



A Platform for Advanced Data Management

Colectica Repository Technical Information

As of Release r1586



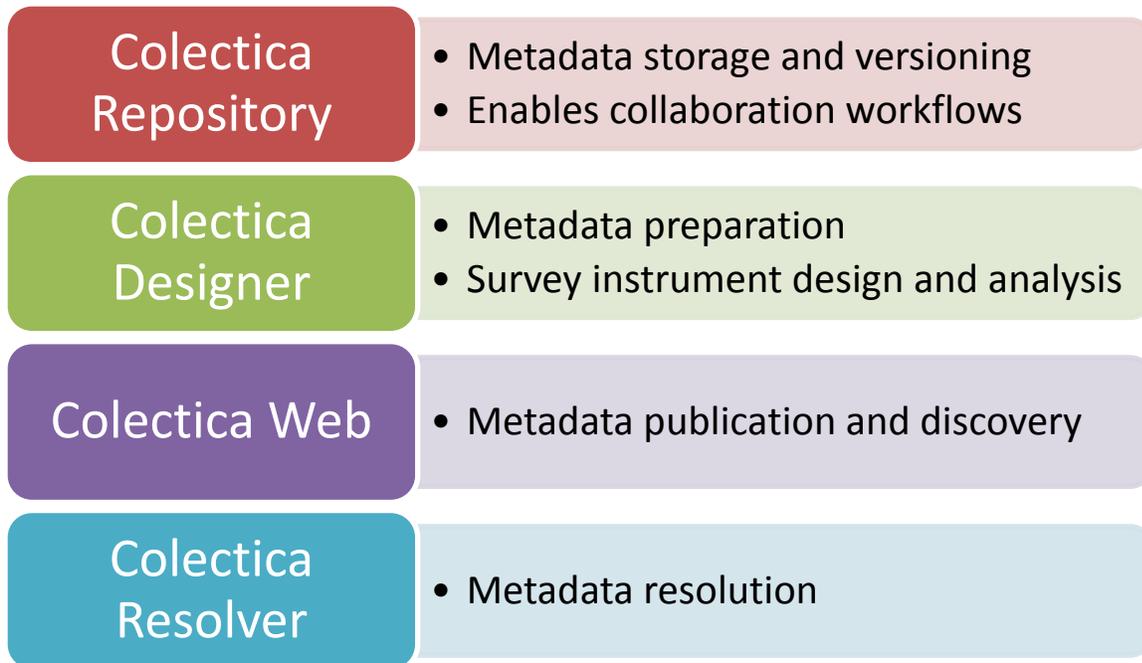
Copyright © 2010 Algenta Technologies. All rights reserved.

COLECTICA PLATFORM OVERVIEW

Colectica is a platform for creating, documenting, managing, distributing, and discovering data. Colectica is fully supported commercial software developed by Algenta Technologies.

Colectica is built on open standards including DDI3, several ISO standards, and RDF.

The Colectica Platform consists of the following components:



Algenta offers several services together with the Colectica Platform, including:

- Maintenance and support
- Custom development
- Training
- Survey and metadata preparation

This document provides a detailed description of Colectica Repository. For more information, please write to contact@algenta.com or call (888) 264-0711.

TABLE OF CONTENTS

COLECTICA PLATFORM OVERVIEW	2
TABLE OF CONTENTS	3
COLECTICA REPOSITORY	4
STORAGE	4
IDENTIFICATION	4
SERIALIZATION.....	5
SECURITY	5
USER AUTHENTICATION	5
ROLE-BASED ACCESS CONTROL.....	5
ENCRYPTION	6
AGENCY MANAGEMENT	6
MULTILINGUAL SUPPORT	6
REPOSITORY WEB SERVICES	6
MANAGING REGISTRATION AUTHORITIES	6
REPOSITORY INFORMATION.....	6
REGISTERING ITEMS	7
RETRIEVING ITEMS	7
VERSIONING	8
DEPRICATING ITEMS.....	8
SEARCH	8
TAGGING	10
REPOSITORY ITEM ANNOTATIONS	10
REPOSITORY ITEM LICENSING	10
DEPLOYMENT OPTIONS	10
ADDITIONAL INFORMATION	11

COLECTICA REPOSITORY

Colectica Repository is a centralized storage system for managing data resources, enabling collaborative workflows, providing automatic version control. Client applications such as Colectica Designer, Colectica Web, or third party applications interact with Colectica Repository using standard SOAP and WS-* communications protocols. Although Colectica Repository is not accessed directly by end-users, it is the backbone of the Colectica platform.

STORAGE

IDENTIFICATION

The Colectica Repository is modeled on the ISO/IEC naming and identification principles outlined in ISO 11179-5 (2005). When an administered item is placed into the repository, it is registered with its international registration data identifier (IRDI). The IRDI is made up of three components which are referred to as the identifier triple throughout the Colectica platform. It consists of three parts:

REGISTRATION AUTHORITY IDENTIFIER (RAI)

When the Colectica repository is used with DDI3, the registration authority identifier comes from an organization's DDI agency id. Currently, the DDI Alliance issues these identifiers upon request. Other RAI issuers such as ARK and DOI are also supported, but use a different resolution system. A single Colectica repository can host multiple Registration Authorities simultaneously.

Implementers Note: The repository can also host read-only copies of items for which it is not the RAI. This allows for a construction of a complete object graph of all related metadata, even if some administered items reference "external" resources.

DATA IDENTIFIER (DI)

The Data Identifier used in the Colectica repository is based on ISO/IEC 9834-8 which describes the creation of globally unique identifiers. Since these identifiers are guaranteed to be unique, it allows users to create their own identifiers without a centralized DI issuance service within the organization. This enables offline use of all Colectica tools and enables federated Colectica repositories within or across Registration Authorities (RAI).

Since many organizations already have non ISO 9834-9 identifiers or use other unique id systems such as ARK, OID, or DOI/Handle, the Colectica Repository can create mappings between unique id systems. When using DDI 3.1 with the repository, these identifier mappings are serialized in the DDI's UserID elements for persistence along with the identification system used.

DDI 3.1 Note: Mappings from ISO 9834-9 identifiers also ensures valid DDI 3.1 generation since DDI restricts the characters allowed in the identifier. Identification systems like ARK and DOI must be mapped to UserIDs to validate with the schema since they make use characters not valid in DDI identifiers.

VERSION IDENTIFIER (VI)

The Colectica repository uses a write once scheme to ensure provenance tracking and data protection. Each change to an administered item therefore increases its revision number within the repository. The revision number corresponds to the ISO Version Identifier. The repository has the capability to retrieve the latest version or version histories of administered items.

Implementers Note: Many organizations may use version numbers in different ways, such as incrementing the version of a survey instrument when a data collection wave is done. The Colectica repository supports this type of change recording through the use of tags. Tags in the Colectica repository work the same way as tags in a version control system. The user can specify a tag, such as 1.1 or Beta 2, and that tag will be associated with the correct revisions of administered items in the repository. In this way, both the write once and an organization's own version control can be accommodated.

SERIALIZATION

Administered Items are stored in the Colectica Repository in an XML serialization. Any format can be accommodated. The repository is currently used to store DDI 3.1 for the Colectica Designer.

When an item is registered with the repository, it may be deserialized and indexed for identification, relationships, and searching. This deserialization is built in for DDI 3.1 items and includes multilingual support. Repository plugins can be made for other standards or item types.

SECURITY

USER AUTHENTICATION

The Colectica Repository supports user profiles. Users must provide username and password credentials in order to access the repository.

Colectica Repository uses the WS-Security Model to implement user authentication. The built in support includes username and password credentials that are verified by a Membership Provider. Custom providers can be built to interface with an organizations current authentication scheme, such as LDAP or active directory. The Repository also supports X.509 client certificates, Windows Authentication, Kerberos Tokens, and SAML Tokens for user authentication, depending on the organizations needs.

Implementers Note: All users must be authenticated with the repository for every call. If an organization wishes to allow guest access, a guest user should be created and given the appropriate roles for access.

ROLE-BASED ACCESS CONTROL

Users may be assigned one or more roles. The roles determine whether the user may search, submit, retrieve, and annotate items in the repository. Each web service method on the repository can specify which roles are needed to complete the invocation.

ENCRYPTION

The Colectica Repository communicates over a Secure Sockets Layer (SSL) connection.

AGENCY MANAGEMENT

The Colectica Repository may be designated as the authoritative source for one or more agencies. An agency is an entity responsible for a particular piece of data or metadata, and corresponds to the registration authority identifier (RAI) of an 11179 international registration data identifier (IRDI). The repository can also host read-only copies of items for which it is not authoritative. This allows for a complete copy of all metadata being located in the repository even if it references external resources.

Colectica DDI Resolver can be used to discover the authoritative repository for a particular agency.

MULTILINGUAL SUPPORT

All metadata stored in the Colectica Repository can be represented in one or more languages. This allows multilingual search and generation of multilingual documentation.

REPOSITORY WEB SERVICES

The Colectica Web Services are built on industry standard SOAP and WS-* messaging protocols to enable easy integration into workflows and third party applications. All web service calls are currently located in the <http://ns.colectica.com/2009/07/> namespace. Web service method names are [highlighted](#).

MANAGING REGISTRATION AUTHORITIES

The repository must be configured with the Registration Authority names for which it is “authoritative”. This is done through two web service calls.

- [CreateRepository](#) – Registers an identifier for a Registration Authority with the repository. This will usually be a DDI agency ID.
- [RemoveRepository](#) – Removes a Registration Authority from a repository. This does not remove any Administered Items; it only changes the repository’s state to Non-Authoritative for that RAI.

REPOSITORY INFORMATION

A user of the Colectica Repository may obtain information about the repository.

- [GetRepositoryInfo](#) – Retrieve a RepositoryInfo object containing repository and user permissions

The repository information includes Dublin Core metadata about the repository, a list of registration authorities (RAI) that the repository is authoritative for, the permissions of the current user, and summary statistics about the types, counts, and revisions of items currently registered.

REGISTERING ITEMS

Administered items can be registered by authorized clients.

- [RegisterItem](#) – Register an administered item with the repository.
- [RegisterItems](#) – Register multiple administered items in a single web service call

Items are submitted by sending the DDI3 serialization of the item along with the item's ISO 11179 IRDI, composed of the RAI, DI, and VI. An object type and commit options are also included to control the repository's registration behavior. It is at this stage that an item may be deserialized from XML for further processing and indexing.

RETRIEVING ITEMS

There are many ways to retrieve administered items stored in the repository.

- [GetItem](#) – takes an ISO 11179 international registration data identifier (IRDI) and returns a `RepositoryItem` object containing information about an Administered Item and its XML serialization.
- [GetLatestItem](#) – takes only the RAI and DI portion of an IRDI, and returns a `RepositoryItem` object containing the latest revision of an administered item.
- [GetItems](#) and [GetLatestItems](#) – These calls take a collection of ISO 11179 IRDIs or RAI and DI pairs, and are for performance improvement. They can be used to substitute successive calls to their single item counterparts.

All four of these web service methods return an object called a `RepositoryItem`. It contains not only the XML serialization of the administered item, but also additional repository metadata about it including a collection of standardized notes, object type, version date, version author, version rationale, published status, deprecated status, and repository authority information.

Implementers Note: The identification information needed for these calls can be obtained from references in other objects, through an appropriate URN or by using the Colectica Repository's search functionality, detailed below.

RETRIEVING ITEM METADATA

When only the administered item's title or description is needed by a client, this series of calls can be used to increase performance.

- [GetRepositoryItemDescription](#) and [GetRepositoryItemDescriptions](#) – These calls take a single or collection of ISO 11179 IRDIs to retrieve a `RepositoryItemMetadata` object. It contains the identity information of the administered item and multilingual fields for `ItemName`, `Label`, `Description`, and `Summary` texts.
- [GetRepositoryItemDescriptionsBySubject](#) and [GetRepositoryItemDescriptionsByObject](#) – These are analogous to the relationship search, taking a `GraphSearchFacet` object as a parameter. They differ in that they return a collection of `RepositoryItemMetadata` objects instead of `RepositoryItems`.

SET MANAGEMENT

Sets, similar to the RDF concept of a named graph, allow for the discovery of working sets. Named sets form the basis of process control in the Repository, allowing states such as development, production, HEAD, etc.

- [GetSet](#) – takes an ISO 11179 international registration data identifier (IRDI) that specifies the root item in the set and returns a list of IRDIs of items present.
- [GetLatestSet](#) – takes an ISO 11179 international registration data identifier (IRDI) that specifies the root item in the set and returns a list of IRDIs of items present. This special case of [GetSet](#) uses the latest version of any referenced administrative item in the object graph, and follows the latest item's references when determining set boundaries.

VERSIONING

The Colectica Repository provides full support for item versioning.

- [GetLatestVersionNumber](#) – find the latest revision (VI) in the repository
- [GetLatestVersionNumbers](#) – find the latest revisions (VI) for a set of administered items. This is for increased performance for multiple calls of [GetLatestVersionNumber](#)
- [GetVersions](#) – retrieve a list of all IRDIs corresponding to the RAI – DI pair of an administered object
- [GetVersionHistory](#) – retrieve all [RepositoryItemMetadata](#) stored by the repository for all revisions of an item based on a RAI – DI pair.

Every revision of an item registered to the repository is saved, allowing clients to retrieve a full version history of any item in the repository. Each revision includes additional information in the [RepositoryItemMetadata](#) object:

- the authenticated user who committed the version
- the date and time the version was committed
- an optional message describing the reason for the version
- additional publication, rights, and deprecation information

DEPRICATING ITEMS

In order to provide complete audit functionality, repository items may not be deleted. If an item is no longer needed it may be deprecated. Search requests can indicate whether they wish to include deprecated items in the results. Items can also be restored, or marked as no longer being deprecated.

- [DepricateItem](#) – marks an item as deprecated based on an ISO 11179 IRDI.
- [RestoreItem](#) – removes the deprecated status of an item based on an IRDI.

SEARCH

The Colectica Repository offers advanced faceted search functionality.

- [Search](#) – The search method takes a collection of facets and search options, returning a search response.
- [GetItemsForScheme](#) – searches for items with the specified parent IRDI, and with the matching facet and search options
- [GetRelationshipBySubject](#) – Request all relationships that the subject IRDI participates in with the given predicate. This includes relations to other IRDIs and is determined based on child objects and references.
- [GetRelationshipByObject](#) – Request all relationships that contain the specified IRDI as the object of a statement based on the given predicate. This can be used to determine parent objects and locations that have references to the specified IRDI.

The [Search](#) method returns a SearchResponse that is outlined below. The [GetItemsForScheme](#) method returns a collection of IRDIs that can be used to retrieve the administered items. The [GetRelationship*](#) methods return a collection of typed IRDIs. The parameters for the Search methods are listed below.

FACETS

A query may consist of the following facets:

- Item type
- Item text
- Location of item text (e.g., choose to find text in an item's name or label)
- Language of item text
- Relationships to other items (using supplemental method calls)

OPTIONS

A search may also specify the following options:

- Maximum number of results to return
- Whether results should be paged
- Whether to rank the results
- Whether to return only the latest version of matching items, or all matching versions
- Whether to search deprecated items
- Whether to perform full-text search by prefix

SEARCH RESPONSE

The Colectica Repository returns results of matching items with the following information:

- Type of object
- Unique identifier (DI)
- Agency identifier (RAI)
- Version number (VI)
- Summary text describing the matching item
- Hostname of the authoritative repository
- Whether the queried repository is authoritative
- Whether the item is deprecated
- A composite identifier (IRDI) consisting of the authoritative agency identifier, the item's unique identifier, and the item's version number

TAGGING

A specific version of any item or set of items may be tagged. A tag is simply a name given to the version so it can be easily referenced in the future. Tagging can be used to mark milestones such as publication.

- [CreateTag](#) – create a tag or tag set for an administered item
- [GetTags](#) – get a listing of tags for an administered item
- [PublishItemGraph](#) – verify all references in an administered item or set of items, and mark them as published.

Implementers Note: To create a tag set, make sure that all references in the item graph are current by calling [PublishItemGraph](#) first.

REPOSITORY ITEM ANNOTATIONS

Users with the appropriate role may annotate items stored in the Colectica Repository in a number of ways.

- [CreateComment](#) – An authenticated user may make a comment on any Administered Item in the repository. This will also include the user's name and date information.
- [GetComments](#) – Get all comments about an Administered Item, and which revision they were applied to.
- [CreateRating](#) – Allows a user to rate an Administered Item for quality
- [GetRatings](#) – Get all ratings about an Administered Item, and which revision they were applied to.

Implementers Note: User annotations are used for social collaboration using Colectica Web.

REPOSITORY ITEM LICENSING

The owner of a repository item may specify a license under which the item is allowed to be used. Before an item is retrieved, the user requesting the item must acknowledge acceptance of the license terms.

DEPLOYMENT OPTIONS

INSTALLED

Colectica Repository is able to be deployed on a Windows server by executing a standard Windows Installer and installing several dependencies.

VIRTUAL APPLIANCE

The Colectica Repository is also available through pre-configured virtual machine images. Using a virtual machine image results in less deployment time since all dependencies are installed and configured. Images are available for VMWare, Virtual PC, and VirtualBox.

CLOUD HOSTED

For organizations that do not wish to dedicate equipment and administrative resources to deploying the Colectica Repository, a dedicated, cloud-based repository is available. A cloud-based repository can be up and running in just minutes, and all ongoing maintenance is handled by Algenta's experienced staff.

ADDITIONAL INFORMATION

For more information, please write to contact@algenta.com or call (888) 264-0711.

<http://www.colectica.com/>

