Long term preservation –
from prototype to tools in operation
Final report

Långtidsbevarande –
från prototyp till fullfärda verktyg
Slutrapport

Kungl. biblioteket, National Co-operation
Dnr 63-141-2005

Stefan Andersson,
Electronic Publishing Centre, Uppsala University Library

June 2008
Table of Contents

Summary .................................................................................................................. 3
Background .............................................................................................................. 3
Overall goals ............................................................................................................ 4

Results ..................................................................................................................... 6
Staffing ...................................................................................................................... 6
Binary distribution, documentation and source code ........................................... 7
Presentations and public dissemination of results ................................................. 8

Evaluation ............................................................................................................... 9

Technical deliverables .......................................................................................... 11
Overview of tools and applications .................................................................... 11
Format registry ....................................................................................................... 11
Metadata mapping tool ......................................................................................... 13
The contract tool ................................................................................................... 13
DIDL and METS tools .......................................................................................... 14
  Binary packager .................................................................................................. 14
  Resource explorer ............................................................................................... 16
SIP Manager ............................................................................................................. 17
  Download and use of the application .................................................................. 18
  GUI usage (Graphical User Interface) ................................................................. 18
  CLI usage (Command Line Interface) ................................................................. 25

Testing .................................................................................................................... 28
Description of a complete build/packaging process ........................................... 28
Input/output samples ............................................................................................ 29
Testing reception of packages at the National Library ........................................ 29
Summary

The focus of this project is the interplay between producer/local archives and the National Library of Sweden (KB). The project’s goal is to facilitate the management and processing of digitally created material at both local and national levels.

The project entitled “Long term preservation – from prototype to tools in operation” (Långtidsbevarande – från prototyp till fullfärddiga verktyg) is based on results achieved within the SVEP project – Sub-project 2. The aim of SVEP SP2 was to create basic preconditions for, and build up parts of, an infrastructure that would make it possible to guarantee access to documents in a longer time perspective. This infrastructure consists of persistent identifiers services and services based on standardised methods and tools for transfer of data between local producers and archives at KB. This infrastructure has been successfully tested on a smaller scale. In order to move up to full scale production it is necessary to further develop the prototypes that were established in SVEP2 into fully developed tools.

This project is not only of national importance, there is also great interest and considerable expectations that Sweden will contribute to international development within this field.

Background

Electronic publishing at Swedish universities and university colleges has begun to assume more discernable forms as concerns organisation, technical solutions and business models. In 2003-2004 many projects and pilot projects have transferred from experimental services to stable services with long term solutions in focus. One of the contributing factors to this trend is the impact of the SVEP Project where focus has included long term issues in a broader national and international context.

One important issue with a direct connection to electronic publishing is how to guarantee, in the best possible manner, access to published documents both now and in the future and how best to preserve these documents in the long term. Researchers and students that publish electronically do not only wish to have a fixed point that they can refer to, but also require a certain guarantee that it will be possible to find, read and understand their research results in the future as well.
Making documents accessible and possible to use in the future is in the interest of national libraries and research libraries. Consequently it is absolutely vital to examine, identify and implement practical solutions concerning how these problems can be solved in cooperation between just these actors.

Within the framework of the SVEP Project – Sub-project 2 focusing on long term preservation – the basis of a possible infrastructure has been examined, tested and laid where the interplay between the local producer at a university and the National Archives at KB are centre stage.

During the course of the SVEP Project, our hypothesis that this infrastructure facilitates and rationalises the collection of documents and relevant metadata for KB’s Archives was confirmed. It was also proved that this infrastructure is necessary if delivery from document producers is to be carried out in a controlled manner, regularly and in standardised form. Results achieved so far have caused a great deal of interest internationally and the project has built up a global network that contributes to placing the Swedish development in an international context but also ensures that project results are applicable in a wider perspective.

In order to ensure that common standards and recommendations are followed – which increases future ability to access, utilise and understand archived documents to a substantial degree – a purpose-designed tool is necessary that is easy to apply. When suitable tools based on the prototypes established within the SVEP project are fully developed, this type of document processing will be economical and realistic.

This project – “Long term preservation – from prototype to finished tool (SVEP – SP2)” – ensures that the preliminary results from SVEP2 are implemented and contribute to the establishment of a functioning infrastructure.

Overall goals

This project is a further development of the prototype of infrastructure for long term preservation of documents that was established and tested within the SVEP Project, SP2. SVEP SP2 has laid good foundations for further development.

The success of the Resolution Service tool that has been in full production since January 2005 and is published as an open source code under a GNU license proves that there is great interest and a wide application area for the results of projects of this type.

Within this continuation project the following prototypes will be developed in order to establish a finished tool that will be published under an open source code license:
1. Format and Data Environment Register
2. Tool for packaging and transfer of objects to be preserved at KB based on standards for complex object format (METS, DIDL)
3. Tool for checking and reception of objects at the local/national archives\(^1\)

\(^1\) This project summary is taken from a supplement to the project proposal of 2005-02-15 written by Eva Müller
Results

This report provides the formal conclusion of the project. It has been compiled by Stefan Andersson from the Electronic Publishing Centre (EPC) at Uppsala University Library based on the documentation provided by the old project wiki (which is now closed down at Uppsala University Library and will be transferred in its entirety to Kungl. biblioteket). The bulk of the descriptive information from the wiki is now also included in this report and all the tools which were developed in the project are still available for download (see detailed information below).

The available background material consisted of:

- A summary of the project proposal by Eva Müller (as cited above)
- An evaluation report by Herbert Van de Sompel
- Descriptions of the technical deliverables by Ronnie Kolehmainen
- One public project presentation by Ronnie Kolehmainen and one poster/abstract by Ronnie Kolehmainen et al. presented at the ETD 2007 conference (see below for full details on these)

Staffing

The project was run by Eva Müller (from the Electronic Publishing Centre at Uppsala University Library) who was responsible for the initial project proposal as well as the project management from the start in 2005 until February 2007 when she left Uppsala University Library for a position at Kungl. biblioteket. The technical and organizational solutions in the project have initially been conceived by Eva Müller.

In the original planning it was stated that:

As the work primarily concerns technical development, the project will be run by UU [Uppsala University Library] in collaboration with experts at KB [Kungl. biblioteket]. This year KB has started up a number of in-house projects in which preservation issues are in focus. These in-house projects dovetail well with this proposed project, at the same time as they guarantee that KB staff will have time reserved for cooperation between projects.

A reference group consisting of representatives for the universities who are members of SVEP will be established for this project. The preliminary results achieved within the SVEP Project on which this application is based
have caused a great deal of positive interest internationally and have opened up opportunities for broad international cooperation. Consequently these tools will not only be developed in cooperation with KB, but also in collaboration with a number of international actors working with preservation issues at both institutional archives and national libraries.²

As planned the technical deliverables were developed by Uppsala University Library. The tools were principally built by Ronnie Kolehmainen at the EPC assisted by Uwe Klosa from the same department.

Two deliverables, according to the project contract, remained to complete when Eva Müller left Uppsala University Library:
1. the tool for checking and reception of objects at the local/national archives (item 3 in the list of expected prototypes above)
2. the final report

It was then up to Uppsala University Library, as the project owner, to finish these. The end of the project had initially been set to March 2007. In the period between February and March 2007 the technical development of the tool for checking and reception of objects was accomplished by the EPC. In addition to that a few improvements, particularly regarding the graphical user interfaces, were needed as well as some testing of the tools.

All the technical deliverables are described in detail in a separate section below as well as a full test case, including sample input and output files, describing a complete build/packaging process.

Regarding the preparation of the final report it was originally suggested by Eva Müller herself in January 2007 that it would be written by her - in co-operation with the other partners - after her move to Kungl. biblioteket, but as it turned out this proposal was never effected. The National Library has since then referred to lack of time and resources, as well as other priorities. Therefore this final report has now been put together by Stefan Andersson at the EPC without any further involvement from Eva Müller or Kungl. biblioteket.

No documentation is available regarding the reference group and the number of international actors that were to co-operate in the development according to the initial project proposal, hence their impact on the results cannot be reviewed here.

**Binary distribution, documentation and source code**

The binary distribution of the tools can be downloaded from [http://svep.epc.ub.uu.se/pub/sip-tools/sip-tools.zip](http://svep.epc.ub.uu.se/pub/sip-tools/sip-tools.zip). Additional files, such as

---

² In supplement to the project proposal of 2005-02-15 by Eva Müller, p. 3
api documentation and source code can be downloaded as well from http://svep.epc.ub.uu.se/pub/sip-tools/.

All technical deliverables (including the source code) have also been transferred to the Digital Library at the National Library and may be published under an open source code license by them, in accordance with the original project goal, if the Digital Library so desires.

Presentations and public dissemination of results

In line with the project proposal the results have been presented publically at two conferences: Mötesplats Open Access, KTH, Stockholm, 12–13 April 2007\(^3\) (presentation) and ETD 2007, 10th International Symposium on Electronic Theses and Dissertations, Uppsala, 13–16 June 2007\(^4\) (poster session).

\(^3\) See: [http://epc.ub.uu.se/files/ronnie_kolehmainen_070413.pdf](http://epc.ub.uu.se/files/ronnie_kolehmainen_070413.pdf)

An external evaluation of the project was done by Herbert Van de Sompel from the Research Library of Los Alamos National Laboratory and published in April 2007. Van de Sompel’s report covers the project activities up to December 2006 (which essentially includes all results and deliverables except the tool for checking and reception of objects which, as previously mentioned, was developed early in 2007).

The evaluation report is based on basically the same background material as listed in the beginning of this report. In addition to this the reviewer spent a full day in December 2006 at the EPC in Uppsala meeting Eva Müller, Ronnie Kolehmainen, and Uwe Klosa from the project team and he also took part in a two-day workshop called Long term access to digital resources: current strategies and tools at Kungl. biblioteket where the project was presented. This workshop was open to selected participants only.

The evaluation is focused on “providing food for thought when considering next steps and eventual reorientations”. Project feedback is given in twelve points where specific items are being discussed in each one. In retrospect this project did not contribute to the national and/or international development of these issues in the way that was anticipated by Eva Müller in the original proposal, and the reasons behind this are transparent in Van de Sompel’s comments.

The fundamental issue, as the reviewer points out (cited below), was clearly that the communication both within the project group and between the project and the world around left a lot to desire.

Clear and broad communication about project goals, approaches and results should be considered essential. This is a proven approach to solicit peer commentary, but it also is an approach to have a constructive impact on the evolving international thinking regarding the Legal Deposit and Digital Preservation challenges. Telling from the nature of the in-person communications (both written and verbal) regarding the Project with the reviewer, it is clear that there is ample room for improvement with this respect.

As a result of this it was never really established in which context the tools would actually be used.

---

For example, it was well into the first day of the Seminar (after the full-day visit to the Uppsala University Library), that some light was shed on basic questions such as “who is the user”, “what is the operating context”, “what is the source/target”, “what is the anticipated workflow, and how do the tools fit in”, etc.

We suggest that clarifying this operating context in close co-operation with the prospective users of the tools would be a suitable revised starting-point if any follow-up of this project should be considered.

Such a clarification would also obviously influence the more technical aspects of the project where Van de Sompel raises some question marks, for example the focus on high level descriptive metadata, the use of URN:NBN, the actual transfer mechanisms, and the data model for the representation of the object in the packages. If the basic questions, as cited above, were the basis of the project from the beginning most of these matters could be rather easily identified and the different technical choices would have been based on actual workflows and users.
Technical deliverables

Overview of tools and applications

The following tools and applications have been developed in the project:

- Format registry
- Metadata mapping tool
- Contract tool
- DIDL and METS tools
- SIP Manager

In addition to these a resolution service ⁶ was developed in a previous project and is now available under the GNU General Public License (GPL).

URN:NBN, which stands for Uniform Resource Name: National Bibliographic Number, is a unique and consistent identifier for electronic publications.

The producers are recommended to use persistent identifiers. KB can provide URN:NBN and ISSN (and sometimes ISBN) for electronic resources. Objects not having URN:NBN will also be given URN:NBN by KB for archive purposes.

A prototype of a resolution service has been developed by the Electronic Publishing Centre at the Uppsala University Library and has been implemented at KB. One of the activities of this project is to further develop this service in corporation with the Uppsala team ⁷.

Format registry

The format registry contains descriptive data of file formats, software, and vendors. Some of the data is now taken directly from PRONOM, the online registry of technical information which was developed by the Digital Preservation Department of the UK National Archives.

---

⁶ See: [http://www.kb.se/ENG/dl/urnnbn.htm](http://www.kb.se/ENG/dl/urnnbn.htm)
⁷ See: [http://epc.ub.uu.se/niwiki/pmwiki.php/ResolutionService/ServiceV3](http://epc.ub.uu.se/niwiki/pmwiki.php/ResolutionService/ServiceV3) for documentation and downloads of the application.
PRONOM is a resource for anyone requiring impartial and definitive information about the file formats, software products and other technical components required to support long-term access to electronic records and other digital objects of cultural, historical or business value.

When initially developing the registry PRONOM was not available but another reason for creating a local application was also that it was preferred to have the possibility to include descriptions in Swedish.

The application developed in this project is publicly available on the web. It supports harvesting of records using OAI-PMH. There are Web Services which can be accessed over SOAP. Current methods available are:

- getMimeType
- getFileExtension
- getDescription
- getVersion

Public web interface to the format registry

---

8 See: http://www.nationalarchives.gov.uk/PRONOM/
9 See: http://svep.epc.ub.uu.se/ffr/
10 The OAI-provider is available at http://svep.epc.ub.uu.se/ffr/OAI?verb=Identify
11 See: http://svep.epc.ub.uu.se/ffr/ws/FFRWebService?wsdl
The schema used for the storage of registry data is also published together with its documentation\textsuperscript{12}.

**Metadata mapping tool**

This tool facilitates a semantic mapping between metadata formats provided by data providers (repositories) and an internal vocabulary used for definition of specific metadata elements (definition list). This mapping is used as a base for transformation to a harmonized metadata format (schema).

It provides a base for a control mechanism and feedback to the data provider (repository) on the quality of the delivered metadata. Output of the tool is a mapping specification (XSL file).

Its GUI enables it to
- assign a semantic label (based on the definition list) to each metadata element
- store the relation between the label and the original elements
- provide control of metadata
- provide feedback and possibility to enhance metadata

The output is controlled against the specification on metadata coming from the *Contract tool*.

The metadata mapping tool can be tested on the web\textsuperscript{13}. The start page is not necessary if the tool is invoked from another application. In that case the XML document or the URI will be passed straight to the next page.

The tool can
- parse an XML file
- display the values in the elements in a table
- map an XPath\textsuperscript{14} to a label from the Contract tool
- test if the XML metadata file complies with the different contract levels

**The contract tool**

The purpose of this tool is to create and maintain a specification which helps to make sure that the data and digital objects delivered by data providers (repository) satisfies the requirements specified by the archive.

\textsuperscript{12} See: at \url{http://epc.ub.uu.se/schema/ffr/2005/}
\textsuperscript{13} See: \url{http://svep.epc.ub.uu.se/maptool/}
\textsuperscript{14} See: \url{http://www.w3.org/TR/xpath}
The tool, which also can be tried out on the web\textsuperscript{15}, is integrated with the packing workflow and provides both a control mechanism and feedback to the data provider (repository) on the quality of the delivered data (objects). It supports the possibility to store several different specifications.

Its GUI enables it to

- edit stored specifications easily
- connect conditions with a specific level of preservation (specified by the archive)
- easily create new specifications
- connect each specification to a specific level of preservation
- extend the number of preservation levels supported by the tool

**DIDL and METS tools**

Two tools have been developed for the examination and manipulation of resources within DIDL and METS XML files: the binary packager and the resource explorer.

**Binary packager**

The tool\textsuperscript{16} is designed to read a DIDL\textsuperscript{17} or METS\textsuperscript{18} XML input stream and augment the contents of referenced resources (i.e., element <Resource> with a ref attribute). The resources may be images, PDFs, or any binary files.

Basically, it is an SAX\textsuperscript{19} parser that copies the input stream to an output stream as it is read until a Resource-element is identified. When this element is encountered, a new input stream is opened (from ref) and that stream is transformed into a Base64-encoded character stream. If desirable (for the DIDL format), the byte stream may be also be compressed using either of the algorithms, \textit{gzip} or \textit{deflate}, before the Base64-encoding occurs.

The tool only reads and writes streams, i.e. only small portions of the documents are held in the memory. Thus memory consumption is reduced to a minimum, even if the DID XML or its referenced contents are very large in byte size. In theory, the tool can run on a simple PC with a very small amount of memory, and still handle documents with sizes of gigabytes.

In short the tool

---

\textsuperscript{15} See: \url{http://svep.epc.ub.uu.se/contract-tool/}
\textsuperscript{16} See: \url{http://epc.ub.uu.se/didl/package/}
\textsuperscript{17} See: \url{http://xml.coverpages.org/mpeg21-didl.html}
\textsuperscript{18} See: \url{http://www.loc.gov/standards/mets/}
\textsuperscript{19} See: \url{http://www.saxproject.org/}
• takes URI of an XML file as a parameter/argument (XML contents can be POSTed directly to the servlet, but in this case a size-limitation is applicable). For examples see: DIDL\textsuperscript{20} and METS\textsuperscript{21}.
• parses XML, and for each \texttt{<Resource>} element with a “ref” attribute downloads contents and stores the contents inline as a base64-encoded xml character stream.
• for DIDL, if optional parameters/arguments \texttt{gzip} or \texttt{deflate} are true, the contents will be compressed as well before base64-encoding.
• can set an additional parameter - also optional - \texttt{xmlbinary}, to true to also base64-encode injected XML resources as binary content instead of regular text/xml.

**Example with DIDL.**

Before:

\begin{verbatim}
<didl:Item>
  <didl:Descriptor>
  </didl:Statement mimeType="text/plain">Background image</Statement>
  </didl:Descriptor>
  <didl:Component>
    <didl:Resource mimeType="image/gif" ref="http://epc.ub.uu.se/uu_img/whiteedge.gif"/>
  </didl:Component>
</didl:Item>
\end{verbatim}

After:

\begin{verbatim}
<didl:Item>
  <didl:Descriptor>
  </didl:Statement mimeType="text/plain">Background image</Statement>
  </didl:Descriptor>
  <didl:Component>
    <didl:Resource mimeType="image/gif" contentEncoding="gzip" encoding="base64">R0lGODlhtAAKAIAAP///8zMzCH5BAQUAP8ALAAAAAC0AAoAAAIXhI+py+0Po5y9rmCz3rz7D4biSjYzqbyrbuC6toTNf2jec6MO/+DwwKc72h8RggAA A7......</didl:Resource>
  </didl:Component>
</didl:Item>
\end{verbatim}


Example with METS.

Before:

```xml
<mets:fileGrp USE="full">
    <mets:file ID="CAE4142-01-full" GROUPID="CAE4142-01"
        MIMETYPE="image/gif" SEQ="1" CREATED="2004-07-22T14:00:00"
        USE="full">
        <mets:FLocat LOCTYPE="PURL"
    </mets:file>
</mets:fileGrp>
```

After:

```xml
<mets:fileGrp USE="full">
    <mets:file ID="CAE4142-01-full" GROUPID="CAE4142-01"
        MIMETYPE="image/gif" SEQ="1" CREATED="2004-07-22T14:00:00"
        USE="full">
        <mets:FContent>
            <mets:binData>
R0lGODlheAN8fQfAABEQsLCsMTExbGyMjIysrKzMzw8PENDQ0xMTFNTU1tbW2NjY2tra3R0dH8fI0d4yMjJSUlJubm6Ojo6ysrLS0tLy8vMPDw8zMzNPT09zc3OPj4++vK......
            </mets:binData>
        </mets:FContent>
    </mets:file>
</mets:fileGrp>
```

Resource explorer

This tool\(^{22}\) is designed to display resources in DIDL or METS XML, and to extract individual resources so they can be viewed in a web browser or saved into a local file.

Similar to the Packager, this tool is working with streams so that documents (DIDs, METS files, or their resources) are never held entirely in the memory - only a small portion at a time.

In short it

- takes URI of an XML file as its argument. For examples see: DIDL\(^{23}\) and METS\(^{24}\).

\(^{22}\) See: http://epc.ub.uu.se/didl/explore/

• renders a web page of a tree view with a hypertext link for each 
  <Resource> or <file> element that can be either extracted from the XML 
  file itself or downloaded from the web. The link contains a description 
  of the resource, and its MIME content type.
• Resources are either linked to directly, or extracted from the XML on 
  demand.
• If a resource is compressed it will be decompressed as well, unless the 
  User-Agent claims it will do it. Binary resources handled are:
  • Uncompressed, base64-encoded
  • Compressed with gzip (GNU zip) and base64-encoded
  • Compressed with deflate and base64-encoded

DIDL Resource Explorer

Public web interface of DIDL resource explorer

SIP Manager

This is the main application for creating, transferring and managing SIP's - 
Submission Information Packages - for archiving purposes. The application 
uses all the above tools created for this project:
• Format Registry
• Metadata mapping tool
• Contract tool
• DIDL and METS Tools

Other components used:

packaged.xml
• DROID (Digital Record Object Identification), a software tool developed by The National Archives to perform automated batch identification of file formats\textsuperscript{25}.
• Jakarta Commons-Net FTP protocol\textsuperscript{26}
• Jakarta POI-HSSF Horrible Spread Sheet Format (creation of xls files)\textsuperscript{27}

Download and use of the application

1. Download the binary distribution from http://svep.epc.ub.uu.se/pub/sip-tools/sip-tools.zip. The ZIP archive includes example XML files (from the DiVA repository) and a metadata mappings file.
2. Extract (unzip) the archive. A subfolder named sip-tools will be created, with following contents:
   • etc configuration files
   • example-files a few sample files from the DiVA repository (in DiVA Document Format\textsuperscript{28})
   • lib the additional libraries needed by the application
   • logs log files will go here
   • README.txt short instructions
   • sip-tools.jar the main executable

Additional files, such as api documentation and source code can be downloaded as well from http://svep.epc.ub.uu.se/pub/sip-tools/.

GUI usage (Graphical User Interface)

Start the application either by double-clicking sip-tools.jar - or, if this is not working - enter the directory (cd /some/path/to/sip-tools) from a command shell and type java -jar sip-tools.jar.

Preparation

Before running either Build or Receive it is advised to store some general settings within the application. These settings are found from the menu: Menu -> Options...

\textsuperscript{25} See: http://droid.sourceforge.net/wiki/index.php/Introduction
\textsuperscript{26} See: http://commons.apache.org/net/
\textsuperscript{27} See: http://poi.apache.org/hssf/index.html
\textsuperscript{28} See: http://epc.ub.uu.se/ddf/
The location for metadata and file format contracts are entered in URI syntax. A file on the web starts with http:// and a local file on the hard drive should be prefixed by file://

Examples:
(Web) http://www.example.com/xml/contract.xml
(Unix) file:/home/user/xmlfiles/contract.xml
(Windows) file://C:/Documents%20and%20settings/user/projects/data/contract.xml

SMTP Host: the mail service provider, usually smtp.yourdomain.com.
Email From: the from address used in emails sent from this application.

Packaging
Build SIP package(s).
1. Select input file(s) by either browsing and selecting an xml file or by selecting a folder (in which case all xml files in the folder will be processed).
   - For testing, the example-files folder in the distribution can be used.
2. Select a mappings file for the metadata elements; a result of the Metadata mapping tool.
   - For testing, the file etc/diva2kb.xsl can be used.
3. Select an output folder for the created packages. Preferably this folder should be empty.
   - For testing, create a new folder.
4. Choose SIP format. Currently two package formats are supported by the tool: METS XML and ZIP.
   - METS is the submission format and should be used by the provider.
5. Choose checksum(s) for the package(s).
   - Separate MD5 file is recommended for submission.
6. Press Build button.

NB: values from 1-3 above are stored in a preferences file in the users home directory and are read each time the application is started.
If a metadata file does not have any referenced resources, the user will be prompted for additional resources to be included in the package (by a filechooser dialog).

<table>
<thead>
<tr>
<th>File / Object</th>
<th>Metadata (basic)</th>
<th>Metadata (extended)</th>
<th>File Format (basic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>urn:nbn:se:kth.diva-3277.xml</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>urn:nbn:se:kth.diva-3376.xml</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>urn:nbn:se:uu.diva-6155.xml</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>urn:nbn:se:uu.diva-7291.xml</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

If one or more contracts are specified in application settings the validation results will open in a new window after a build/packaging process is completed. This enables the producer to find packages which will not be accepted by the archive.

The validation results can be saved in various formats. For human processing xls format is recommended (for use with MS Excel or OpenOffice.org).

**Transfer**

Transfer files (packages and optional checksum files). A very simplistic FTP client is provided within the application so that files created with the tool can be transferred to the archive.
Enter properties for the TP server:

1. **IP/Host** the server address:
   - IP or hostname (followed by :port if other than standard port 21)
   - Use passive mode click to enter passive mode before transferring any files (recommended if behind a firewall)
   - Username the FTP user account
   - Password the password for login
   - Remote Folder if given, the FTP browser will enter this directory immediately after a successful connection

2. Click Connect / Transfer in order to connect with the FTP server. If/when a connection has been established a new window - FTP Browser - will open.
3. After navigating to the destination folder, press **Transfer** to send all files within the output folder (selected from **Build** tab) from the client to the FTP server.

**Receive**

Validate SIP XML packages and create archive files.
1. Select input file(s) by either browsing and selecting an xml file or by selecting a folder (in which case all xml files in the folder will be processed).
   - If the input file/folder textfield is empty when starting the application, it will be pre-filled with the value from output folder in Build tab.
   - For testing, build package(s) from Build tab and use that output folder for indata.
2. Select an output folder for the created packages. Preferably this folder should be empty.
   - For testing, create a new folder.
3. Choose archive package format. Currently two package formats are supported by the tool: METS XML and ZIP.
   - The METS format is the same as the submission format.
   - The ZIP format has following entries:
     - sip-metadata.xml The submitted METS file, with resources removed (replaced with href's).
     - mods-metadata.xml The bibliographic metadata, in MODS xml.
     - One file for each resource extracted from the original METS file. The original filename of the resource is used when applicable).
4. Choose checksum(s) for the package(s).
5. Press Run button.

Validation results will always open in a new window after a build/packaging process is completed. This enables the archive to find identify packages which will not be accepted due to lack of metadata, unsupported file types, checksum mismatch etc.
Columns of validation results table:
- Metadata quality (if a contract is specified in application settings). There will be one column for each level found in contract.
- Supported file types of resources (if a contract is specified in application settings).
- Resource integrity. This is a comparison of the resource checksum(s) within the submitted package versus calculated checksums from extraction of package.

The validation results can be saved in various formats. For human processing, XLS format is recommended (for use with MS Excel or OpenOffice.org).

**Resource Explorer**
The resource(s) within the METS XML package can be viewed and extracted with the built-in SIP Resource Explorer (*Tools -> SIP Resource Explorer* in the menu).

1. Use the *Browse* button to select a local file, or enter the URL of a web file in the text field.
2. Press *Explore* button to render a tree view of resources.

The Explorer tool can also view DIDL XML.

**CLI usage (Command Line Interface)**
Start application from commandline by invoking
java -jar sip-tools.jar cli PARAMS [OPTIONAL PARAMS]

PARAMS:
--input The input file or folder.
--output The output folder.

OPTIONAL PARAMS:
--xsl Mappings file.
--mdcontract Metadata contract file or URI.
--ffcontract File formats contract file or URI.
--stdout File for UI messages (defaults to STDOUT).
--action Action command: build|extract (defaults to 'build').
--ftp.server The FTP server address, hostname|IP[:port].
--ftp.user FTP user account.
--ftp.password FTP password.
--ftp.folder The remote folder on server where to store files.
--validation Validation results output format: txt|csv|html|xls.

Example: build and transfer
In following example all xml files in example-files folder will be processed. The built packages - METS xml - will be written into folder output. After the run is completed all .xml and .md5 files will be transfered to the FTP server ftp.kb.se and a report file in xls format will be written in current folder. All output from the program goes to file run.log.

java -jar sip-tools.jar cli \
   --input=example-files \
   --output=output \
   --xsl=etc/diva2kb.xsl \
   --ffcontract=contracts/accepted-fileformats.xml \
   --mdcontract=contracts/metadata-obligations.xml \
   --ftp.server=ftp.kb.se \
   --ftp.user=uu123 \
   --ftp.password=xxx123 \
   --ftp.folder=/data/uu \
   --validation=xls \
   --stdout=run.log

FTP transfer is only possible when all four ftp.* parameters are given. If one or more ftp.* parameters are omitted there will be no file transfer attempt.
Example: receive
The following example will process all .xml files in folder /data/uu, validate metadata and file formats, and create ZIP files in folder /storage/uu. All output from the program goes to the console.

```
java -jar sip-tools.jar cli \
   --input=/data/uu \
   --output=/storage/uu \
   --ffcontract=http://svep.epc.ub.uu.se/pub/accepted-fileformats.xml \
   --mdcontract=http://svep.epc.ub.uu.se/pub/metadata-obligations.xml \
   --action=extract \
   --validation=xls
```

The parameter --validation=xxx will produce a file in current directory: validationresults-yyyy-MM-dd-HH.mm.ss.xxx (xxx will be replaced with either txt, csv, html or xls).

Regardless of the parameter --stdout= the normal built-in logging will be written to file logs/sip-tools.log.
Testing

All the tools created in the project have been tested by the EPC according to the following complete build/packaging process with sample results from the testing also available on the web.

Description of a complete build/packaging process

For each metadata file a number of tasks are executed in order to create a package with bibliographic metadata, technical metadata and files. The steps are as follows:

1. The metadata is transformed into a simple internal xml format, using a stylesheet exported from the metadata mapping tool.
2. The internal metadata structure is validated against the metadata contract exported from the contract tool.
3. The (bibliographic) metadata is transformed into MODS\textsuperscript{29} format and is placed within a METS “master”, along with references to external resources, represented by METS:file elements with \textit{href} attributes pointing to objects on the web or local file system. From this point this “master” file is used in the remaining steps of the packaging process.
4. The METS data is fed into the sip-packager which downloads referenced resources and augments the METS container with binary data.
5. The resources are analyzed in order to gather additional technical information:
   1. Each resource is extracted (with the help of \textit{sip-explorer}) into a temporary folder. This is necessary for DROID - see below - which does not work properly on streams.
   2. During the extraction of a resource a number of checksums are calculated from the output streams; for now MD5, SHA-1 and CRC-32.
   3. DROID (hopefully) identifies the file format identifier, PUID, for PRONOM technical registry.
   4. A query is sent to FRS (a file format resolution service) with the PRONOM identifier. FRS responds with the identifier used in File

\textsuperscript{29} See: \url{http://www.loc.gov/standards/mods/}
Format Registry (FFR) at EPC. The resolution service is accessed as a Web Service using SOAP.

5. The EPC identifier is used in two calls to FFR, one to export technical metadata for the format, another to get the MIME type. The former is a simple fetch by URL, the latter, again, using a Web Service at FFR.

6. The METS master is enriched with the technical data gathered from the steps above. Checksums, DROID identification report and FFR data are inserted as XML fragments into the METS:techMD section for each resource. The mimetype for the resources are added as attributes to each METS:file element.

Input/output samples

The resulting METS file may look like this\(^30\) (XML - 1.1 MB). It is a real output example created from a metadata file from DiVA\(^31\) (XML - 18 KB) which is processed with metadata mappings\(^32\) (XSL - 19 KB, also included in the download archive) from the metadata mapping tool.

The URI of the METS file above may be entered in the SIP Explorer web form\(^33\) to view and/or extract the packaged resource, a PDF file.

Testing reception of packages at the National Library

Attempts to test the whole process and actually also transfer packages created like the above example to the National Library were carried out in the spring of 2007 by the EPC. As no response from the Digital Library regarding the test packages was ever received the results on the National Library side are unknown.

---

\(^30\) See: [http://svep.epc.ub.uu.se/pub/sip-urn_nbn_se_kth_diva-3177.xml](http://svep.epc.ub.uu.se/pub/sip-urn_nbn_se_kth_diva-3177.xml)


\(^32\) See: [http://svep.epc.ub.uu.se/pub/diva2kb.xsl](http://svep.epc.ub.uu.se/pub/diva2kb.xsl)

\(^33\) See: [http://epc.ub.uu.se/didl/explore/](http://epc.ub.uu.se/didl/explore/)