report errors/warnings in the reader and the ARC file will be considered invalid. Any garbage in front of a record will not be reported in the reader but will trigger errors/warnings on the record level since it is in the beginning of the record.

Records with incorrect content-length values will be handled according to the situation. A content-length which is different from the actual payload will result in garbage errors/warnings in the preceding record or the reader. If the length is lesser the next record will be read and returned. On the other hand, if the length is greater, the next record header will not be read and instead result in errors/warnings being reported in the next record found or the reader if no more records are found.

In cases where there is a known payload, like an http response/request, errors/warnings are reported if the expected length of the payload is different from the one stated in the ARC header.

#### 4.4.1 Reported version errors/warnings

The following errors can be reported when trying to parse the version/record.

- Error "**Invalid data before WARC version**": if there is data in front of the version line.
- Error "**Invalid empty lines before WARC version**": if there are empty lines in front of the version block or record.
- Error "**Invalid result-code**": if the result code is not a number between 100 and 999.
- Error "**Invalid offset**": if the offset is negative.
- Error "**Invalid archive-length**": if the archive-length is negative.

#### 4.4.2 Reported ARC Record base errors

The following errors can be reported by the common ARC Record Base.

- Error "**Invalid offset value**": if the offset in the record does not match the actually offset in the input stream.
- Error "**Expected a version block as the first record**": if the first record is not a version block.
- Error "**Expected an arc record not a version block**": if a version block appears but not as the first record in the ARC file.
- Error "**ARC record does not match the version block definition**": if a version block or ARC record header line does not match the version declared in the version block payload.(\*)
- Error "**Payload length mismatch**": in case the payload was truncated.
- Error "**Invalid expected 1 trailing newline**": if there are more or less than 1 newline.

(\*) technically a version block is a normal ARC record with the only difference that it's payload is the version description.

#### 4.4.3 Reported Version Block errors/warnings

The following errors/warnings can be reported while parsing a version block.

- Error "**Expected a content-type**": in no content-type was found.
- Warning: "**Expected text/plain content-type**": if the content-type deviated from text/plain.
- Error "**Version block is not valid**": if a version header could not be parsed in the payload.
- Error "**VersionBlock length missing**": if there is only a filedesc record header but no payload.
- Error "**Expected metadata payload not found in the version block**": the ARC v1.1 version block should have trailing metadata after the version header in the payload.





- Error **“Metadata payload must not be present in this version”**: if metadata is present beyond the version header and ARC version is not v1.1.

#### 4.4.4 Reported Version Header errors/warnings

The following errors can be reported while parsing a version header.

- Error **“Invalid version description”**: if the version line is not a triple.
- Error **“Invalid version”**: if the version number is not a valid version <x.x>.
- Error **“Version line empty”**: if the version line is empty.
- Error **“Unsupported version block definition”**: if the version block definition is neither version 1 or 2.
- Error **“Block definition empty”**: if the version block definition is empty.
- Error **“Version number does not match the block definition”**: if the version number and the version block definition are not compatible.

#### 4.4.5 Reported ARC record errors

The following errors can be reported for an Arc Record.

- Error **“Unable to parse HTTPHeader”**: if for some reason the HTTP Header is malformed.
- Error **“Expected payload not found in the record block”**: if the archive-length is zero but the URL scheme and content-type indicate a payload should be present.

#### 4.4.6 HeaderLine reader errors

The (header)line reader used to read ARC and HTTP headers reads and validates lines according to the specifications. It will validate “quoted strings” and “encoded words”.

The following errors can be returned from the (header)line reader.

- Error **“Unexpected EOF”**: if a (header)line is not completed by a LF but an EOF.
- Error **“Misplaced CR”**: if a CR is not preceded by a LF.
- Error **“Missing CR”**: if the reader is expecting CRLFs and only encounters a LF.
- Error **“Excessive CR”**: if the reader is expecting LFs and encounters a CRLF.
- Error **“Invalid US-ASCII character”**: basically if the character is not 7-bit.
- Error **“Invalid control character”**: if the character is between 0-31, with some exceptions.
- Error **“Invalid separator character”**: if a separator character appears where it is not supposed to be, like in the header name.
- Error **“Missing quote”**: if a quoted string ends without a quotation mark.
- Error **“Missing quoted pair character”**: if a quoted string ends with a backslash but with no quoted pair character.

#### 4.4.7 Reported field format errors/warnings

The following header value formats are validated.

- String
- Integer
- Long
- ARC date
- IP address
- URI
- Content-Type

The following errors/warnings can be reported.

- Error **“Required/missing field”**: if a value is missing.

## JHOVE2

- Error “**Invalid numeric format**”: if the value is not a valid integer or long.
- Error “**Invalid ARC date format**”: if the date format does not conform to the standard.
- Error “**Invalid IPv4 or IPv6 format**”: if the value is not a valid IP.
- Error “**Invalid URI format**”: if the value is not a valid URI according to the URI profile used to validate.
- Error “**Invalid relative URI**”: if the URI is not absolute.
- Error “**Invalid Content-Type format**”: if the value is valid according to rfc2616.

### 4.4.8 Reported digest errors

- Error “**Invalid block/payload digest encoding scheme**”: in case an unknown encoding scheme was requested.

## 5 Reportable properties

The list below does not exhaustively traverse the hierarchy of all possible properties which may be reported by the ARC Module.

### 5.1 ArcModule properties

ArcModule	
<b>Property</b>	ArcFileName
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/arc/ArcModule/ArcFileName
<b>Type</b>	String
<b>Description</b>	ARC file name
<b>Property</b>	ArcFileSize
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/arc/ArcModule/ArcFileSize
<b>Type</b>	Long
<b>Description</b>	ARC file size
<b>Property</b>	LastModified
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/arc/ArcModule/LastModified
<b>Type</b>	Date
<b>Description</b>	Last modified date of the ARC file
<b>Property</b>	ArcRecordNumber
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/arc/ArcModule/ArcRecordNumber
<b>Type</b>	Int
<b>Description</b>	Number of ARC records
<b>Property</b>	ReaderConsumedBytes
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/arc/ArcModule/ReaderConsumedBytes
<b>Type</b>	Long
<b>Description</b>	ARC reader consumed bytes
<b>Property</b>	FileVersion
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/arc/ArcModule/FileVersion
<b>Type</b>	String
<b>Description</b>	File version, only present if it is the same for all records
<b>Property</b>	ArcBlockDescVersion
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/arc/ArcModule/ArcBlockDescVersion
<b>Type</b>	String
<b>Description</b>	Block description version, only present if it is the same for all records



<b>Property</b>	Protocols
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/arc/ArcModule/Protocols
<b>Type</b>	Map<String,Integer>
<b>Description</b>	URL record protocols
<b>Property</b>	Errors
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/arc/ArcModule/Errors
<b>Type</b>	Map<String,Integer>
<b>Description</b>	The number of errors by error type

Validator (ArcModule)	
<b>Property</b>	isValid
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/Validator/isValid
<b>Type</b>	Validator\$Validity
<b>Description</b>	Validation status.
<b>Property</b>	Coverage
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/Validator/Coverage
<b>Type</b>	Validator\$Coverage
<b>Description</b>	Format module validation coverage.

## 5.2 ArcRecordSource properties

In JHove2 terms each record found inside an ARC file is represented as an *ArcRecordSource*. An *ArcRecordSource* always contains a set of *ArcRecordBaseProperties* which represents all the common properties irrespective of the record type. In addition to the base properties each *WarcRecordSource* also include properties which are relative to the record type. These are defined here as *Warc<WarcType>Properties*.

ArcVersionBlockProperties

ArcRecordProperties

ArcRecordBaseProperties	
<b>Property</b>	StartOffset
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/StartOffset
<b>Type</b>	Long
<b>Description</b>	ARC record start offset, in bytes
<b>Property</b>	ArcBlockDescVersion
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/module/format/arc/ArcModule/ArcBlockDescVersion
<b>Type</b>	String
<b>Description</b>	ARC Block description version header value
<b>Property</b>	IpAddress
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/IpAddress
<b>Type</b>	String
<b>Description</b>	ARC record IP address
<b>Property</b>	IpAddressVersion
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/IpAddressVersion
<b>Type</b>	String
<b>Description</b>	IP address version, 4 or 6 if valid IP address



<b>Property</b>	ArchiveDate
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/ArchiveDate">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/ArchiveDate</a>
<b>Type</b>	String
<b>Description</b>	ARC record archive date
<b>Property</b>	RawArchiveDate
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/RawArchiveDate">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/RawArchiveDate</a>
<b>Type</b>	String
<b>Description</b>	ARC record raw archive date
<b>Property</b>	ContentType
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/ContentType">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/ContentType</a>
<b>Type</b>	String
<b>Description</b>	ARC record content type
<b>Property</b>	Length
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/Length">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/Length</a>
<b>Type</b>	Long
<b>Description</b>	Network doc length, in bytes
<b>Property</b>	ResultCode
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/ResultCode">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/ResultCode</a>
<b>Type</b>	Integer
<b>Description</b>	ARC record result code
<b>Property</b>	Checksum
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/Checksum">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/Checksum</a>
<b>Type</b>	String
<b>Description</b>	ARC record object checksum
<b>Property</b>	Location
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/Location">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/Location</a>
<b>Type</b>	String
<b>Description</b>	ARC record location
<b>Property</b>	Offset
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/Offset">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/Offset</a>
<b>Type</b>	Long
<b>Description</b>	ARC record offset, in bytes
<b>Property</b>	FileName
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/FileName">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/FileName</a>
<b>Type</b>	String
<b>Description</b>	ARC record file name
<b>Property</b>	hasPayload
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/hasPayload">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/hasPayload</a>
<b>Type</b>	Boolean
<b>Description</b>	Specifies whether the ARC record contains a payload
<b>Property</b>	ObjectSize
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/ObjectSize">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/ObjectSize</a>
<b>Type</b>	Long
<b>Description</b>	ARC record object size, in bytes
<b>Property</b>	isNonCompliant
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/isNonCompliant">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseSource/isNonCompliant</a>
<b>Type</b>	Boolean
<b>Description</b>	ARC record non-valid member status



<b>Property</b>	Computed-Block-Digest
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedBlockDigest">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedBlockDigest</a>
<b>Type</b>	String
<b>Description</b>	Computed block checksum
<b>Property</b>	Computed-Digest-Algorithm
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedBlockDigestAlgorithm">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedBlockDigestAlgorithm</a>
<b>Type</b>	String
<b>Description</b>	Digest algorithm for the block
<b>Property</b>	Computed-Digest-Encoding
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedBlockDigestEncoding">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedBlockDigestEncoding</a>
<b>Type</b>	String
<b>Description</b>	Digest encoding for the block (for example base 32, base 64...)
<b>Property</b>	Computed-Payload-Digest
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedPayloadDigest">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedPayloadDigest</a>
<b>Type</b>	String
<b>Description</b>	Computed payload checksum (if any)
<b>Property</b>	Computed-Digest-Algorithm
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedPayloadDigestAlgorithm">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedPayloadDigestAlgorithm</a>
<b>Type</b>	String
<b>Description</b>	Digest algorithm for the payload
<b>Property</b>	Computed-Digest-Encoding
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedPayloadDigestEncoding">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordBaseProperty/ComputedPayloadDigestEncoding</a>
<b>Type</b>	String
<b>Description</b>	Digest encoding for the payload (for example base 32, base 64...)

#### ArcVersionBlockProperties

<b>Message:</b>	VersionNumber
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/VersionBlockSource/VersionNumber">http://jhove2.org/terms/property/org/jhove2/core/source/VersionBlockSource/VersionNumber</a>
<b>Type</b>	Integer
<b>Description</b>	Version number
<b>Message:</b>	Reserved
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/VersionBlockSource/Reserved">http://jhove2.org/terms/property/org/jhove2/core/source/VersionBlockSource/Reserved</a>
<b>Type</b>	String
<b>Description</b>	ARC file version format
<b>Message:</b>	OriginCode
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/VersionBlockSource/OriginCode">http://jhove2.org/terms/property/org/jhove2/core/source/VersionBlockSource/OriginCode</a>
<b>Type</b>	String
<b>Description</b>	Version block origin code

#### ArcRecordProperties

<b>Property</b>	ProtocolResultCode
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordSource/ProtocolResultCode">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordSource/ProtocolResultCode</a>
<b>Type</b>	Integer
<b>Description</b>	Protocol response result code

## JHOVE2

<b>Property</b>	ProtocolVersion
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordSource/ProtocolVersion">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordSource/ProtocolVersion</a>
<b>Type</b>	String
<b>Description</b>	Protocol version
<b>Property</b>	ProtocolContentType
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordSource/ProtocolContentType">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordSource/ProtocolContentType</a>
<b>Type</b>	String
<b>Description</b>	Protocol content type
<b>Property</b>	ServerName
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordSource/ServerName">http://jhove2.org/terms/property/org/jhove2/core/source/ArcRecordSource/ServerName</a>
<b>Type</b>	String
<b>Description</b>	Server name header value

## 6 Configuration

As stated in the JHOVE2 *Architectural Overview*:

JHOVE2 uses the Spring Java enterprise platform ([www.springsource.org](http://www.springsource.org)) to manage module instantiation. JHOVE2 is implemented in terms of the Spring philosophy of *dependency injection*: all relationships between JHOVE2 components are defined in external configuration files, not the Java source code, so they can be modified easily by a JHOVE2 user at the time of installation or invocation.

### Spring XML configuration files

The Spring configuration file for the ArcModule and the ARC Format is:

`config/spring/module/format/arc/jhove2-arc-config.xml`

The Spring bean with `id="ArcModule"` is used to create instances of the module and, among other things, sets the values for following properties:

The property `"format"` is set to the ARC Format bean.

The Boolean property `"recurse"` defines whether the ARC module should recursively characterize the content of the network documents found in the ARC records.

### Parallelization

The ARC module does not employ parallelization. Since all records in a ARC file are processed sequentially, it makes little sense to process them in parallel. At most a slight increase in performance could be accomplished at the expense of increased I/O and the potential for out of order reporting of records.

### Java properties files

The Java property file for the ArcModule is `config/message/arc_messages.properties`. It defines the module format validation error message templates.

Each line in a Java properties files defines a name/value pair and has the general form: `name = value`. Colons (`:`), meaningful spaces (`" "`), and non-printing characters in property names must be escaped in order for the file to be parsed correctly, e.g. `"\"`, `"\"`, `"\000"` (the NUL character, 0x00).

## JHOVE2

### Temporary files

Since the JHove2 characterization process must be able to work recursively with potentially large files, which in turn can be container formats, JHove2 employs a strategy of automatically creating temporary files when necessary. Temporary files can to some extent be avoided by using a large enough `BufferSize`, but this unfortunately more often than not results in `OutOfMemory` errors.

Working on data files only creates temporary files in cases where any contained data requires a transformation in the unwrapping. The most common transformation being data decompression. As an example working on a ARC file will not create any temporary files for the records since the data is available directly in the ARC file. On the other hand working on GZip compressed ARC records will create temporary files for the GZip entries but not for the ARC records since they are available directly in the temporary GZip entry files. Zip and tar.gz files would also create temporary files, where tar, html, gif, jpeg and pdf files would not.

The maximum size of the data chunk held in memory is defined by the property `BufferSize` of the JHOVE2 Invocation object. If no such object is defined in the JHOVE2 configuration, the default value applies: 131072 bytes. To adjust the value, one needs to modify the main JHOVE2 configuration file (`config/spring/jhove2-config.xml`) to define an Invocation object and set the property value. Alas, the JHOVE2 command line tool (`jhove2.sh/bat/cmd`) ignores the Invocation objects defined in the configuration and creates its own, populated from the command line arguments. The “-b” option corresponds to the `BufferSize` property.

As temporary file creation, writing and reading rely on slow disk I/Os, it is recommended, if characterization performance (in CPU time) is important, to have the temporary file directory pointing to a RAM disk. See your operating system documentation for the procedure to create a RAM disk.

## 7 Implementation Notes

The compressed ARC format (`arc.gz`) is an interweaved file format: while a GZIP-compressed TAR file is made of a single GZIP member, the tarball itself, a compressed ARC file is made of a series of individually compressed ARC records.

Only the first record (the ARC version block) bears the ARC file signature (the “`filedesc://`” URI scheme). Hence, only the first record shall be submitted for identification; subsequent records need to be directly handed over to the format module object associated to the first record source.

So in order to support the characterization of GZIP compressed ARC files there is a co-dependence between the GZIP and ARC modules. Since it is a requirement that one can validate a ARC file as a single entity, a single `reader` must be used to read the individually compressed records. In order to circumvent certain restrictions imposed by the current persistence model, a solution was found by manipulating the `Source` hierarchy and persisting the reader in a static field inside the GZIP module.

The JHOVE2 ARC module is an original implementation of the ARC specification ([ARC]). It does not reuse code from any other existing implementation (namely [Heritrix]).

## N JHOVE2 GZip module specifications





## JHOVE2: Next-Generation Architecture for Format-Aware Characterization GZIP Module

Version 2.1.0

Issued 2013-02-10

Status Draft

### 1 Introduction

JHOVE2 is a framework and application for next-generation format-aware characterization of digital objects. The function of JHOVE2 is encapsulated in a series of modules that can be configured for use within the framework's plug-in architecture. The GZIP format module provides characterization services for the GZIP format.

#### **Important information for users of the JHOVE2 GZIP module**

The authoritative specification for GZIP [RFC1952] is *unambiguous*.

Validation of GZIP files by this module is *selective*.

**NOTE** A format specification is considered *unambiguous* if there is broad community consensus regarding the intention of *all* normative requirements of the format's authoritative specification; otherwise it is considered *ambiguous*, and areas of potential ambiguity will be documented below.

Module validation is considered *comprehensive* if *all* normative requirements defined by that specification are validated by the module; otherwise it is considered *selective*, and non-validated features will be documented below.

### 2 Identification

Primary format or format family	
Canonical format name:	GZip
Alias format name(s):	GZIP
Canonical format identifier:	JHOVE2 <a href="http://jhove2.org/terms/format/gzip">http://jhove2.org/terms/format/gzip</a>
Alias format identifier(s):	PRONOM PUID: x-fmt/266 MIME application/x-gzip RFC RFC 1952 UTI org.gnu.gnu-zip-archive

JHOVE2 Gzip module	
JHOVE2 module name:	GzipModule
JHOVE2 module identifier:	JHOVE2 <a href="http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule">http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule</a>
JHOVE2 module class	org.jhove2.module.format.gzip.GzipModule.java org.jhove2.module.format.gzip.GzipModule.class
JHOVE2 module jar	



<b>GZIP File or Stream Signatures</b>		
<b>File format</b>	<b>Jhove2 Profile</b>	<b>File Header(s) Signature(s)</b>
<b>GZip</b>	-	0x8b1f

### 3 References

For the purposes of the JHOVE2 GZip module the authoritative format specifications are:

- [RFC1952] GZIP file format specification version 4.3, P. Deutsch, May 1996, RFC 1952.  
<http://www.ietf.org/rfc/rfc1952.txt>
- [RFC1951] DEFLATE Compressed Data Format Specification version 1.3, P. Deutsch, May 1996, RFC 1951  
<http://www.ietf.org/rfc/rfc1951.txt>

#### Other Useful References:

- [GZIP] *gzip* compression utility community site  
<http://www.gzip.org/>
- [Gzip Manual] *gzip* manual, edition 1.2.3, July 1993  
[http://www.math.utah.edu/docs/info/gzip\\_toc.html](http://www.math.utah.edu/docs/info/gzip_toc.html)
- [Gzip FSF] Free Software Foundation's GNU Gzip page  
<http://www.gnu.org/software/gzip/>
- [Gzip Wikipedia] Wikipedia's GNU Gzip page  
<http://en.wikipedia.org/wiki/Gzip>

### 4 Validity

#### 4.1 General

The validity of a GZIP file or member is defined in terms of compliance with section 2.3.1.2 of [RFC1952] which defines the requirements for a « compliant decompressor ». Moreover if the flag FHCRC is set, the header CRC16 value shall be equal to the computed one. Finally, the CRC32 of the compressed data and its weight (ISize) shall be verified.

A GZIP file or member is:

- Valid if the GZIP file or member is compliant with the above controls,
- Invalid (non compliant) if any error was found while performing the above controls.



Here's an excerpt of [RFC1952] describing the GZIP format:

A gzip file or stream consists of a series of "members" (compressed data sets).

A gzip file or stream follow the gzip [RFC1952] specification if:

- its members simply appear one after another in the file, with no additional information before, between, or after them
- and, the format of each member follows the following structure :

```

+-----+-----+-----+-----+-----+-----+
|ID1|ID2|CM|FLG|      MTIME      |XFL|OS| (more-->)
+-----+-----+-----+-----+-----+

(if FLG.FEXTRA set1)
+-----+-----+-----+-----+
| XLEN  |...XLEN bytes of "extra field"...| (more-->)
+-----+-----+-----+-----+

(if FLG.FNAME set2)
+-----+-----+-----+-----+
|...original file name, zero-terminated...| (more-->)
+-----+-----+-----+-----+

(if FLG.FCOMMENT set3)
+-----+-----+-----+-----+
|...file comment, zero-terminated...| (more-->)
+-----+-----+-----+-----+

(if FLG.FHCRC set4)
+-----+
| CRC16 |
+-----+
+-----+-----+-----+
|...compressed blocks...| (more-->)
+-----+-----+-----+
  0   1   2   3   4   5   6   7
+-----+-----+-----+-----+
|      CRC32      |      ISIZE      |
+-----+-----+-----+-----+

```

where the FLG flag byte is divided into individual bits as follows:

```

bit 0  FTEXT
bit 1  FHCRC
bit 2  FEXTRA
bit 3  FNAME
bit 4  FCOMMENT
bit 5  reserved
bit 6  reserved
bit 7  reserved

```

#### §2.3.1.2:

“A compliant decompressor must check ID1, ID2, and CM, and provide an error indication if any of these have incorrect values. It must examine FEXTRA/XLEN, FNAME, FCOMMENT and FHCRC at least so it can skip over the optional fields if they are present. It need not examine any other part of the header or trailer; in particular, a decompressor may ignore FTEXT and OS and always produce binary output, and still be compliant. A compliant decompressor must give an error indication if any reserved bit is non-zero, since such a bit could indicate the presence of a new field that would cause subsequent data to be interpreted incorrectly.”

<sup>1</sup> bit 2 of flag byte FLG

<sup>2</sup> bit 3 of flag byte FLG

<sup>3</sup> bit 4 of flag byte FLG

<sup>4</sup> bit 1 of flag byte FLG



## 4.2 Format versions

JHOVE2 treats the GZIP as a format as there are no GZIP declinations.

## 4.3 Validation implemented

To correctly validate a GZip file the module tries to read one GZip entry at a time until the end of the file is reached. At least one entry must be present. The GZip file is reported valid only if all entries found are valid and there are no excess bytes at the end.

Initially 10 bytes are read from the stream which is the minimum amount a header always includes. If no bytes are available and at least one entry was found the validation process stops. If on the other hand between 1 and 9 bytes could be read an error is reported on the reader.

The header is validated by verifying that all the fixed header fields have valid values. This includes checking that the two first bytes equal the defined magic number and that the fields for compression mode, flags, extra flags and operating system adhere to the specification. The mtime can be zero or a date representation.

The presence of more header fields depends on which flags and extra flags are set. If either of the FNAME and FCOMMENT flags is set they are read and the content is checked for encoding and characters used. Errors are reported on the entry.

If the FEXTRA flag is set the field is read but the sub entries are NOT validated in this version of the module. (The use of this field does not seem widespread)

If the FHCRC flag is set the CRC16 header field is compared to the one calculated on the header data. If the values are different an error is reported on the entry.

Next the entry's content is decompressed and examined further by the JHOVE2 workflow. Any decompression problems are reported on the entry.

If the entry's content was decompressed without any problems the trailing fields are read and validated. These include the decompressed size (modulo 32bit) and a CRC32 of the decompressed data. If either value does not match the computed one an error is reported on the entry.

### 4.3.1 Reported header errors

- Invalid expected magic value.
- Invalid expected compression method. (deflate only one defined)
- Reserved eXtra FLags used.
- Invalid eXtra FLags. (Slow/fast at the same time)
- Reserved FLAgS used.
- Unknown Operating System value.
- Invalid FName encoding. ISO-8859-1 expected.
- Invalid FComment encoding. ISO-8859-1 expected.
- Invalid expected CRC16 value.

## JHOVE2

- Invalid GZip file, unexpected EOF.
- Error expected one or more records.
- Invalid data, unexpected trailing data.

### 4.3.2 Reported trailing header errors

- Invalid expected CRC32 value incorrect.
- Invalid expected ISize value incorrect.
- Invalid GZip file, unexpected EOF.

### 4.3.3 Reported compression errors

- Invalid GZip file, unexpected EOF.

## 5 Reportable properties

The list below does not exhaustively traverse the hierarchy of all possible properties that may be reported by the Gzip Module.

The following properties are reported on the GZip module instance and characterize the main source object (e.g. Gzip file).

GzipModule	
<b>Property</b>	GzipFileName
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule/GzipFileName">http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule/GzipFileName</a>
<b>Type</b>	String
<b>Description</b>	GZip file name.
<b>Property</b>	GzipFileSize
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule/GzipFileSize">http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule/GzipFileSize</a>
<b>Type</b>	Long
<b>Description</b>	GZip file size.
<b>Property</b>	FileLastModified
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule/FileLastModified">http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule/FileLastModified</a>
<b>Type</b>	Date
<b>Description</b>	Last modified date of GZip file.
<b>Property</b>	DeflateMemberCount
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule/deflateMemberCount">http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule/deflateMemberCount</a>
<b>Type</b>	Long
<b>Description</b>	Number of members compressed with the deflate compression method.
<b>Property</b>	InvalidMembers
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule/InvalidMembers">http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule/InvalidMembers</a>
<b>Type</b>	Long
<b>Description</b>	Number of non-valid members.
<b>Property</b>	InvalidMembers
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule/InvalidMembers">http://jhove2.org/terms/property/org/jhove2/module/format/gzip/GzipModule/InvalidMembers</a>
<b>Type</b>	Long
<b>Description</b>	Number of non-valid members.

## JHOVE2

<b>Property</b>	ValidationMessages
<b>Identifier</b>	<a href="http://jhove2.org/terms/message/org/jhove2/module/format/gzip/GzipModule/ValidationMessages">http://jhove2.org/terms/message/org/jhove2/module/format/gzip/GzipModule/ValidationMessages</a>
<b>Type</b>	Collection<Message>
<b>Description</b>	Validation error messages.

Validator (GzipModule)	
<b>Property</b>	isValid
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/module/format/Validator/isValid">http://jhove2.org/terms/property/org/jhove2/module/format/Validator/isValid</a>
<b>Type</b>	Validator\$Validity
<b>Description</b>	Validation status.
<b>Property</b>	Coverage
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/module/format/Validator/Coverage">http://jhove2.org/terms/property/org/jhove2/module/format/Validator/Coverage</a>
<b>Type</b>	Validator\$Coverage
<b>Description</b>	Format module validation coverage.

GzipModule	
<b>Property</b>	Format
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/module/format/FormatModule/Format">http://jhove2.org/terms/property/org/jhove2/module/format/FormatModule/Format</a>
<b>Type</b>	Format
<b>Description</b>	Format module format.

The following properties are reported on the Source object corresponding to an entry (member) of the main GZIP source object (e.g. Gzip file).

GzipEntryProperties	
<b>Property</b>	isNonCompliant
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/isNonCompliant">http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/isNonCompliant</a>
<b>Type</b>	boolean
<b>Description</b>	GZip entry non-compliance status.
<b>Property</b>	Offset
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/Offset">http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/Offset</a>
<b>Type</b>	long
<b>Description</b>	GZip entry (computed) offset in source file/stream.
<b>Property</b>	Name
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/Name">http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/Name</a>
<b>Type</b>	String
<b>Description</b>	GZip entry name.
<b>Property</b>	Comment
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/Comment">http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/Comment</a>
<b>Type</b>	String
<b>Description</b>	GZip entry comment.
<b>Property</b>	LastModified
<b>Identifier</b>	<a href="http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/LastModified">http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/LastModified</a>
<b>Type</b>	Date
<b>Description</b>	GZip entry last modified date.



<b>Property</b>	CompressionMethod
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/CompressionMethod
<b>Type</b>	CompressionMethod
<b>Description</b>	GZip entry compression method.
<b>Property</b>	OperatingSystem
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/OperatingSystem
<b>Type</b>	OperatingSystem
<b>Description</b>	GZip entry operating system.
<b>Property</b>	Crc16
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/Crc16
<b>Type</b>	String
<b>Description</b>	GZip entry header CRC16 hex value.
<b>Property</b>	Crc32
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/Crc32
<b>Type</b>	String
<b>Description</b>	GZip entry CRC32 hex. value.
<b>Property</b>	ISize
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/ISize
<b>Type</b>	long
<b>Description</b>	GZip entry extracted size (ISIZE) value.
<b>Property</b>	Size
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/Size
<b>Type</b>	long
<b>Description</b>	GZip entry (computed) uncompressed size, in bytes.
<b>Property</b>	CompressedSize
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/CompressedSize
<b>Type</b>	long
<b>Description</b>	GZip entry (computed) compressed size, in bytes.
<b>Property</b>	CompressionRatio
<b>Identifier</b>	http://jhove2.org/terms/property/org/jhove2/core/source/GzipEntryProperties/CompressionRatio
<b>Type</b>	double
<b>Description</b>	GZip entry (computed) compression ratio.

## 6 Configuration

As stated in the JHOVE2 *Architectural Overview*:

JHOVE2 uses the Spring Java enterprise platform ([www.springframework.org](http://www.springframework.org)) to manage module instantiation. JHOVE2 is implemented in terms of the Spring philosophy of *dependency injection*: all relationships between JHOVE2 components are defined in external configuration files, not the Java source code, so they can be modified easily by a JHOVE2 user at the time of installation or invocation.

### Spring XML configuration files

The Spring configuration file for the GzipModule and the GZIP format is:

```
config/spring/module/format/gzip/jhove2-gzip-config.xml.
```

## JHOVE2

The Spring bean with id="GzipModule" creates instances of the module and, among other things, sets the values for following properties:

The property "format" is set to the GZIP Format bean.

The Boolean property "recurse" defines whether the GZip module should recursively characterize the content of the GZip entries.

### Parallelization

The GZip module does not employ parallelization. Since all entries in a GZip file are processed sequentially, it makes little sense to process them in parallel. At most a slight increase in performance could be accomplished at the expense of increased I/O and the potential for out of order reporting of records.

### Java properties files

The Java properties file for the GzipModule is `config/message/gzip_messages.properties`. It defines the module format validation error message templates.

Each line in a Java properties files defines a name/value pair and has the general form: `name = value`. Colons (":"), meaningful spaces (" "), and non-printing characters in property names must be escaped in order for the file to be parsed correctly, e.g. "\:", "\ ", "\000" (the NUL character, 0x00).

### Temporary files

Since the JHove2 characterization process must be able to work recursively with potentially large files, which in turn can be container formats, JHove2 employs a strategy of automatically creating temporary files when necessary. Temporary files can to some extent be avoided by using a large enough BufferSize, but this unfortunately more often than not results in OutOfMemory errors.

Working on data files only creates temporary files in cases where any contained data requires a transformation in the unwrapping. The most common transformation being data decompression. As an example working on a WARC file will not create any temporary files for the records since the data is available directly in the WARC file. On the other hand working on GZip compressed files will create temporary files for the GZip entries but not for the data since they are available directly in the temporary GZip entry files. Zip and tar.gz files would also create temporary files, where tar, html, gif, jpeg and pdf files would not.

The maximum size of the data chunk held in memory is defined by the property BufferSize of the JHOVE2 Invocation object. If no such object is defined in the JHOVE2 configuration, the default value applies: 131072 bytes. To adjust the value, one needs to modify the main JHOVE2 configuration file (`config/spring/jhove2-config.xml`) to define an Invocation object and set the property value. Alas, the JHOVE2 command line tool (`jhove2.sh/bat/cmd`) ignores the Invocation objects defined in the configuration and creates its own, populated from the command line arguments. The "-b" option corresponds to the BufferSize property.

As temporary file creation, writing and reading rely on slow disk I/Os, it is recommended, if characterization performance (in CPU time) is important, to have the temporary file directory pointing to a RAM disk. See your operating system documentation for the procedure to create a RAM disk.





As the number of GZip entries in a GZip file and the uncompressed size of each of them may greatly vary, it is recommended that the amount of disk space available on the file system hosting the temporary file directory may be at least five (5) times the size of the largest GZip file to characterize (to cope with the worst case scenario of a GZip file containing a single GZip member with a compression ratio of 80%).

## 7 Implementation Notes

The GzipModule uses the Java Inflater class directly to decompress deflated data. Reading the headers and surrounding data is handled by the module itself.

In order to support GZip compressed ARC and WARC files the GZip module is required to be aware of any ongoing reading of multiple entries by the same ARC/WARc reader. While a GZIP-compressed TAR file is made of a single GZIP member, the tarball itself, a compressed ARC/WARC file is made of concatenated individually compressed ARC/WARC records.

In the case of ARC files, only the first record (the version block) bears the ARC file signature (the “filedesc://” URI scheme). So any subsequence ARC record will not be processed correctly unless the GZip reader uses the same reader as for the first record. Hence, only the first record shall be submitted for identification; subsequent records are directly handed over to the format module object associated to the first record source.