

# XSEDE Data Management Use Cases

Version 2.0

February 27, 2018

These use cases describe how XSEDE researchers need to manage and use metadata: the data that describes their research data. Use cases DM-01 through DM-04 have not changed significantly in this version but have been reformatted. Use cases DM-05 through DM-09 were extracted from the original DM-05, which provided an overview of metadata needs.

These use cases describe what is needed by XSEDE stakeholders, not what is easily accomplished using the current XSEDE system. This will be used as a basis for further system engineering activities.

<b>DM-01: Share a common repository of data with a distributed user community</b>	<b>2</b>
<b>DM-02: Data management for distributed simulation and analysis</b>	<b>2</b>
<b>DM-03: Shared use of large-scale/streaming sensor input data</b>	<b>2</b>
<b>DM-04: Migration of data associated with change of primary computational site</b>	<b>2</b>
<b>DM-05: Manually create metadata for a data object</b>	<b>2</b>
<b>DM-06: Run a researcher-supplied tool to generate metadata for data objects</b>	<b>2</b>
<b>DM-07: Automatically extract metadata from data objects</b>	<b>3</b>
<b>DM-08: Store metadata for later use</b>	<b>4</b>
<b>DM-09: Search metadata for specific objects of interest</b>	<b>4</b>
<b>DM-10: Add metadata search features to an application</b>	<b>5</b>
<b>References</b>	<b>6</b>
<b>History</b>	<b>6</b>

## DM-01: Share a common repository of data with a distributed user community

See original version. [1]

## DM-02: Data management for distributed simulation and analysis

See original version. [1]

## DM-03: Shared use of large-scale/streaming sensor input data

See original version. [1]

## DM-04: Migration of data associated with change of primary computational site

See original version. [1]

## DM-05: Manually create metadata for a data object

A **researcher** needs to manually create metadata for a data object. The metadata is a set of attributes that describes the object and that can be used to index and/or search for the object when using metadata functions.

In most cases, the **researcher** wants to experience it as follows.

1. First, the researcher accesses the metadata function and selects a specific metadata collection to work with.
2. Then, the researcher provides an identifier for the data object being described.
3. Then, the researcher uses the interface to add or remove attributes to the object's metadata and to change the values of the object's metadata attributes.
4. Finally, the researcher is able to review the new metadata and either store it or save a local copy for later use.

It will always be like this except when the user chooses to store the metadata immediately, in which case the experience described in DM-08 should happen in parallel with this. (The steps may interweave.)

We'll accept any solution as long as in Step 4, the resulting metadata can be used with other metadata functions.

## DM-06: Run a researcher-supplied tool to generate metadata for data objects

A **researcher** needs to generate metadata for one or more data objects using a tool or application selected by (and possibly provided by) the researcher. The metadata is a set of attributes that describes

the objects and that can be used to index and/or search for the objects when using metadata functions. The researcher is responsible for installing the tool or application.

In most cases, the **researcher** wants to experience it as follows.

1. First, the researcher logs into the system where the tool or application is available.
2. Then, the researcher invokes the tool and specifies the data objects to be processed. The tool generates new metadata for the data objects.
3. Finally, the researcher is able to review the new metadata and either store it or save a local copy for later use.

It will always be like this except when the user chooses to store the metadata immediately, in which case the experience described in DM-08 should happen in parallel with this. (The steps may interweave.)

We'll accept any solution as long as in Step 3, the resulting metadata can be used with other metadata functions.

## DM-07: Automatically extract metadata from data objects

A **researcher** needs to automatically extract metadata for one or more data objects. The metadata is a set of attributes that describes the objects and that can be used to index and/or search for the objects when using metadata functions.

In most cases, the **researcher** wants to experience it as follows.

1. First, the researcher logs into a system from which the data objects are accessible.
2. Then, the researcher invokes the metadata function and specifies the data objects to be processed. New metadata is automatically generated for the data objects.
3. Finally, the researcher is able to review the new metadata and either store it or save a local copy for later use.

It will always be like this except when the user chooses to store the metadata immediately, in which case the experience described in DM-08 should happen in parallel with this. (The steps may interweave.)

We'll accept any solution as long as the following are true.

1. In Step 2, the automatically generated metadata should include "ordinary" attributes like the following: filesystem metadata including filename, path, timestamps; database metadata including database and table names, record indices, query or view; the user's identity; the hostname and basic attributed about the host; a timestamp for the metadata extraction.
2. In Step 3, the resulting metadata can be used with other metadata functions.

## DM-08: Store metadata for later use

A **researcher** needs to add metadata to a metadata collection so that it can be used later in search operations. We assume that the metadata has already been generated and is currently accessible to the researcher.

In most cases, the **researcher** wants to experience it as follows.

1. First, the researcher invokes the metadata function and either specifies a metadata collection to work with or creates a new collection.
2. Then, the researcher requests to add new metadata to the collection and references the new metadata. The metadata function performs any necessary validation of the new metadata and then it either returns an error (if there's something wrong with the metadata or the request) or it adds the metadata to the collection.
3. Finally, the researcher should be able to use the interface to confirm that the new metadata has been added.

It will always be like this except when the researcher needs to generate the metadata and add it to the collection at the same time. In that case, the steps described in either DM-05, DM-06, or DM-07 will be interwoven with these.

We'll accept any solution as long as the following are true.

1. The metadata management function should allow for metadata collections to have configurable requirements on the metadata that can be added to them. (E.g, formal schema or simpler attribute rules.)
2. If the metadata management function supports shared metadata collections (collections accessible by multiple people), it must also support and enforce configurable access controls for collections.
3. The metadata management function should also provide a mechanism for researchers to delete collections and manage the existing metadata in collections.

## DM-09: Search metadata for specific objects of interest

A **researcher** needs to search a metadata collection to identify data objects of interest. We assume that the collection has already been populated with metadata about the relevant data objects.

In most cases, the researcher wants to experience it as follows.

1. First, the researcher invokes the metadata function and specifies a metadata collection to work with.
2. Then, the researcher enters a search query. The metadata function returns a set of data object identifiers whose metadata matches the search query.
3. Finally, the researcher is able to save the list of identifiers for subsequent use.

We'll accept any solution as long as in Step 2, the metadata function clearly indicates whether no metadata exists or no metadata was found that matched the parameters given.

## DM-10: Add metadata search features to an application

A **science gateway developer** or **application developer** (hereafter referred to as the **developer**) needs to add metadata search features to an application he or she is developing for use by researchers. We assume that the collection(s) to be searched have already been populated with metadata about the relevant data objects. The developer is, of course, responsible for creating the application.

In most cases, the **developer** wants to experience it as follows.

1. First, the developer visits the XSEDE website, navigates to the developer documentation section, and scans or searches for information about XSEDE's metadata functions. He or she should find documentation for a metadata search API.
2. Then, the developer uses the API documentation to add metadata search features to the application.
3. Finally, the developer makes the application available to researchers and the metadata features in the application work as described.

We'll accept any solution as long as the following are true.

1. In Step 1, the API should support the most common development environments used for science gateways and research applications, like: Python, Java, C/C++. (Ideally, the API would offer a generic API specification allowing developers to generate bindings for different languages.)
2. In Step 1, the API should be reasonably easy for research developers to understand and use.
3. In Step 2, if the developer encounters issues or needs help, he or she can submit a ticket to the XSEDE help desk ([help@xsede.org](mailto:help@xsede.org)) and receive support consistent with XSEDE expectations.
4. In Step 3, if research users of the application encounter issues with the metadata features and report them to the developer and the developer needs help resolving them, the developer can submit a ticket to the XSEDE help desk ([help@xsede.org](mailto:help@xsede.org)) and receive support consistent with XSEDE expectations.

## References

- [1] XSEDE Data Management Use Cases v1.6. A Hossain, C Jordan, March 20, 2014.  
(<http://hdl.handle.net/2142/48909>)
- [2] XSEDE Data Analytics Use Cases v0.3. Strande, et al, June 14, 2013.  
(<http://hdl.handle.net/2142/45702>)

## History

	Version	Date	Changes	Author
Preceding document (See [2].)	0.3	6/14/2013	Final version of the original XSEDE-1 DA-02 use case: "Data Preparation"	Strande, et al (see doc)
Preceding document (See [1].)	1.6	3/20/2014	Final version of the original XSEDE-1 DM-05 use case: "Metadata Storage and Access"	Altaf Hossain, Chris Jordan
Entire document	2.0	2/27/2018	Divided original DM-05 into multiple use cases, each for a specific role/actor or variation	Liming