

## Harvest Service Genre: *harvest*

The appropriate attribution for a derivative of this work is: "This document is derived from work created as part of the Federated Repositories for Education (FRED) Project within the Australian ADL Partnership Laboratory. Copyright © University of Southern Queensland and University of Memphis. 2008."

### 1 Introduction

Harvest is the process by which an application collects or gathers (harvests) information about objects that are managed by a data source. A harvest service interface for the data source provides a mechanism for the external agent (harvester) to contact the data source (harvestee) to harvest the data. The harvested data is then used by the harvester to provide other services (e.g., to aggregate the data from multiple data sources). While harvesting is generally focused on gathering metadata from the objects in a collection, any data, as defined by the service interface and data source, may be harvested.

This is a general description of a harvest service genre, independent of application end point, data source, data collected, harvesting protocol or underlying communications protocols and service models. The service genre does not include a mechanism to authorize clients. How to discover service implementations that support harvesting is out of scope.

The words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in [RFC 2119].

The service genre description that follows uses e-Framework service genre description elements as of 2006-12-09, updated to include draft e-Framework service classifications. Other terms, e.g., client, provider, resource, are used as defined in the e-Framework.

Items are tagged and identified using names assigned by the FRED project. Formal e-Framework names will be assigned by the e-Framework.

### 2 Service Genre Definition

#### 2.1 Name

- FRED Service Genre Name: harvest
- e-Framework Service Genre Registry Name: TBD

#### 2.2 Rationale

The harvest service genre is authored to support content delivery from data source to data source as a push mechanism, particularly in the context of repository federations. It

Copyright © University of Southern Queensland and University of Memphis.



This work is licensed under the Creative Commons Attribution-Share Alike 2.5 Australia License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.5/au/>

This work is created as part of the Federated Repositories for Education (FRED) Project within the Australian ADL Partnership Laboratory. The FRED project is sponsored by the Australian Commonwealth Department of Education, Science and Training under the Framework for Open Learning Programme.

specifies the minimum expected behaviours a harvest service is expected to provide, and the common design issues involved in implementing such a service.

### 2.3 Classification

<b>To be provided by the submitter:</b>				
<b>Domain(s)</b>	<input type="checkbox"/> Learning & Teaching	<input type="checkbox"/> Research <input type="checkbox"/> Libraries	<input type="checkbox"/> Administration <input type="checkbox"/> IT Services	<input checked="" type="checkbox"/> Common
<b>Maturity</b>	<input checked="" type="checkbox"/> Immature	<input type="checkbox"/> Mature		
<b>Deployment Scale</b>	<input checked="" type="checkbox"/> Isolated	<input type="checkbox"/> Ubiquitous		
<b>To be determined by the e-Framework:</b>				
<b>Status</b>	<input type="checkbox"/> Approved	<input type="checkbox"/> Placeholder <input type="checkbox"/> Unapproved	<input type="checkbox"/> Superseded <input type="checkbox"/> Withdrawn	
<b>Confidence Level</b>	<input type="checkbox"/> High	<input type="checkbox"/> Medium	<input type="checkbox"/> Low	

### 2.4 Version

- FRED Version: 1.0.1
- e-Framework Service Genre Version: TBD

Version History

Version	Date	Author	Description	Organization/Project
0.1	2007-01-06	DR	Initial version hdl:FREDNA/2BF7810E0B434CE4B7B1124662A98B05	FRED
0.11	2007-03-15	NN	Editorial	FRED
0.12	2007-03-26	DR	Addressed queries	FRED
0.13	2007-03-30	NN	Editorial	FRED
0.14	2007-04-02	DR	Clarifications	FRED
0.15	2007-04-04	NN	Editorial	FRED
0.16	2007-05-23	NN	Finalised	FRED
1.0.0	2007-05-24	DR	Submission to the e-Framework	FRED
1.0.1	2008-06-18	DR	Editorial to align with LSAL ADL work	FRED

## 2.5 Description

The harvest service genre provides the mechanism to gather information from a data source. It is an example of a polling request-response process. (“Polling” is the process of information gathering that is initiated by the client, and recurs periodically.) The data source is assumed to be a managed collection of objects, each with harvestable data. The prime use is gathering information (metadata) about the individual elements or objects within the collection. Harvest is not normally used to gather information about the collection as a whole or the data source itself, unless that information pertains to and is used to manage the harvesting process.

This service genre focuses on gathering information from repositories and other similar data collections of content objects. Typically the data gathered will be metadata about the objects in a repository. Typically this metadata is gathered for federation purposes. For example, harvesting can be used to gather metadata about the objects in the constituent repositories in the federation, to create a federation registry of metadata. The harvest service genre may also be used to gather the objects from the data source.

The service genre is specialized in service expressions to harvest a particular type of information about the objects in the collection, e.g., their metadata, by specifying data formats and communication protocols.

Data sources (collections) expose a “harvest” interface, defined in the service expressions that specialize this service genre. Clients may send requests to this interface to gather information stored or managed by the data source. This information could be metadata describing objects stored on the data source; or it could be the objects themselves. The service end point associated with the data source will respond to the request with a set of information, one (logical) item corresponding to each object managed by the data source. The client may request all information for a specified subset of the objects managed by the data source, or particular information about the objects. The service end point will determine what data (if any) to return in response to the harvest request. The details of the data model used to return results to the client are defined in the service expressions that specialize this service genre.

Note, as with other service genre, there is no expectation that all service implementations will be interoperable. Different service expressions MAY take different approaches to defining interfaces and data models.

As defined, the harvest service genre is not access controlled, i.e., any client may attempt to contact a harvest service end point. There are no authentication controls. The service end point for the data source is responsible for determining what results it will return.

## 2.6 Functionality

The harvest service genre supports two types of functions:

*Harvesting Control Functions.* These functions are used to get control and descriptive information about the data source (as a whole) being harvested. This information enables a client to successfully communicate with the service end point that is providing the interface to the data source. Control information gathered MAY include:

- Description (machine processible) of communications and transport protocols supported.
  - Protocol information SHOULD include version numbers.
  - Protocol and communications information MAY include overall flow and control parameters (e.g., synchronous, asynchronous, flow control, time outs).

- Description of information that can be gathered from or disseminated by the data source for the objects it manages (e.g., metadata formats). Schema and format information SHOULD include version numbers.
- Structural information about the collection of objects managed by the data source, used by the client to request particular information about an object.

*Data Harvesting Functions.* These functions are used to get information about the objects in the data source being harvested. Requests MAY specify:

- What objects to harvest: all or a subset, specified by object identifier, object association (e.g., specifying a metadata object through the identifier of the corresponding content object), or selection condition, e.g., when created, type of object, what part of a collection.
- What information to disseminate: all or a subset, specified by type (e.g., object versus metadata versus identifier) or format of what to disseminate (e.g., attributes, applied dissemination transformation).

Mechanisms MAY exist to provide flow control so that large results sets are returned in chunks.

No other functionality is defined. The functionality that is defined MAY be extended.

## **2.7 Usage Scenarios**

### *2.7.1 Enable–Harvest–Ingest*

The typical workflow for harvesting involves three discrete pieces:

1. A (harvestee) service exposes a harvest *interface* to the data source managed by the harvest service. Developing this interface includes selecting which information about an object in the data source may or may not be requested through the harvest service. Often there is no differentiation made between information which can or cannot be harvested (i.e., the data source is open access).
2. A (harvester) external agent makes a request to collect information through the harvest service from the (harvestee) data source. The external agent “*harvests*” through this service genre: the data source, which is the behind the service end point for the service instance of the service genre “*exposes to harvest*” the available, harvestable information from the data source.
3. The (harvester) external agent adds the harvested information to a distinct, aggregating data source. This is done through an “*ingest*” service, which can be referred to in this context as an “*ingest (harvest)*” service. Because the aggregating data source ingests the harvested data, it is often referred to as “*harvesting*” information from the harvestee data source. However, the harvest service genre described herein details only the delivery of information to a harvester. It makes no assumptions on what the harvester then does with the information.

### *2.7.2 Registry–Harvest Metadata*

A common scenario for harvesting, instantiating the workflow above, is to facilitate discovery of content objects across a set of data providers. This is done by harvesting metadata describing the objects from the various data providers (as opposed to harvesting the content objects themselves), by a single harvesting client. The harvesting client then builds a central data source (often called a registry), containing the harvested metadata. In the workflow outlined above, the harvestees are the data providers; the harvester is the

harvesting client; and the central data source is the aggregating data source, i.e., the endpoint for “ingest (harvest)”.

Discovery of objects is performed against the data central source; not against the harvested repositories. Once the metadata has been harvested and collected in the central source, discovery across the range of repositories involves only a single query on the central data source, searching through the harvested metadata, rather than launching a federated search across all of the data providers.

The metadata providers form a federation of repositories by having their metadata harvested and centralised. The federation can be ad hoc, with metadata harvested as found on open access repositories; or it can be controlled and formal, with policy guarantees underpinning it such as uniform metadata profiles and service-level agreements. The typical scenario is:

1. A constituent (source) repository in the federation exposes an interface to the harvest service.
2. Client identifies, through *harvest control functions* on the source repository, content objects with associated metadata records to be harvested, and metadata formats available for the harvest.
3. Client harvests a metadata instance for each content object from the source repository.
4. Client validates the harvested metadata against the registry schema.
5. Client ingests the harvested metadata into its registry.
6. End users may perform discovery using the harvested metadata on the registry.

The accompanying diagram illustrates this scenario as applied in an identifier-enabled repository:

- Content objects are identified and accessed through identifiers that resolved to the source repository-specific labels (Step 2);
- The harvestee data source provides a metadata record corresponding to the specified content object (Step 3);
- The harvested information is published to end users through the registry (Step 6).

The metadata record in the central data source is assumed to include a retrieval key for the associated content object, which can be used with an obtain request for that content object in the source repository. (See Obtain Resource service genre.) The retrieval key value may be populated before or after harvest, depending on the federation business model.

The metadata in the registry is expected to stay reasonably up to date. As a result, harvesting will usually be a periodically recurrent activity, and restricted only to those metadata records updated on the source repository since the last harvesting session.

A client may not be interested in ingesting all the metadata records available from the provider, but only those relevant to the registry subject matter. The provider needs to offer a mechanism of restricting harvesting to those records identified as relevant.

### 2.7.3 Notify

An alternate use for harvesting is to notify consumers of new objects added to a repository. A periodic harvesting operation gathers descriptions of the objects added to the repository since the last harvesting cycle; these descriptions are exposed through harvested metadata

records. The descriptions are then made available to consumers through some form of notification or syndication.

#### *2.7.4 Harvest Content*

Harvest may be used to harvest other data disseminations besides metadata, e.g., an abstracting service may harvest abstracts or summaries of articles from a set of repositories to build a central collection of abstracts exposed to discovery.

#### *2.7.5 Mirror*

Harvest may be used to harvest data objects from a source directly in support of providing a data mirror or archive. The harvesting client will periodically harvest the source to get changes and updates from the source and reflect these in the mirror or archive.

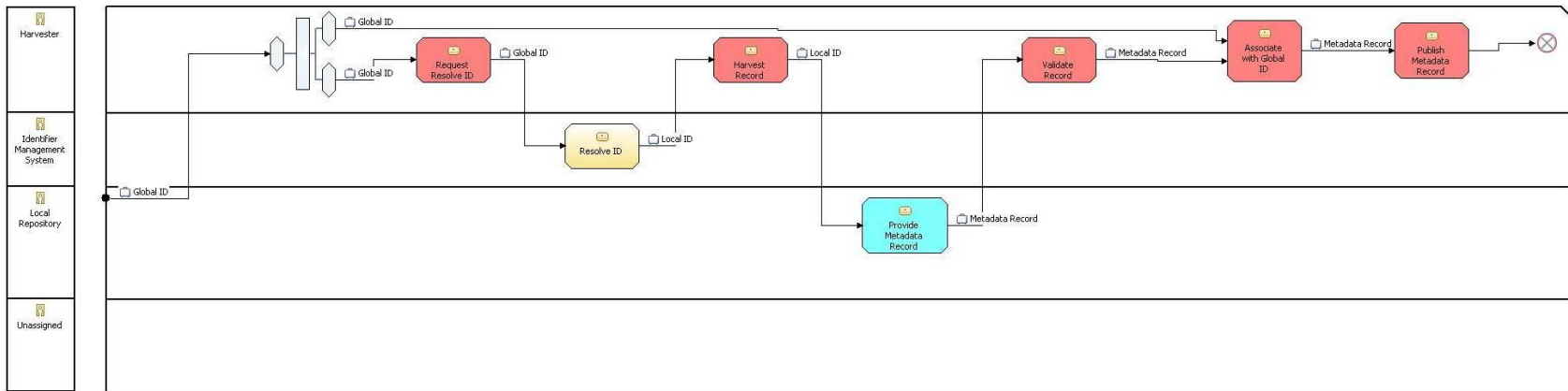


Figure 1: Harvest Metadata Workflow

## 2.8 Applicability

As described in the Usage Scenario element, this service genre describes requests for information from a data source and the delivery of that information in the response. Any subsequent processing of that information, such as its ingestion in a second data source, is out of scope for this service genre.

As defined, the service genre is applicable for harvesting any defined data dissemination for any object within a data source. It is not restricted to harvesting metadata.

Behaviour of the service genre is not defined when the data source requires authentication to permit harvest.

Behaviour of the service genre is not defined when the data source requires authorization or access controls to permit harvest.

Behaviour of the service genre is not defined when the harvest service end point filters results that are returned based on authorization policies or rules that need to be communicated to the data source through the harvest control interface.

Behaviour of the service genre is not defined if communications need to be secure.

## 2.9 Requests & Behaviours

The format and definitions for requests and responses SHALL be defined by the service expressions that specialize the service genre. Requests and behaviours SHALL meet the following conditions:

- At least one defined data harvesting function SHALL be defined.
  - The data harvest function SHALL be capable of harvesting information for all objects managed by the data source exposed to harvest.
    - Restrictions on harvest SHALL be exposed through *Data Harvesting Functions*.
    - Restrictions on harvest SHALL NOT result from extrinsic factors such as size constraints.
  - The request SHOULD include mechanisms to specify or control the format or content of results sets.
  - The request SHOULD include mechanisms to specify the part of the results set to be returned.
  - Responses SHOULD include flow control information.
- At least one defined data harvest control function SHOULD be defined.
  - The response SHALL return basic metadata about the target data source.
  - The response SHOULD include protocol and result format information.
- Responses SHALL include error indicators or other needed control information.

## 2.10 Use & Interactions

The model for a client to interact with a service implementation SHALL be defined by the service expression that specialize the service genre.

## 2.11 Structure

The data source is assumed to be a managed collection of objects. The harvestable data could be the content of the object, metadata describing the object, or both. This data can be expressed in one or more consistent data formats (including the possibility of on-the-fly transformation—"disseminations"). A request to harvest information specifies the objects from which data is to be harvested, and the formats in which the data will be presented. As a precondition to that request, the requestor must ascertain through harvest control functions what the objects to be harvested are, and what the available data formats are.

The diagrams illustrate the flow for *Harvest Control Functions* and *Data Harvest Functions*.

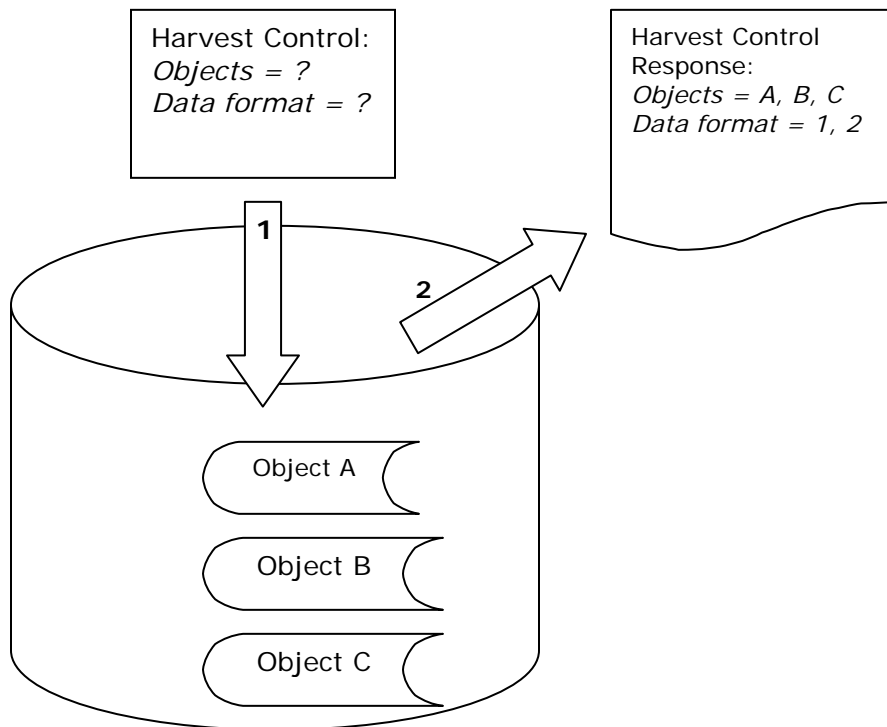


Illustration of Harvest Control Function

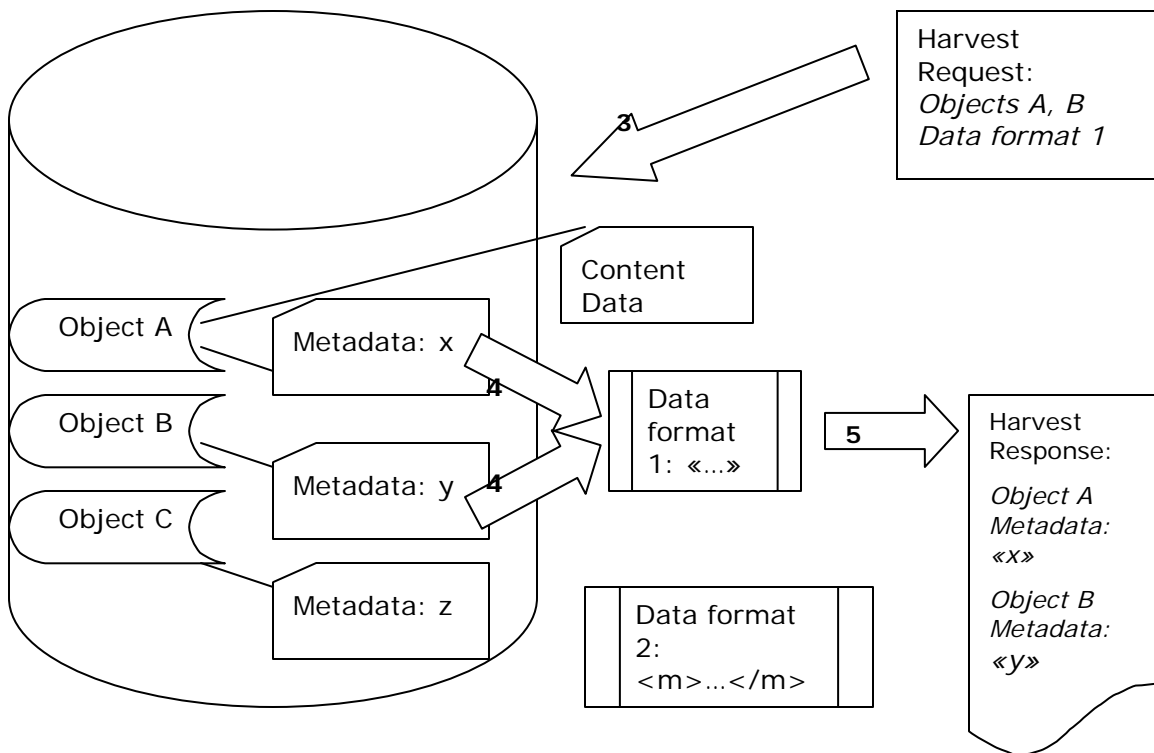


Illustration of Data Harvesting Function

### 2.12 Applicable Standards

None. No standards are directly applicable to the service genre as a whole. The service expressions that specialise the service genre SHALL be defined in terms of standards:

- Service expressions SHALL specify applicable harvesting standards and protocols.
- Service expressions SHALL specify applicable data formats and schemata for harvested data.
- Service expressions SHALL specify applicable communications and transport protocols.

*The Open Archives Initiative Protocol for Metadata Harvesting, Protocol Version 2.0*

<http://www.openarchives.org/OAI/openarchivesprotocol.html> is an example of a protocol that can be used to define a service expression that specialises the service genre.

### 2.13 Design Decisions & Tradeoffs

Client authentication and authorisation is not a part of this service genre.

### 2.14 Implementation Guide & Dependencies

The following design decisions apply to the service expression that specialises the service genre.

Design:

- The service expression MAY include the specification of the communications protocol as part of its definition (e.g., as in OAI-PMH) or it MAY layer the functions on top of

another communications protocol (e.g., using SQL as the communications and control protocol).

- The service expression SHOULD clearly and cleanly separate control functions from harvest functions, e.g., global control functions should not be part of individual harvest request.

Consistency:

- The service implementation SHALL ensure that all objects and all disseminations managed by the data source are harvestable at all times, i.e., timing of updates and transactions on the data source do not impact harvest requests in a way that would omit objects from result sets.

Performance:

- A service implementation SHALL be capable of handling simultaneous requests from different clients.
- A service implementation SHOULD implement an indexing scheme or equivalent method to permit efficient harvesting by identified selection criteria.
- A service implementation SHOULD NOT implement harvest as a search of the objects in the data source if that imposes a severe performance penalty.
- Load balancing SHOULD be implemented for large data sources or those which are harvested frequently (continuously).

Security and Privacy Considerations:

- Service implementations may be subject to denial-of-service attacks.
- Care should be taken to maintain privacy of any personal data or other records that may disclose usage patterns.
- There are no authorization or authentication controls. Care should be taken to maintain data privacy.

### ***2.15 Known Uses***

Actual: None

Potential: The service genre could be used in a service usage model for a repository federation containing a central federated metadata registry. The federated metadata registry would be the client that sends harvest requests to the service implementations providing harvest interfaces to the repositories in the federation. The requests would be used to gather the metadata used to populate the federation registry. The client would periodically harvest the repositories in the federation to obtain updates to the objects held in the repositories. The federated metadata registry could also provide a service implementation interface to allow other clients to harvest the metadata in the federation registry, building a federation of federations. The typical implementation would use an OAI-PMH-based service expression as a specialization of the service genre.

### ***2.16 Service Genre Dependencies***

None.

### ***2.17 Related Service Usage Models***

(FRED) repository federation: [V1.0.0](#).

[\[link to service usage model\]](#)

Harvest (genre) is a part of the repository federation service usage model (genre based) and is used to gather metadata from the repositories and collections that participate in the federation to build the registry data used for discovery. The repository federation service usage model provides an integrated set of service genres used to populate and use the metadata registry that supports a repository federation. Functionality includes content management (creation and management of metadata objects within a repository federation), content discovery (discovery of content objects from a repository federation) and content delivery (retrieval of and access to content objects discovered through a repository federation).

(ADL-R) repository federation: V1.0.0.

[\[ADL-R repository-federation-sum-v100 / hdl:1870/E2FE4AD428A1468FA284E270245F72D7\]](#)

The (ADL-R) repository federation service usage model is a derivative subset of the (FRED) repository federation service usage model. For the purposes of this service genre they are interchangeable.

### **2.18 Related CORE SUMs**

None.

### **2.19 References**

*The Open Archives Initiative Protocol for Metadata Harvesting, Protocol Version 2.0.*