Project no. 033572

CASPAR

Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval

Instrument: Information Society Technologies

Thematic Priority: 2.5.10 Access to and preservation of cultural and scientific resources

PROTOTYPES OF AUTHENTICITY TOOLS AND OF OAIS-BASED INFORMATION BROWSING

<table>
<thead>
<tr>
<th>Document identifier:</th>
<th>CASPAR-2303-RP-0101-1_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submission Date:</td>
<td>04-06-2009</td>
</tr>
<tr>
<td>Due date:</td>
<td>31-05-2009</td>
</tr>
<tr>
<td>Work package:</td>
<td>2300</td>
</tr>
<tr>
<td>Partners:</td>
<td>All Partners</td>
</tr>
<tr>
<td>WP Lead Partner:</td>
<td>STFC/CNR</td>
</tr>
<tr>
<td>Document status</td>
<td>FINAL</td>
</tr>
</tbody>
</table>
Abstract: This document provides information about the prototypes of the Authenticity tools and OAIS-based information browsing using the Finding Aids.
### Delivery Type
Report

### Author(s)
CASPAR Consortium

### Approval
David Giaretta

### Summary

### Keyword List

### Availability
PUBLIC

#### Document Status Sheet

<table>
<thead>
<tr>
<th>Issue</th>
<th>Date</th>
<th>Comment</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>1_0</td>
<td>31 May 2009</td>
<td>Material gathered on collected into one document</td>
<td>Matt Dunckley, Carlo Meghini, David Giaretta</td>
</tr>
<tr>
<td>1_1</td>
<td>4 June 2009</td>
<td>Added Finding Registry functionality</td>
<td>Henri Avancini</td>
</tr>
</tbody>
</table>
Project information

<table>
<thead>
<tr>
<th>Project acronym:</th>
<th>CASPAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project full title:</td>
<td>Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval</td>
</tr>
<tr>
<td>Proposal/Contract no.:</td>
<td>IST-2006-033572</td>
</tr>
</tbody>
</table>

**Project Officer: Carlos Oliveira/Martin Muehleck**

<table>
<thead>
<tr>
<th>Address:</th>
<th>INFSO-E3 Information Society and Media Directorate General Content - Learning and Cultural Heritage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Postal mail:</td>
<td>Bâtiment Jean Monnet (EUFO 1167) Rue Alcide De Gasperi / L-2920 Luxembourg</td>
</tr>
<tr>
<td>Office address:</td>
<td>EUROFORUM Building - EUFO 1167 10, rue Robert Stumper / L-2557 Gasperich / Luxembourg</td>
</tr>
<tr>
<td>Phone:</td>
<td>+352 4301 33052</td>
</tr>
<tr>
<td>Fax:</td>
<td>+352 4301 33190</td>
</tr>
<tr>
<td>Mobile:</td>
<td></td>
</tr>
<tr>
<td>E-mail:</td>
<td><a href="mailto:Martin.MUEHLECK@ec.europa.eu">Martin.MUEHLECK@ec.europa.eu</a></td>
</tr>
</tbody>
</table>

**Project Co-ordinator: David Giaretta**

<table>
<thead>
<tr>
<th>Address:</th>
<th>STFC (formerly CCLRC), Rutherford Appleton Laboratory Chilton, Didcot, Oxon OX11 0QX, UK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone:</td>
<td>+44 1235 446235</td>
</tr>
<tr>
<td>Fax:</td>
<td>+44 1235 446362</td>
</tr>
<tr>
<td>Mobile:</td>
<td>+44 (0) 7770326304</td>
</tr>
<tr>
<td>E-mail:</td>
<td><a href="mailto:david.giaretta@stfc.ac.uk">david.giaretta@stfc.ac.uk</a></td>
</tr>
</tbody>
</table>
CONTENTS

1 INTRODUCTION ............................................................................................................................................... 6

2 AUTHENTICITY TOOL ........................................................................................................................................ 7
   2.1 BRIEF DESCRIPTION OF THE TOOL CONTEXT AND POSITIONING .............................................. 7
       2.1.1 The problem the tool is intended to solve .................................................................................... 7
   2.2 THE NATURE OF THE TOOL ................................................................................................................. 7
   2.3 TOOL REQUIREMENTS .......................................................................................................................... 8
   2.4 WHEN AND BY WHOM THE PRODUCT WOULD BE USED ................................................................. 8
   2.5 HOW THE PRODUCT WOULD BE ACCESSED .................................................................................... 8
   2.6 BENEFITS THE TOOL WILL PROVIDE ................................................................................................. 8

3 AUTHENTICITY CASE STUDY BASED ON IONOSONDE DATA ................................................................. 10

4 OAIS-BASED INFORMATION BROWING ................................................................................................. 12
   4.1 ELEMENTS OF CONCEPTUAL MODEL ............................................................................................... 12
   4.2 FINDING AIDS INTERFACE REFINEMENT .......................................................................................... 14
       4.2.1 Finding Manager ............................................................................................................................ 15
       4.2.1.1 Finding Manager main sequence diagrams ............................................................................. 15
       4.2.2 Finding Registry ............................................................................................................................ 17
       4.2.2.1 Finding Manager management ............................................................................................... 17
       4.2.2.2 DescInfo indexer ....................................................................................................................... 19
1 INTRODUCTION

This document aims to detail the proposed prototype of a proposed Authenticity Tool based on the Authenticity Model developed by Urbino [3] as part of the CASPAR project [1], and provide further details of the Finding Aids for browsing information in Archival Information Packages.

The software will be available at the CASPAR developers’ site (http://developers.casparpreserves.eu:8080).

For the Authenticity tool, the document provides a discussion on why such a tool is a necessary endeavour, information about its functionality and concludes in a real life case study based on how the tool would be used in recording the ingestion and archival of Ionosonde atmospheric data currently archived by the World Data Centre [4] at Rutherford Appleton Laboratory, STFC.

The documentation for the Finding Aid includes functional design details and image capture of user interfaces.
2 AUTHENTICITY TOOL

2.1 BRIEF DESCRIPTION OF THE TOOL CONTEXT AND POSITIONING

There is a recognised need within the preservation community for tools which can facilitate the capture and maintenance of authenticity and provenance information about digital assets over the long term. Work has been done to address this problem of authenticity and build on the OAIS reference model [2] to produce an Authenticity Model. The development of the Authenticity Tool will be based upon this model, facilitating the capture of all relevant Preservation Description Information (PDI) deemed necessary for the a member of the designated community to make an informed judgment as to the trustworthiness of a preserved digital asset.

2.1.1 The problem the tool is intended to solve

The problem is that there is no simple way to state a digital asset is authentic and trustworthy; is it what it is purported to be? How can we verify it is what it says it is or should be? The Authenticity Model has been developed to address this problem. The model suggests a sensible approach to solving this problem is to keep a chain of information about the asset and its lifetime compiled. This record must be kept up to date and associated for the length of the preservation lifetime of the asset. The report of its lifetime allowing one to make an informed judgment as to its authenticity. There are many things that can affect the authenticity throughout the lifetime of digital information, for instance it will have been created at a location by somebody or a process at some point in time, it may have been transferred from one location to another location, it may have been edited, corrupted, encrypted or processed in some other fashion, it may have been migrated into another digital format. These are some generic factors but there are many others which maybe more domain specific which also affect the trustworthiness of the asset and should be captured and recorded, for example a data scientist may wish to know the specification of an instrument that captured the data, or what internal processing algorithm was applied to the data by the instrument hardware itself.

2.2 THE NATURE OF THE TOOL

Based on the CASPAR Authenticity Model, the prototype tool will import a set of Authenticity Protocols (AP), an AP defining a set of procedures that should be undertaken by the capturer. Each AP will be made up of Authenticity Steps (AS). An Authenticity Step will define specific PDI information required for capture and impose questions on the capturer, for instance asking what standards or methodologies have been followed and detail other criteria that must be satisfied. Each step will be executed by an actor, usually a human capturer but some steps maybe automated by
software plug-ins, for example these may read information from file headers or work out fixity information. After each protocol is signed off as finalised, the results will be compiled into an Authenticity Execution Report which will be used to make the informed judgment as to the authenticity of the digital asset. The report will be in a suitable digital format allowing it to be attached to an Archival Information Package (AIP), thus storing it directly with the asset itself. When the AIP and the digital asset it preserves is moved, processed or transformed, the Archival Report can be updated and maintained, keeping the provenance of the digital information relevant.

2.3 TOOL REQUIREMENTS

- The tool will focus on capturing information in textual form, making the capture process as fast and user friendly as possible
- The tool must be customisable, Authenticity Protocols and Steps will be encoded in XML defined by XML schema and imported into the tool, these will capture non generic, project specific information
- Some Authenticity Steps and the information they capture will be generic and therefore standard to the tool
- Some information capture could be automated through the use of plug-ins, for instance an envisioned plug-in may read file format descriptions to pull relevant information from file headers
- An Authenticity Execution Report must be compiled from the results of information capture, this should be exportable in various formats such as RDF or XML for inclusion into an AIP
- The tool must provide a mechanism for local information storage either using a database or file system in order to allow a user to save their current progress and come back to the information capture at a later time.
- The tool may make suggestions as to what information should be captured by an Authenticity Protocol
- The tool must collect provenance information about who is doing the capture, for example name, organisation, time and date

2.4 WHEN AND BY WHOM THE PRODUCT WOULD BE USED

The tool will be used by data producers, administrators and analysts with responsibility for creating and managing data as part of a projects preservation strategy. The results of the tool, the Authenticity Execution Report will be used by those who need to judge the authenticity of the digital information. Project staff with responsibilities from creating a preservation strategy will most likely define the Authenticity Protocols and Authenticity Steps. The tool could be used throughout the life of the digital asset, importing new Protocols and Steps as use of the data changes and evolves. For example in the creation of data by a scientific instrument, the project scientist may define Authenticity Protocols to be followed on initial first level data processing, then when the data is archived for the long term a new set of Protocols may be applied by that archive, allowing the archive administrators to capture information about format migration or encryption.

2.5 HOW THE PRODUCT WOULD BE ACCESSED

The prototype tool will initially be available online, accessed from a web browser, there will be no need for special browser plug-ins. There will be an Authenticity Administrator and an Authenticity Capture sections to the tool, Administrator allowing the setting up of Authenticity project and the importing of Authenticity protocols. The Authenticity Capture section will allow the capture of authenticity information and the application of capture plug-ins.

2.6 BENEFITS THE TOOL WILL PROVIDE

- Basing the tool on the Authenticity Model provides a standard process and terminology
- The main benefit is the capture of all information deemed necessary to make an informed judgement as to the authenticity of a digital asset
– Providing a customisable and flexible mechanism for project preservation staff to define what are the important characteristics that need capturing about their own data
3 AUTHENTICITY CASE STUDY BASED ON IONOSONDE DATA

This case study will briefly discuss how the authenticity tool would be used to capture prominent PDI about the process of ingestion and archival, resulting in the creation of an AuthenticityExecutionReport. The resultant AuthenticityExecutionReport will be a useful guide to a data user wishing to determine the authenticity of these data files.

However when we say a data user in the future will be able to make a judgement as to the authenticity of the digital object, this does not inform us of whether this means that either the files are genuine, they are indeed what they say they are or we can trust the data within them, the data is of sufficient quality to be trusted by the data user.

Therefore in order for the Authenticity Protocol designer to model what PDI needs to be captured and also for a future data user to establish if all the evidence is available, the Authenticity Protocol should include a qualifying statement of intent from the start forming part of the Authenticity Recommendations. For each AP there could also be Authenticity Steps stating clearly and unambiguously criteria that the intended captured evidence must support, only when this is decided can the correct PDI necessary to support this claim be determined.

This case study is based on World Data Centre (WDC) archived data held at Rutherford Appleton Laboratory, STFC and focuses on the long term archival of Ionosonde scientific research data. To design the WDC Authenticity Protocols, it is important to identify the major event types that occur between the data arriving and its final storage. These event types map to the APs.

If the WDC set the following criteria “to record all PDI necessary to verify the authenticity and quality of received data files for long term archival within the WDC” then the appropriate event types can be identified within the archival ingestion system. In this scenario there are three determined event types which can occur between ingestion and archival of the incoming Ionosonde data files. The occurrence of these event types will trigger the execution of the following APs:

- Ingestion of raw data files in varying formats
- Transformation of received data files into IIWG format
- Final validation and archival of IIWG file within WDC

Each AP should have an Authenticity Step stating the criteria against which these steps are measured, for example:

- Ingestion of raw data files in varying formats
  
  In order for this digital data to be accepted as Ionosonde data of sufficient quality the reliability of its source must be verified and recorded by a WDC accredited archivist

  The following must therefore be recorded:

  - Source of dataset
    - Evidence that this is indeed the source
  - Archivist name and details
    - Ideally some form of credentials would be attached

- Transformation of received data files into IIWG format
For the received data file to be deemed as sufficient quality to support data analysis it must have been successfully transformed into standard IIWG data format, the use of processing software must be recorded.

The following must therefore be recorded:

- Details of transformation used
  - Software details including name, version, source
  - Time/date of process
  - System details
  - Details of person responsible
  - Reason for believing this software is reliable

- Details of Transformational Information Properties checked
  - Information Property descriptions
  - Values checked
  - Details of how checked
  - Details of who was responsible for the check

– Final validation and archival of IIWG file within WDC

Successful validation of the IIWG structure and syntax must be achieved and recorded before long term archival can take place

- Details of validation process
  - Details of checks performed
  - Details of person responsible

- Details of transfer to storage
  - Details of person responsible
  - Checks performed on transfer e.g. Fixity checks
  - Details of archive and storage system
  - Date/time of transfer

The Authenticity Protocols can now be completed with Authenticity Steps modelling the capture of PDI supporting these claims. Work to modelling exactly which PDI fields would be required to meet the above claims is underway and cannot be shown in detail within this document.
4 OAIS-BASED INFORMATION BROWSING

4.1 ELEMENTS OF CONCEPTUAL MODEL

The CASPAR Finding Aids is a software package that provides the functionality for the discovery of Archival Information Packages (AIPs) according to the OAIS Reference Model.

In order to achieve its goal, the FA is based on two basic components:

1. Finding Registry, and
2. Finding Manager.

A Finding Registry supports the publication and discovery of Finding Managers, in the same way a UDDI server supports the publication and discovery of Web Services. In CASPAR, the concept of Finding Registry is introduced in order to be independent from any specific technology or de facto standard. Clearly, UDDI is a source of inspiration for Finding Registry and any UDDI implementation is a possible choice for implementing the Finding Registry; yet it is important to maintain these two notions separate and independent from each other.

In a CASPAR installation, there can be several Finding Registries, cooperating with each other in order to carry out their overall goal, possibly in a way that is transparent to applications; that is, an application needs not, and in general will not be aware of the existence of many Finding Registries that cooperate with the one the application interacts with.

Finding Registries follow authentication and authorization mechanisms as the rest of CASPAR components. This way just users with appropriate roles will get access to Finding Aids functionality, e.g. look for AIP ID from querying DescInfo stored objects, store / update DescInfo objects, discovery of Finding components.

A Finding Manager supports the management of Descriptive Information (DescInfo, for short), and is bound to a language for defining and for querying DescInfo. For example, a Finding Manager may talk SQL, another one RDF/S, another one XML.

Every Finding Manager registers with at least a Finding Registry in order to be discovered by applications. The following UML class diagram illustrates the relations amongst the entities introduced so far within a CASPAR installation.

Every CASPAR installation has at least a Finding Registry, while a Finding Registry can only serve a single installation. A Finding Registry registers at least one Finding Manager, while a Finding Manager may be registered with more than one Registry for reliability reasons. Every Finding Manager has exactly one Language, while several managers can talk the same language. Finally, every Manager manages exactly one DescInfo Repository (set of schema elements and associated objects), and vice versa a DescInfo Repository belongs to exactly one Manager.

![Figure 3 - Finding Registries and Finding Managers](image)
From a functional point of view, a Finding Registry supports three main functionalities:

1. Synchronization with the other Finding Registries of the same installation.
2. Registration and Deregistration of a Finding Manager.

A Finding Manager registers by providing a description of itself to the Registry. This description (FMDesc) contains required information, such as:

- (Data definition & query) language spoken by the Finding Manager.
- Handle for invoking the Finding Manager.

Additionally, an FMDesc object can contain information concerning properties of the Finding Manager that applications consider useful for discovery purposes.

From a functional point of view, a Finding Manager supports three main functionalities by using a Data Manipulation Language:

1. Management of DescInfo:
   a. At the schema level: create, delete and browse DescInfo schema elements (i.e., tables or classes or DDTs).
   b. At the object level: create, delete, update and browse DescInfo objects (i.e., tuples or objects or documents).
2. Management of the association between DescInfo objects and AIP identifiers, including usage of these associations for AIP discovery:
   a. Create, delete query and browse association instances (i.e. (AIP-id, DescInfo-id) pairs).
   b. Discovery of AIPs via queries on DescInfo objects.
3. Primitives for the preservation of DescInfo, at all levels. These are: export and import of all of the above: DescInfo schema elements, objects and associations with AIPs.

A user application interacts with the Finding Manager using a particular Query Language (QL) for querying and/or browsing the DescInfo Repository.

Below, Figure 4 sketch relations between above-mentioned Data Definition and Query languages, Descriptive Information schema and object, and Archival Information Packages.

![Figure 4 – Finding Aids conceptual components model](image)

Finally, on Figure 5 the Finding Aids overall interface is presented thought the Finding Manager and Finding Registry components.
4.2 FINDING AIDS INTERFACES REFINEMENT

According to the above sections, Finding Aids interface from D1301 is refined, into Finding Manager and Finding Registry, and by adding methods signatures.

4.2.1 Finding Manager

As we show below into class diagram, the storage of Descriptive Information is also supported due updated FA requirements after last All Hands meetings (Oxford and Crete).
4.2.1.1 Finding Manager main sequence diagrams

Three main sequence diagrams are presented: the first is the standard FM startup sequence diagram, the second is the sequence corresponding to the storage of a concrete DescInfo (schema and object), and the third is to perform a query.

Main actor involved on FM starts up FM owner (see D1301), i.e. institution that taking FM logical view with corresponding framework implementation instantiate a concrete FM by implementing DDL and QL operations.

CASPAR project provides an implementation of the FM that uses RDF Suite, based on that context we describe the steps involved on FM start up (see Figure 6) in following table:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FM creation. A new FM is initialized with a data definition language object and query language object.</td>
</tr>
<tr>
<td>2, 3, 4, 5, 6</td>
<td>Repository initialization. FM initializes its own repository to store its own data structures, namely DescInfo schemas and DescInfo objects.</td>
</tr>
<tr>
<td>7, 8, 9, 10</td>
<td>DescInfo Object initialization. Through these methods a new manager for DescInfo objects and AIPs is created.</td>
</tr>
<tr>
<td>11, 12, 13, 14</td>
<td>DescInfo schema initialization. Through these methods a new manager for DescInfo schemas is created.</td>
</tr>
<tr>
<td>15, 16, 17, 18</td>
<td>FM initialization. Finding manager data structures, e.g. to preserve DescInfo-AIP associations, are created and stored into repository.</td>
</tr>
</tbody>
</table>
Storing a new DescInfo object on a particular FM (see Figure 7) is performed by tree main steps:

- **DescInfo schemaID**: create a new schema for that object or get the schema ID of a previously stored schema.
- **DescInfo object**: create a new descriptive information object based on information provided as well as AIP object used from FM point of view. Associate these objects.
- **Store**: DescInfo object and association with AIP is stored on repository.

See table below:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3, 4</td>
<td>When a new AIP object is created and a DescInfo is provided using the DescInfoSchemaManager interface also a schema that represents that information is created and stored into the repository. If the schema already exists, the calls to get the schema ID replace these calls.</td>
</tr>
<tr>
<td>5, 6, 7, 8</td>
<td>A new DescInfo object is created with provided information and an ID is assigned to it.</td>
</tr>
<tr>
<td>9, 10</td>
<td>A new (FM) AIP object is created with CASPAR AIP ID.</td>
</tr>
<tr>
<td>11, 12, 13</td>
<td>An association between DescInfo object and AIP object is created and all information stored on repository.</td>
</tr>
<tr>
<td>14</td>
<td>Created DescInfo object ID is returned to caller.</td>
</tr>
</tbody>
</table>

Figure 7 - FM start up

Figure 8 - FM store DescInfo schema and object
Finding Manager main functionality is help on looking for AIP objects thought DescInfo querying. In order to do so at least two queries on Repository should be made from a logical point of view. One to get related DescInfo objects and other to get those associated AIP IDs. Lets take a closer look on following steps from sequence diagram (see Figure 8):

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2</td>
<td>A generic user (or application) submits a query by creating it and calling the discoveryAIP method.</td>
</tr>
<tr>
<td>3, 4</td>
<td>The query is appropriately parsed if necessary and passed to the RDF Suite to get the related items.</td>
</tr>
<tr>
<td>5, 6</td>
<td>The result set is parsed and relevant items that can be show to caller (DRM functionality) are used to retrieve all associated items (AIP IDs).</td>
</tr>
<tr>
<td>7, 8</td>
<td>AIP IDs query is performed.</td>
</tr>
<tr>
<td>9, 10</td>
<td>Result set is created and returned to caller.</td>
</tr>
</tbody>
</table>

![Figure 9 - FM search (AIPs)](image)

Finding Manager query language is determined by its own implementation, i.e. RQL for CASPAR implementation of Finding Manager, however if FM is registered to a FR it is also possible to perform full text query by using DescInfo Indexer from Finding Registry (see Section 4.2.2.2) directly from Finding Manager.

Full text queries are performed by calling a FM Web service that uses FR full text query functionality. This internal calls are transparent for users and can be performed by calling FM Web method or interaction with the CASPAR Web Desktop.

### 4.2.2 Finding Registry

As we show below into Finding Registry class diagram (Figure 10), there are two main functionalities areas. First, the Finding Managers management (RegisterFM class); and second, the DescInfo indexing support (DIIndexer class).

Complete the Finding Registry design the extension from CASPARKeyComponent class, as Registry is a CASPAR component. Light blue classes are the provided implementation, and the rest of the classes are the FR foundation classes.

Finding registry provided implementation can be accessed both by a client-application doing Web Service calls or by the GUI ‘CASPAR Web Desktop’. Finding Registry functionality is described below, for each functionality the signature of the methods and the corresponding GUI is showed.

#### 4.2.2.1 Finding Manager management

This functionality area from Finding Registry architecture provides basic functionality to register FMs and to query for registered Managers.
- Save FM, allow to register a new deployed Finding Manager into the Finding Registry. For Web service based Finding Managers following information need to be provided: Data Definition Language (ie. RDF/S), Query Language (ie. RQL), server URI and WSDL. An ID is assigned to new registered Finding Manager.

- Delete FM, is used to cancel a registered FM from the Finding Registry by a given FM ID.

- Browse FMs, this functionality show all registered Finding Managers with the corresponding data, ie. assigned ID, Data Definition Language, Query Language, server host, and WSDL.

- Search FMs, provides the functionality to query the Finding Registry by providing Data Definition Language and/or Query Language. A list of Finding Managers that satisfy query is returned.
Get FM, allows user to get Finding Manager registration information (Data Definition Language, Query Language, server host, and WSDL) from a valid Finding Manager ID.

4.2.2.2 DescInfo indexer

This second functionality area from Finding Registry architecture allows user to directly query the Registry for a particular DescInfo through a simple text query. Moreover this functionality supports Finding Managers on indexing DescInfo to support text queries at FM level.

- Search DescInfo, by performing a full text query the system returns the list of DescInfo that satisfy the query by providing added information also, like Finding Managers that contains that DescInfo objects.
Figure 10 – Finding Registry class diagram

DescInfo Indexer implementation is based on SOLR, and is used by Finding Registry to index all DescInfo objects imported on any registered Finding Manager.

SOLR is a standalone enterprise search server with a web-services like API. It is based on the Lucene Java search library. DescInfo objects are indexed via XML over HTTP. When Finding Registry receives a DescInfo object from a registered Finding Manager, it parses it creating an XML and adding necessary information to the object like FM ID.

After indexing some DescInfo objects query is possible by calling the FR web service method or by interaction with the CASPAR WEB Desktop. Internally the query is forwarded to the SOLR server via HTTP GET and receives the result as an XML. DIIndexer component formats the query in order to have independence between current used indexer (SOLR) and the rest of the FIND component. In this way future change of the Indexer is straightforward because do not affect the rest of the FIND implementation.

DIIndexer supports cursors over formatted result set, so caller can specify the maximum number of elements that get after each call.

This functionality is also used internally by FIND component to support full text search directly from each registered Finding Manager. Calls between sub-components (FM-FR) are transparent to user and automatic filter by FM ID is performed in order to only return DescInfo objects from calling Finding Manager.
References


