Next Generation Integration Scorecard
Repository Functional Area

The Repository Functional Area defines operations for finding, managing and composing digital assets.

Asset

An Asset represents a unit of content, whether it be an image, a video, an application document or some text. The Asset defines a core set of definitions applicable to digital content, such as copyright and publisher. Asset content, such as a document, is defined such that there may be multiple formats contained within the same asset. A document may be accessible in both PDF and MS Word, but is the same document, for example. An image may have both a large size and a thumbnail version. Generally, an asset contains more than one version of content when it is left to the application to decide which is most appropriate.

Also known as: *Image, Video, Document, Resource*

❑ Read Operations for *Assets:*   *Score*

❑ Get Assets given unique Ids	
❑ Get all the Assets in a system or Catalog	
❑ Query (Search) Assets based on attribute-based query terms or keywords	
❑ Register for notifications that Assets have been created, updated or deleted	
❑ Get the temporal coverage for the subject of an Asset	
❑ Get Assets covered by a given date-time interval	
❑ Get the locations related to the subject of an Asset	
❑ Get the Assets included within a given location	
❑ Get the Assets mapped to a given Composition	
❑ Get a list of compositions that include a given Asset	

❑ Write Operations for *Assets:*

❑ Create, Update and Delete Assets	

❏ Add an alias Id to reference an existing Asset	
❏ Organize Assets into Repositories	
❏ Add and Remove temporal coverage (start and end date-times) for Assets	
❏ Add and Remove spatial coverage/locations for Assets	
❏ Add and Remove Assets from Compositions	
❏ Sequence/order Assets within a Composition	

Minimally Supported Attributes of *Assets:*

❏ A unique and permanent identifier	
❏ The name of the Asset	
❏ A description of the Asset	
❏ The type of the Asset: photograph, music, transcript, illustration, tax forms	
❏ The provider of this Asset	
❏ Any available branding for the provider of this Asset, for example an organizational logo	
❏ The license/terms of usage of the Asset if available	
❏ The proper title of the Asset	
❏ Boolean test of whether this Asset is in the public domain.	
❏ The copyright statement and of this asset which identifies the current copyright holder	
❏ The copyright registration information for this Asset	
❏ Boolean test if there are any license restrictions on this Asset that restrict the distribution, re-publication or public display of this Asset, commercial or otherwise, without modification, alteration, or inclusion in other works.	
❏ Boolean test if there are any license restrictions on this Asset that restrict the distribution, re-publication or public display of any alterations or modifications to this Asset, commercial or otherwise, for any purpose	
❏ Boolean test if there are any license restrictions on this Asset that restrict the distribution, re-publication or public display of this Asset as an inclusion within other content or composition, commercial or otherwise, for any purpose, including restrictions upon the distribution or license of the resulting composition	
❏ The Source this Asset, the original owner of the copyright of this Asset and may differ from the creator of this Asset	
❏ The list of sources of this Asset in order from the most recent provider to the originating source	

❏ The created date of this Asset, which is generally not related to when the object representing the Asset was created	
❏ Boolean test whether this Asset has been published	
❏ The published date of this Asset. Unpublished Assets have no published date.	
❏ The credits of the principal people involved in the production of this Asset as a display string	
❏ The Composition associated with the Asset, if this Asset is a representation of a Composition of Assets	

Minimally Supported Attributes of *AssetContents:*

● A unique and permanent identifier
● The name of the AssetContent
● A description of the AssetContent
● The type of the AssetContent: jpeg, mp3, ppt, pdf, psd
● The Accessibility Type associated with this AssetContent
● The length of the data represented by this content in bytes, if applicable
● The data of this AssetContent
● A URL associated with the AssetContent, if available
● The Asset associated with the AssetContent

Composition

A Composition is a group of Assets and compositions may be structured into a hierarchy for the purpose of "building" larger content. A content management system may make use of this interface to construct a web page. The Composition hierarchy may map into an XHTML structure and each Asset represents an image or a link in the document. However, the produced web page at a given URL may be represented by another single Asset that whose content has both the URL and the XHTML stream.

Required Programmatic Operations for *Compositions:*   *Score*

❏ Get a Composition given a unique id	
❏ Get a list of Compositions given a list of unique IDs	
❏ Get all the Compositions in a system or Catalog	
❏ Query (Search) Compositions based on attribute-based query terms or keywords	

❏ Create a new Composition	
❏ Update an existing Composition	
❏ Delete a Composition	
❏ Issue notification that a Composition has been created, updated or deleted	
❏ Organize Compositions into Repositories	

Minimally Supported Attributes of *Compositions:*

● A unique and permanent identifier
● The name of the Composition
● A description of the Composition
● The type of this Composition
● The Children of this Composition

Repository

A directory or other kind of organizational construct for managing digital assets. Such an organization serves to separate the management of assets owned by different parties and to scope searches.
Also known as: *Catalog*

❏ Read Operations for *Repositories:*   *Score*

❏ Get Repositories given unique Ids	
❏ Get all the Repositories in a system or Catalog	
❏ Query Repositories based on attribute-based query terms or keywords	
❏ Traverse Hierarchical relationships between Repositories	
❏ Register for notifications that Repositories have been created, updated or deleted	
❏ Register for notifications that hierarchical relationships between Repositories have been created, updated or deleted	

❏ Write Operations for *Repositories:*

❏ Create, Update and Delete Repositories	
❏ Add an alias Id to reference an existing Repository	
❏ Manage Hierarchical relationships between Repositories	

Minimally Supported Attributes of *Repositories:*

❏ A unique and permanent identifier.	
❏ The name of the Repository	
❏ A description of the Repository	
❏ The type of the Repository	
❏ The provider of this Repository	
❏ Any available branding for this Repository, for example an organizational logo	
❏ Any licencing (terms of usage) associated with this Repository	