



D21.3 Finding Aids Toolkit Installation, Deployment and User Manual

| | | |
|----------------------------|---|---------|
| Work package | WP21 Services/Toolkits Development and Adaptation | |
| Task | T21.8 Finding Aids Toolkit Development | |
| Author (s) | Wim De Smet | GIM |
| Authorized by | Name Surname | Company |
| Reviewer | Name Surname | Company |
| Doc Id | | |
| Dissemination Level | CONFIDENTIAL/PUBLIC | |
| Issue | 1.0 | |
| Date | 28/02/2014 | |


Abstract:

This document represents the Deployment and User Manual for the Finding Aids Toolkit developed in the frame of SCIDIP-ES project and contains all useful information on how to install, configure and use the software.

Document Log

| Date | Author | Changes | Version | Status |
|------------|-------------|--|---------|-----------|
| 13/02/2014 | Wim De Smet | Created base document from D21.4 and updated for the M30 release | 0.1 | Draft |
| 28/02/2014 | Wim De Smet | Added Section 2 and minor changes to grammar and style. | 1.0 | Published |

TABLE OF CONTENTS

| | | |
|---|--|-----------|
|  | SCIDIP-ES SCIENCE DATA INFRASTRUCTURE FOR PRESERVATION - EARTH SCIENCE | 1 |
| 1 | INTRODUCTION | 5 |
| 1.1 | PURPOSE AND SCOPE | 5 |
| 1.2 | WHO SHOULD READ THIS DOCUMENT | 5 |
| 1.3 | SYSTEM CONTEXT | 5 |
| 1.4 | LICENSE INFORMATION AND TERMS OF USE | 5 |
| 1.5 | DOWNLOAD INFORMATION | 5 |
| 2 | SOFTWARE DESIGN | 6 |
| 3 | INSTALLATION AND CONFIGURATION GUIDE | 6 |
| 3.1 | OVERVIEW | 6 |
| 3.2 | WEB FRONT | 6 |
| 3.3 | FINDINGAIDSERVER | 6 |
| 4 | USAGE GUIDE | 6 |
| 4.1 | WEB FRONT | 6 |
| 4.1.1 | NAVIGATION | 7 |
| 4.1.2 | INTEGRATION WITH METADATA CATALOGUES | 8 |
| 4.2 | REST SERVICE INTERFACE | 8 |
| 4.2.1 | {FAS-URL}/REST/GET/\$CPID | 9 |
| 4.2.2 | {FAS-URL}/REST/SEARCHRIL/\$SEARCHTERM | 9 |
| 4.2.3 | {FAS-URL}/REST/SEARCHMANIFEST/\$SEARCHTERM | 9 |
| 4.2.4 | {FAS-URL}/REST/RIN/\$CPID?FILTER | 9 |
| 4.2.5 | {FAS-URL}/REST/RESETCACHE | 9 |
| 5 | REFERENCE MANUAL | 9 |
| 5.1 | KEYBOARD SHORTCUTS | 9 |
| 5.2 | COMMAND-LINE COMMANDS | 9 |
| 5.3 | PUBLIC APIS | 9 |
| 6 | TROUBLESHOOTING COMMON ISSUES | 10 |
| ANNEX A. | FIGURES AND TABLES | 11 |
| A.1. | LIST OF FIGURES | 11 |
| A.2. | LIST OF TABLES | 11 |



ANNEX B. TERMINOLOGY 11

1 Introduction

1.1 Purpose and Scope

This manual gives an overview of the Finding Aids Toolkit and describes its installation procedure and general usage.

1.2 Who should read this document

Users who wish to install and/or use the Finding Aids Toolkit.

1.3 System Context

The Finding Aids Toolkit (also known as Knowledge Browser) is developed as part of the SCIDIP-ES e-infrastructure to help connect the archived data with structured knowledge represented as a network of ReplInfo objects. ReplInfo is defined in OAIS¹ as the additional information that maps a data object into more meaningful concepts which help data consumers to understand and use the data object correctly. Finding Aids also enables users to navigate a ReplInfo network, which may contain ReplInfos distributed in different SCIDIP-ES ReplInfo registries, and explore the knowledge represented.

The concepts of ReplInfo, ReplInfo network and label as well as Manifests are described in Section 3 of the [D21.3] Master document.

1.4 License Information and Terms of Use

The SCIDIP-ES Finding Aids Toolkit is licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with the License. A copy of the License could be obtained at: <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.5 Download information

The recent stable source code could be accessed from SVN at *Sourceforge*. The URL to the svn trunk is: <svn://svn.code.sf.net/p/digitalpreserve/code/SCIDIP-ES/software/toolkits/FindingAidsToolkit>

Releases of the software could also be downloaded from the SCIDIP-ES maven nexus repository at: <http://nexus.scidip-es.eu/content/repositories/releases/eu/scidipes/toolkits/findingaids/>

¹ OAIS - <http://public.ccsds.org/publications/archive/650x0m2.pdf>

2 Software Design

The Finding Aids Toolkit is a two-component library, consisting of a webpage (the Web Front or WF, written in JavaScript) which provides the visual browsing interface to ReplInfo Registries, and a FindingAidsServer, a Java REST service which proxies the SCIDIP-ES Framework component (see Section 4.1) and translates its results into a JSON format, for easy parsing by the WebFront.

The Web Front (WF) API is designed in JavaScript, to provide an agile and light-weight web-based tool. An important security limitation in JS is the Same Origin Policy, which prevents data acquisition across different internet domains. There is an exception to this rule however: *JSONP*², or *JSON with Padding*. This technique requires the data requested in a JavaScript program to be a JSON (JavaScript Object Notation) object, encapsulated as a JS-script.

To provide this functionality, the FindingAidsServer (from which the WF requests its data) queries the Framework for the correct information from the registries and translates it into JSON-objects. The result is then wrapped as a JSONP-object and returned to the WF. The REST-interface provided by the FindingAidsServer is implemented through the REStEasy library, while the JSON translation is handled by the Jackson plugin.

3 Installation and Configuration Guide

3.1 Overview

This section describes how to install and configure the Web Front and FindingAidsServer.

3.2 Web Front

The Web Front consists of a single HTML page, *index.html*, declaring the structural layout of the Finding Aids Toolkit, and a library of JavaScript files. This entire package can be hosted by an HTTP-server (e.g. Apache, Joomla, etc.), no requirements are made on this environment.

Customization is done in the file *config.js*. The parameter **restURL** needs to be set to the location of the RegistryProxy URL, see next section.

3.3 FindingAidsServer

The FindingAidsServer is an Apache Tomcat webservice. On a running Tomcat server, deploy the file FindingAidsServer-{version}.war into the *webapps* folder. You may need to restart Tomcat if it does not support hot deployment.

4 Usage Guide

4.1 Web Front

The Web Front (WF) of the Finding Aids Toolkit is a GUI interface for browsing registries. Two layers lie between the WF and the registries:

² JSONP - <http://json-p.org/>

- the Framework which abstracts away the concept of different registries, and provides a single calling point for registry interaction, and
- the FindingAidsServer which calls the correct functions of the Framework and translates the results into a format which is easily parsable by the WebFront.

4.1.1 Navigation

To start up the Web Front in a browser, navigate to its base URL, and add a CPID as a URL parameter (e.g. *base_url?RIL-MERIS_L1_data*). The following view is then presented to the user.

The screenshot shows the Knowledge Browser interface with four numbered sections:

- 1**: A circular graph with 'MERIS L1 data' at the center. It is connected to several related terms: 'MAN-Missing_Struct...', 'MAN-MERIS_L1_DPM', 'MAN-MERIS_Vol1', 'MAN-DQC_MERIS_...', 'MAN-BEAM_MS_Wi...', and 'MAN-BEAM_MAC_O...'. Lines of different colors (green, red, blue) connect the central node to the peripheral nodes.
- 2**: A search bar with the text 'Keywords: Type product:'. Below it is a dropdown menu showing 'Replinfo Label'. There are 'Search' and 'Clear cache' buttons.
- 3**: A box labeled 'Results:' which is currently empty.
- 4**: A box labeled 'Resource technical details:' containing the following information:
 - > typecpid: RIL-MERIS_L1_data
 - > cpid: RIL-MERIS_L1_data
 - > status: APPROVED
 - > type: ril
 - > locations: ACS
 - > structural:
 - MAN-Missing_Structur.
 - > semantic:
 - AND
 - MAN-MERIS_L1_DPM
 - MAN-MERIS_Vol1
 - MAN-DQC_MERIS_N1_
 - > other:
 - OR
 - MAN-BEAM_MAC_OS_I
 - MAN-BEAM_MS_Windo.

The numbered parts of the screen are:

1. Browsing Circle:

This part of the screen contains the visual representation of the ReplInfo Network. The center node represents the current ReplInfo Label being inspected (eg. MERIS L1 data); connected Labels (eg. MAN-Meris_Vol1) are located on the concentric circle. These labels are clickable for navigation through the network.

The type of connection is encoded in the color of the connecting edges: links derived from a structural ReplInfo List are represented in green, semantic in red, and other in blue. The connection from a Manifest to the ReplInfoLabel which represents its type is colored gray.

As ReplInfo Networks can contain many ReplInfo objects; displaying a networks graphically will often result in a busy or dense-looking graph, therefore, only the immediate connections from the central node are shown. This avoids cluttered graphs. To give the user some indication of how the rest of the network is structured, a visual aid is presented in the form of an outstretching branch, where the thickness is an indication of how many further links that branch represents.

2. Search Option:

This section of the WF provides a simple search interface to the Registry (which uses queries embedded in URL parameters). A drop-down box restricts the search results by type: ReplInfoLabels or Manifests.

3. Resource Description:

In the Resource Description section, a quick overview of the central node (i.e. the currently selected ReplInfo object) of the Browsing circle is shown. This includes the description provided by the registry, plus a download-button if external data is associated with the selected resource.

4. Resource technical details:

A more detailed description of the central node (i.e. the selected resource) is presented in this view. It includes information about the CPID and TypeCPID, known locations of the resource, its status, and, if present, a tree-based graph depicting the ReplInfo Lists. These Lists include the *AND*- and *OR*-nodes as present in the registry. Clicking the other nodes, those with a CPID, have the same effect as clicking the nodes in the Browsing Circle.

4.1.2 Integration with Metadata Catalogues

Finding Aids provides a platform-independent web client which can be accessed using standard browser technology. This design facilitates integration of the Finding Aids Toolkit with the Metadata Catalogues of the Preservation Archive/s. An external catalogue application merely has to include a URL pointing to a given CPID, which is the starting element used in the WF, to integrate the Finding Aids' Registry browsing capability.

4.2 REST Service Interface

Communication with the FindingAidsServer is done through a REST-interface. A dynamic URL maps the methods and parameters to resources (e.g. ReplInfoLabels, Manifestsm etc.) Assuming the FindingAidsServer itself is hosted at *{FAS-URL}*, the following URL's provide the different functionalities:

4.2.1 {FAS-URL}/rest/get/\$CPID

This method returns the RepInfoLabel or Manifest with $\$CPID$ as its CPID as a JSON object. Information includes the CPID, displayname and type-CPID, its known locations and categories, status, and its structural, semantic and other RepInfoLabel lists.

4.2.2 {FAS-URL}/rest/searchRIL/\$searchTerm

This method calls the Framework's *searchForRILSMatching(\$searchTerm)* functionality, which searches all known registries for RepInfoLabels whose keywords match the searchTerm. A JSON-list containing the CPID's and displaynames of the results is returned.

4.2.3 {FAS-URL}/rest/searchManifest/\$searchTerm

Same functionality as the previous method, but this time the results are restricted to be of type Manifest.

4.2.4 {FAS-URL}/rest/RIN/\$CPID?filter

This method calls the *get* method, which fetches the RepInfoLabel or Manifest with CPID = $\$CPID$, and then recursively calls itself on each element in the result's structural, semantic and other RepInfoLabel lists. The resulting branching graph has a default depth limit of 4, as this method can run for a long time.

The WF is responsible for calling this method again from a RepInfoLabel or Manifest at the "edge" of the graph, i.e. those nodes where branching was culled because the default depth was reached. Although caching is performed during a search, in order to avoid revisiting nodes that already have been obtained it is possible to provide the known CPID's in the *filter* parameter part of the URL. This speeds up the method considerably.

4.2.5 {FAS-URL}/rest/ResetCache

This method purges the caches, and is necessary to incorporate recent updates that are made on the underlying registries.

5 Reference Manual

None

5.1 Keyboard shortcuts

None

5.2 Command-line commands

None

5.3 Public APIs

See Section 4.2 above.

6 Troubleshooting Common Issues

NA

Annex A. Figures and Tables

A.1. List of Figures

No table of figures entries found.

A.2. List of Tables

No table of figures entries found.

Annex B. Terminology

| ACRONYM | DESCRIPTION |
|----------|---------------------------------|
| CPID | Curation Persistent Identifier |
| ReplInfo | Representation Information |
| REST | Representational State Transfer |