



Data Store User Guide

Introduction to Reference Links

Unlike most traditional metadata systems, the power of the Data Store comes in relating a reference to other references. This module discusses the different ways to link or relate references, the logic behind the different ways, and when to use each.

Background

Here are some of the benefits of establishing links between references:

- **Minimize Typing** – In many cases, the metadata for one reference applies to another. For instance, a book and its related chapters.
- **Improve Context** – Information resources rarely are stand-alone. In many cases, reports and datasets are related or created in conjunction with a research project.
- **Improve Traceability** – Many datasets, and even some documents, undergo regular updates. You may need to track these changes through time.
- **Easier Discovery** - Consistent management and presentation of your records makes related information easier for users to find.

While there may be many reasons why information should be linked, here are some of most common scenarios and four different links within the Data Store.

- **Part-Of** - One piece of information is part of a larger information set
- **Project/Product** - A project yields one or more products
- **Version** - There is a newer (or older) version of an information resource
- **Cross-Reference** - The information refers to other information resources, similar to “see also.”

“Is-Part-Of” Relationship (Parent/Child)

The “is-part-of” relationship is used when the identity of a reference depends entirely on the attributes of another. In other words, attributes from the parent reference are critical for the description and discovery of the child reference.

For example, a newsletter, newsletter issue, and newsletter article, each of which can be entered in the Data Store as separate references. A reference for a newsletter article is part of the issue; the issue is part of the newsletter series. You can’t adequately describe the article without knowing something about both the issue and newsletter series. The “Is-Part-Of” relationship is the most formal and restrictive link offered by the Data Store and because of this, it is often referred to as the Grandparent/Parent/Child relationship.

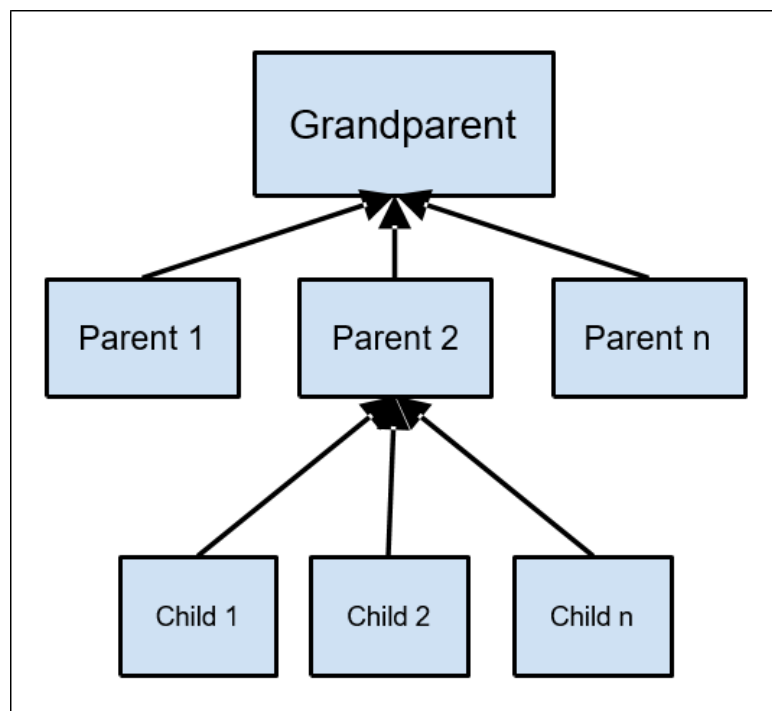


Image 1: Parent/child relationship.

There are a number of 2-level and 3-level hierarchies:

Child Reference Type	Parent Reference Type	Grandparent Reference Type
Aerial Photograph	Aerial Photograph Series	
Map	Map Series	
Newspaper Article*	Newspaper	
Project	Program	
Published Report	Published Report Series	
Satellite Image	Satellite Image Series	
Newsletter Article*	Newsletter Issue**	Newsletter
Book Chapter*	Book	Book Series
Conference Proceeding Paper*	Conference Proceeding	Conference Proceeding Series
Journal Article*	Journal Issue**	Journal

*= Parent is required, **=Grandparent is required

Table 1: Illustrates the levels of relationships possible.

The reasons to use the 'Is-Part-Of' relationship are

- **Prevent duplicate data entry** - If information already exists about the parent, there is no need to re-enter the information for the related children.
- **Cascade update of information** - Corrections to the parent cascade to the children.
- **Establish consistency** - Since all children are pointing to a common parent, there is complete consistency in how the parent information is presented.
- **Enforce organization** - If all children are linked to a report series, it is easy to present the related reports in the context of the report series.

The rules for using this relationship type are very strict:

- Only certain reference types may be part of another (e.g., a book chapter can only be part of a book)
- The child reference may only point to a single parent (e.g., a book chapter may be part of only one book)
- A parent reference may have zero or more children (e.g., a book may have more than one book chapters)

“Project/Product” Relationship

Within the NPS, there are many types of projects, both formal and informal, that create one or more products which are valid to be added to Data Store.

What all projects (formal or informal) share in common, is that their products need to be discoverable individually, but also need to be seen in context with other project products. This is sometimes called ‘bundling.’

With this relationship, you can search for and find an individual map, a report, or a dataset, but you can also see how they all relate to each other because they are outcomes of the same project.

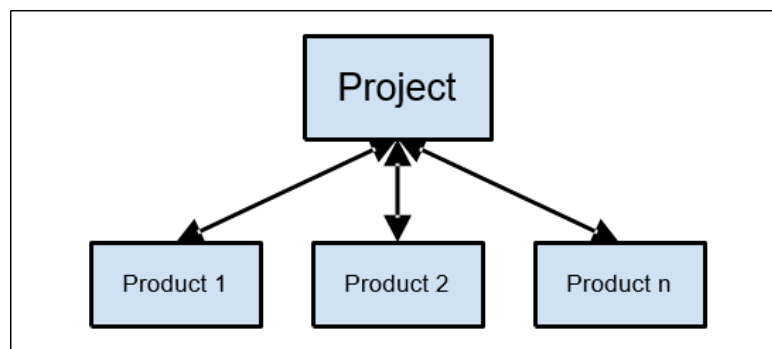


Image 2: Product/Project relationships.

The rules for the project-product relationship type are

- A project reference may create zero or more related products
- A product may be created by only one project (i.e., there is only one originating source for the product)
- Only certain types of information may be created by a project
- A project may not create another project
- The user must own both the project and product references to define the relationship
- The project reference must be active to add related products
- Each related product must be active to be added to the project

Projects can be large, multi-year efforts involving many staff and result in the creation of many products; or, in some cases, can be short-term efforts by single individuals that yield only a single product. Some projects are clearly defined with official project numbers and planning documentation, while others are more spontaneous and informal.

Relating references in the context of a project is powerful. It clearly associates all of the products together at their point of origination, yet still lets each be discovered and used individually.

Bundling references together into a project is a good idea when

- The project clearly has a termination point (i.e., it is not an ongoing operation)
- The information was actually created by the project and not just used to inform the project.

A project reference can also have attached files. Is it better to simply attach files to the project? Or is it better to create separate references for each of the files and link them to the project?

As a general rule, you're better off documenting each product individually with its own reference. The metadata for each product is very different and will have unique descriptions, formats, dates, authors, and content. You'll lose this detail if you simply attach the files to a project. Some guidelines to keep in mind

- Products that can stand alone should stand alone as individual references.
- If an item is a clear and valuable deliverable, then it should probably be described with its own reference.
- If the project has resulted in many products that can't effectively stand alone, consider zipping everything as one zip archive and attach that to the project.

“Cross-Reference” Relationship Type:

Another relationship type is the cross-reference, with is analogous to saying “see also.” Examples include

- Reference A refers to reference B because A used content from B
- If a user is interested in reference A, he or she may also likely be interested in reference B.

Governing rules for this type of relationship are

- Any reference can cross-reference 0 or more other references
- Any reference can be cross-referenced by 0 or more other references
- This relationship is one-way. Reference A can refer to B, but reference B does not automatically link back to A.
- If reference A is cross-referenced to reference B, reference B cannot be deleted until the cross-reference is removed by the owner of reference A.

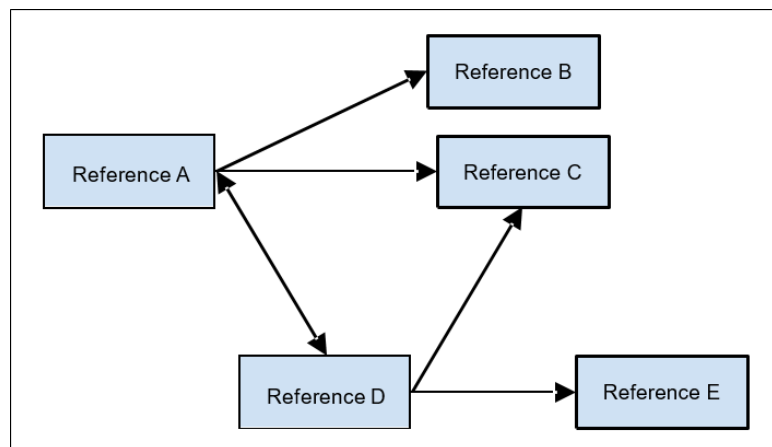


Image 3: Cross-Reference Relationships.

Because the logic for cross-referencing is minimal, this option should be used only when the other relationships do not apply.

Version Relationship Type:

You may have cases where there is a newer version of a dataset or document. In many instances preserving the older version(s) is essential. While most often users should use the most current version, all versions should be discoverable and accessible, especially if they have been used for decisions, analyses, or cited in other reports or publications. Examples include park boundaries, annual status reports, protocols, or standard operating procedures.

References (and therefore the related information resources) should be versioned when:

- There has been enough of a change to the information resource that interpretation/use of its contents may result in different results.
- The metadata used to describe the information resource is different from past versions.

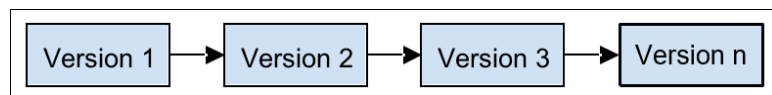


Image 4: Versioning relationship.

Replacing the Holding Location (and not versioning) is appropriate when simple clerical mistakes are fixed that do NOT affect the interpretation of the information resource. For example, a simple spelling change may (although not always) fall into this category.

In all cases, the following rules apply for versioning:

- The user attempting to create a version link owns both references. (i.e., is the owner of the older and newer version)
- Versions are linear and do not branch or merge. An older version may only point to one newer version. Newer versions may only have one older version.
- Only specified Reference Types can be versioned.
- Circular versioning is not allowed. (e.g., $A \rightarrow B \rightarrow C \rightarrow A$)

Summary

Two points to emphasize regarding links

- A Reference can have more than one relationship. Examples include:
 - A published report can be part of a report series and also be generated by a project.
 - An annual report can supersede older versions and also cross-reference other materials
- Use relationships when describing materials created by the NPS. Developing the links ensures that all NPS content is organized consistently and can be grouped and reported on efficiently.