Functional Requirements for Phase 1 Technical Development of the Unified Digital Format Registry (UDFR)

UDFR Technical Working Group
April 12, 2010

Table of Contents

Document purpose

In January 2010 a proposal for technical development support for the Unified Digital Format Registry (UDFR) was sent to the Library of Congress. The proposal describes a plan of work to be conducted at the California Digital Library resulting in the public release of the first version of the UDFR registry. This document does not supersede the plan of work described in the proposal, rather it clarifies the functional requirements that were only generally described in the proposal.

General high-level requirements for phase 1

The general requirements for the UDFR registry include:

- Provide public access to the information
- Provide high availability of information
- Maintain unique UDFR identifiers
- Store registry records
- Be capable of expressing the UDFR data model

Use cases for phase 1

The use cases include one or more of the following actors:

- Registry user - a human who is able to search, browse, read and export registry records
- Registry editor - a human who is also able to add and update registry records
- Registry system - the UDFR web interface, software and hardware
- Client system - any software and hardware system that is outside of the Registry system and can send or receive registry queries or record information
    - Examples include repository scripts and format identification tools

Use case - Add new record

Use Case ID	**Add new record**
Description	Add a new record to the registry via a web interface
Actors	Registry editor, Registry system
Assumptions	1. The editor has discovered the registry system's web interface 2. The editor is accessing the registry through a web browser that supports the technology used by the registry's web interface 3. The editor is authorized to execute this use case
Pre-conditions	1. The editor is authenticated and is allowed to execute this feature. 2. The web interface should present all required and optional fields in their proper format. The interface should be self-explanatory and intuitive with hints/helps provided for some fields for additional assistance in data entry. 3. The web interface should populate related data from existing records in the registry. For example, when the editor associates software, say Acrobat Reader, to a format record (ex. PDF 1.6), the web interface should display existing software for users to choose from. There would also be a mechanism to add new related data records to the registry. 4. All controlled vocabularies (ex. Format relationships, etc) should be presented directly on the web interface. The editor should not need to look up additional documents to enter controlled vocabularies. No deep knowledge of the registry data model should be required.
Primary functional path	1. On the registry web interface, the editor fills in all mandatory data attributes, and optionally, non-mandatory data attributes, according to the record requirements. 2. The editor submits the record to the registry. 3. The system validates all data attributes, assigns a unique and persistent identifier for the new record and adds the record to the registry. 4. If the record is associated with other data records, for example a format record may be associated with certain software, the system adds the associations into the registry 5. The system adds administrative information to the registry used in the record's audit trail (invoke *Amend audit trail* use case).

Primary result	The system provides verification of successful record insertion to the registry along with the identifier(s) associated with the record. If an error occurs during attribute validation, the web interface should provide explicit hints for the editor to fix the data entry.
Post-conditions	1. The record is added to the registry with a unique and persistent identifier to the record. 2. Attributes are indexed.
Exceptional path	If a system error has occurred, the web interface should roll back all data insertion to the registry.
Issues	1. User authorization and the management of the registry's persistent identifiers are currently pending the decisions of the UDFR Governance Working Group. 2. This use case currently does not provide a mechanism for detecting duplicate records. For example, if a record for PDF 1.6 already exists in the registry, the web interface would not be able to properly detect the duplication when a user adds a new record. We may want to consider a number of attribute validation rules to attempt to detect the duplication and prompt users on possible duplication during data entry. However, it is ultimately the Registry Editor's responsibility to ensure unique format records in the registry.

Use case - Update record

Use Case ID	**Update record**
Description	Update an existing record in the registry via a web interface
Actors	Registry editor, Registry system
Assumptions	1.The editor has discovered the registry system's web interface 2.The editor is accessing the registry through a web browser that supports the technology used by the registry's web interface 3.The editor has located the most recent version of the record by invoking the *Retrieve record via a web interface* use case. 4.The editor is authorized to execute this use case
Pre-conditions	1.The editor is authenticated and is allowed to execute this feature. 2.The identifier to the record is valid and already exists. 3.The web interface displays all data attributes of the record in the proper format. 4.The system would be able to concurrent update requests. 5.There is a single UDFR registry node where the write would occurs.
Primary functional path	1.On the registry web interface, the editor find the record by searching the registry (invoke *Search records* use case) or by resolving a previously saved identifier. 2.The registry's web interface displays the current version of the record (invoke *Retrieve record via a web interface*. Read-only data attributes such as persistent record identifiers shall be displayed as read-only. No change can be made on those attributes on the web interface. 3.The editor updates the data attributes according to the record schema. 4.The system validates the data attributes and submits those attributes to the registry. 5.If applicable, the system updates the associations of the record with other data in the registry 6.Add a new version of the record to the registry. 7.The system adds administrative information (ex. date/time of this update) about this record and this action to the registry (invoke *Amend audit trail* use case).

Primary result	The system provides verification of successful record update to the registry. If an error occurs during attribute validation, the web interface should provide explicit hints for the editor to fix the data entry.
Post-conditions	1.Record is correctly updated in the registry. 2.Attributes are indexed.
Exceptional path	If a system error occurs, the system would rollback the changes.
Issues	1.There is currently a pending issue on determining the read-only data attributes in the registry.

Use case - Amend audit trail

Use Case ID	**Amend audit trail**
Description	Amend audit trail recording changes to a UDFR format record
Actors	Registry system, Registry editor
Assumptions	1. Registry System has permission to update audit record 2. Audit functionality has been deemed a requirement for the type of UDFR format record 3. Registry System knows the action (update / new / rollback) the Register Editor is performing
Pre-conditions	1. A new or updated UDFR format record has been submitted by the Registry Editor
Primary functional path	1. Invoke the *Update record* or *Add new record* use case 2. Registry System copies the UDFR format record / retrieves cache of original and copies it as part of new audit record 3. Registry System amends timestamp; details of the Registry Editor and the action performed 4. Registry System adds (audit) record to the registry
Primary result	On successful execution the registry system provides no user feedback. The registry system may indicate successful execution to *Update record* / *Add new record*.
Post-conditions	1. Audit record for selected UDFR format record is added to the registry. 2. Attributes are indexed
Exceptional Path	Registry system provides feedback to *Update record* / *Add new record* if adding new audit record fails.
Issues	1. A validation step might be required / preferable. While a diff option might be nice in *Retrieve audit record*, the amount of data stored could be less if we check whether UDFR format records have been edited on submit. 2. The action performed in the Primary Functional Path might be entered manually by a user instead of the Registry System maintaining knowledge of this.

Use case - Search records

Use Case ID	**Search records**
Description	Search records via a web interface or a web service. The result is a formatted set of record abstracts (identifiers of records + a limited set of attributes), not actual records.
Actors	Registry user or Client system, Registry system
Assumptions	1. The Registry User has discovered the registry system's web interface, or, alternatively, the Client System has obtained a URI handle to the Registry System. 2. The Registry User is accessing the Registry System through a web browser that supports the technology used by the Registry System's web interface, or, alternatively, the Client System knows the URI naming pattern used to invoke the registry services. 3. The Registry User (or the Client System) knows the query language that is used for searching the Registry System, and has some familiarity with the data in the collection.
Pre-conditions	The query statement is valid.

Primary functional path	1. Search the Registry for all records that match the query statement. The query is defined by a text string, which is made up of one or more search elements. The following search types are possible:
	    1. Simple search (free text string).
	    2. Advanced search. This should include at least the following search methods:
	        1. search by format name and/or extension
	        2. search by software name and/or vendor
	        3. search by identifier (e.g. UDFR, PRONOM, MIME type).
	1. Format the result set using a default schema, or as requested by the Registry User or the Client System.
	2. Return the formatted result set.

Primary result	A set (possibly empty if the query returned no matches) of formatted abstracts of records. For each record in the set, the abstract contains its known identifiers and some unique and relevant identifying attributes. At the very least, the result should contain the unique identifiers that would be needed to retrieve (display, export) the actual (full) records.

Post-conditions	True
Exceptional path	The Registry System will return an error if the pre-conditions are not met or a system error occurs.
Issues	The output of this use case is identical to the output that is produced by the *Browse records* use case.

Use case - Browse records

Use Case ID	**Browse records**
Description	Browse records via a web interface. The result is a formatted set of record abstracts (identifiers of records + a limited set of attributes), not actual records.
Actors	Registry user, Registry system
Assumptions	1. The Registry User has discovered the Registry System's web interface. 2. The Registry User has discovered the URI that gives access to the 'browse records' function. 3. The Registry User is accessing the Registry System through a web browser that supports the technology used by the Registry System's web interface. 4. The Registry User has some familiarity with the data in the collection. 5. Some of the (sub)categories that are defined to browse the collection may have no actual records associated with them.
Pre-conditions	The URI is correct.
Primary functional path	1. Present the Registry User with a number of pre-defined categories (e.g. file format by application type, software by application type) by which the Registry System can be browsed. Each category may be further divided into subcategories. 2. The Registry User selects one or more categories or subcategories using the designated entry elements in the web interface. 3. Search the Registry for all records that match the selection made by the Registry User. 4. Format the result set using a default schema, or as requested by the Registry User. 5. Return the formatted result set.
Primary result	A set (possibly empty if the selection returned no matches) of formatted abstracts of records. For each record in the set, the abstract contains its known identifiers and some unique and relevant identifying attributes. At the very least, the result should contain the unique identifiers that would be needed to retrieve (display, export) the full actual records.
Post-conditions	True
Exceptional path	The Registry System will return an error if the pre-conditions are not met or a system error occurs.

Issues	The output of this use case is identical to the output that is produced by the *Search records* use case.

Use case - Browse audit trail

Use Case ID	Browse audit trail
Description	Retrieve a summarised audit trail
Actors	Registry system, Registry editor
Assumptions	1. The Registry Editor has discovered the registry system's web interface 2. The Registry Editor is accessing the registry through a web browser that supports the technology used by the registry's web interface 3. The Registry Editor knows the web interface URI for the summarised audit trail 4. The Registry Editor is authorized to execute this use case 5. The UDFR format record type has an audit trail functionality associated with it 6. The URI to access the audit trail is not accessible to the standard Registry User
Pre-conditions	The URI is correct
Primary functional path	1. The Registry Editor requests the audit trail summary from the Registry System 2. The Registry System processes the URI to retrieve the requested data 3. The Registry System returns the audit summary data formatted for display in a web browser
Primary result	On successful execution the Registry System returns the audit summary data formatted for display in a web browser. If no records exist then the Registry System states this.
Post-conditions	The operation is idempotent.
Exceptional Path	The Registry System will return an error if the pre-conditions are not met or a system error occurs.
Issues	

Use case - Retrieve record via a web interface

Use Case ID	**Retrieve record via a web interface**
Description	Retrieve a UDFR record formatted for display in a web interface.
Actors	Registry user, Registry system
Assumptions	1. The Registry User knows the web interface URI for the record of interest 2. The Registry User is authorized to execute this use case
Pre-conditions	The URI is correct.
Primary functional path	1. The Registry User requests the record URI from the Registry System. 2. The Registry System processes the URI to retrieve the requested record. 3. The Registry System returns the requested record information formatted for display in a web browser.
Primary result	The Registry System returns the requested record information formatted for display in a web browser.
Post-conditions	This operation is idempotent.
Exceptional path	The Registry System will return an error if the pre-conditions are not met or a system error occurs.
Issues	

Use case - Retrieve record via a web service

Use Case ID	**Retrieve record via a web service**
Description	Retrieve a UDFR record via a web service. The Client System will request that either: a.) all records be retrieved and returned b.) only records updated after a given date be retrieved and returned
Actors	Client system, Registry system
Assumptions	1. The Client System correctly invokes the Registry System web service Application Programming Interface (API) to obtain the record(s) of interest in the desired format (e.g., XML) 2. The Client System is authorized to execute this use case
Pre-conditions	The Client System request is correct.
Primary functional path	1. The Client System requests the record from the Registry System. 2. The Registry System processes the request to retrieve the record. 3. The Registry System returns the requested record information formatted per the Client System request.
Primary result	The Registry System returns the requested record information formatted per the Client System request.
Post-conditions	(Success) 1. The requested records are returned or 2. A message indicating that no records meeting the request criteria were found (Failure) 1. an error message/code is returned to the Client System 2. the error is logged by the Registry System
Exceptional path	The Registry System will return an error if the pre-conditions are not met or a system error occurs.

Issues	
Notes	Merged with “Export Records via a Web Service” use case, “Unified Digital Format Registry (UDFR) Export Use Cases Version 0.4”, Library and Archives Canada, 2010-02-10.

Use case - Retrieve format identification information

Use Case ID	Retrieve format identification information
Description	The UDFR format registry exports identification information according to a standard schema for format identification tools
Actors	Client system, Registry system
Assumptions	1. Client System always downloads format information, i.e. format information has been modified so the Client System wants to download it 2. Format information is formatted using a single standard schema 3. Multiple Client Systems can execute this use case simultaneously. 4. The Client System does not require authentication. 5. The Registry System understands the format of the request from the Client System
Pre-conditions	1. The client has opened a connection to the Registry System 2. The Registry System is listening for Client System requests
Primary functional path	1. Client System connects to the Registry System and sends a request string for the format information 2. Registry System processes the request string and returns to the Client System a modification status of the format information 3. Client System returns a request string stating it wants to download the new data. 4. Registry System processes the new request string and streams the format information to the Client System
Primary result	The registry system returns a stream of format information data in the format requested by the Client System. The Client System will handle this as it sees fit.
Post-conditions	The operation is idempotent
Exceptional Path	The Registry System will return an error to the Client System if there is a failure in retrieval. If the Client System fails or connection closes from the Client System to the Registry System the Registry System will finish executing this use case gracefully.

Issues	Do we want other standard output formats, with the exception of the DROID identification tool; do other standard identification tools exists that might benefit from UDFR output? It needs to be decided whether the format information stream is created at runtime or is cached through a separate process which creates this information. PRONOM previously delivered this stream from a file created storing the signature information. A technical decision would need to be made about which way works best. Related, PRONOM uses a versioning system which is used by the Client System to decide whether or not to download the new version. This might want to be changed for approach that sees signature file information output at run-time using modified dates as an approach.

Use case - Retrieve audit record

Use Case ID	Retrieve audit record
Description	Retrieve individual audit trail record and show the differences between the current version of the corresponding UDFR record
Actors	Registry system, Registry editor
Assumptions	1. The Registry Editor has discovered the registry system's web interface 2. The Registry Editor is accessing the registry through a web browser that supports the technology used by the registry's web interface 3. The Registry Editor is authorised to execute this use case 4. An audit record exists for the corresponding audit trail and UDFR format record 5. The URI to access the record is not accessible to the standard registry user
Pre-conditions	1. The URI is correct 2. A record exists that is viewable
Primary functional path	1. The Registry Editor requests an audit record from the Registry System 2. The Registry System processes the URI to retrieve the requested data 3. The Registry System also retrieves the corresponding, current UDFR format record 4. The Registry System returns the audit summary data and the UDFR format record highlighting the differences and formatted for display in a web browser
Primary result	On successful execution the Registry System returns the audit record and diff data formatted for display in a web browser
Post-conditions	The operation is idempotent
Exceptional Path	The Registry System will return an error if the pre-conditions are not met or a system error occurs
Issues	The diff functionality may or may not be required so may need removing from this use case

Additional requirements for future phases

Although the following use cases are not UDFR Phase 1 requirements, the UDFR TWG wanted to bring them to the attention of the CDL development team. They are likely to have a large impact on the system architecture and should be taken into consideration when designing the phase 1 architecture.

- suppress legally-encumbered information
- store sample files
- store specs and documentation
- replicate registry information and files

The complete set of UDFR use cases are located at:

- Use case list. “UDFR Technical Working Group - UDFR Wiki”, Excel spreadsheet linked to from web page. <http://www.udfr.org/wiki/images/2/21/Use_cases_jan26_2010.xls>.

The complete set of UDFR functional requirements are located at:

- “All Functional Requirements for UDFR - UDFR Wiki”, Web page. <http://www.udfr.org/wiki/index.php/All_Functional_Requirements_for_UDFR>.