U.S. Fish and Wildlife Service

ServCat Help Document

Introduction to Reference Links

Unlike most traditional metadata systems, the power of ServCat comes in relating a Reference to others. This module discusses the different types of links (aka relationships), their logic and when they should be used.

Introduction

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

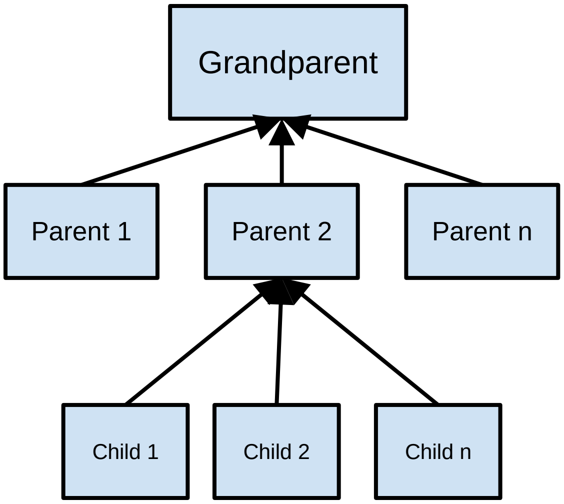
- **Minimize Typing** – In many cases, the metadata for one reference applies to another. For instance, a book and its related chapters.
- **Improve Context** – Information resources rarely are stand-alone. In many cases, reports and datasets are often created in conjunction with a research project.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

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

- **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 version of an information resource
- **Cross-Reference** - The information refers to other information resources

“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. A case-in-point is a newsletter, newsletter issue, and newsletter article, each of which can be described in ServCat as separate references. A reference for a newsletter article” is-part-of” the issue; the issue is part of the newsletter series. It would be impossible to 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 ServCat, because of this, it is often referred to as the Grandparent/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

The purposes for implementing the 'Is-Part-Of' relationship are fourfold:

- **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.

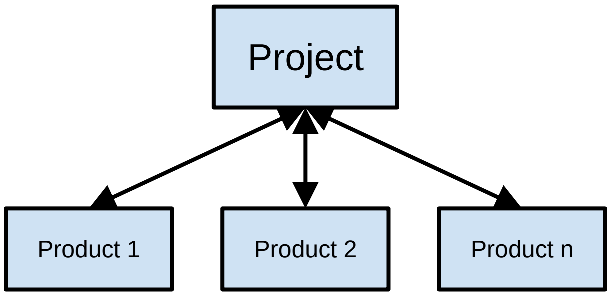
- **Enforce organization** - If all children are linked to a report series, it is easy to present the related reports in the context of the report series.

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 (Bundling):

Within the FWS, there are many types of projects, both formal and informal, that create one or more products which are valid to be added to ServCat. What all projects (formal or informal) share in common, is that the products need to be discoverable on their own yet also travel together since they inform each other. For example, a report will inform and add context to the related dataset and map.



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 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 artifacts or, in some cases, are efforts by single individuals that are completed in a day and 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 them be discovered and used individually.

Bundling References together into a project is encouraged 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.

Because projects can also have attached files, one question is whether it is preferable to add files to the project or create separate linked references that themselves have one or more attached files. In general, it is preferable to document each product individually with its own reference - the metadata for each product should be very different and will have unique descriptions, dates, authors, and content. In summary, there is no definite answer as to how and when to lump or split content. Nevertheless, here are some guidelines:

- Products that can stand alone should stand alone as separate references.
- If the item was something defined as a clear and valuable deliverable, then it should probably be described with its own reference.
- Will the user be overloaded with too many references? If so, better to have fewer.
- It is much easier to start by lumping everything together. The simplest scenario is to zip everything as one zip archive and attach that to the project.

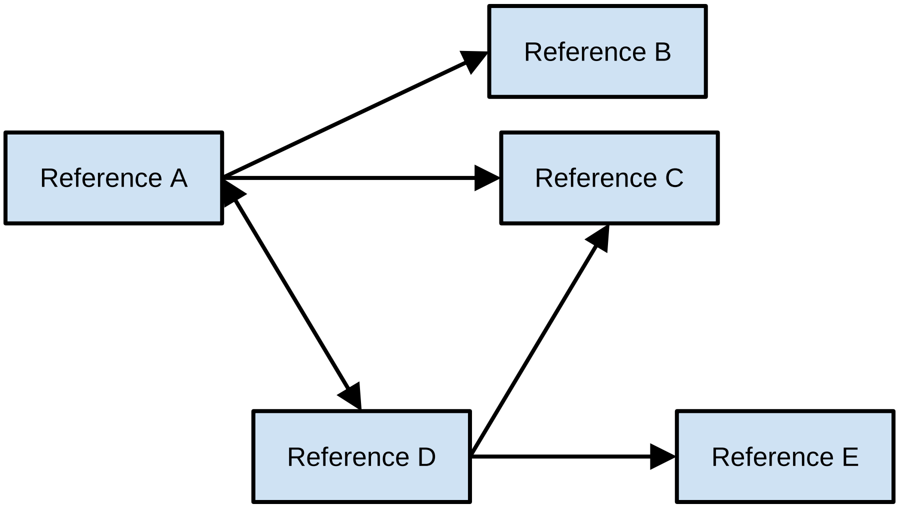
“Cross-Reference” Relationship Type:

There may be other reasons why it is important to link a reference to other that are not addressed by the previous set of relationships. Two of the most common cases are:

- A reference (reference A) refers to another (reference B), because it used content from reference B.
- If individuals are interested in reference A, they will also likely be interested in reference B.

There are few governing rules for this type of relationship:

- Any reference can cross-reference 0 or more other references
- Any reference can be cross-referenced by 0 or more other references



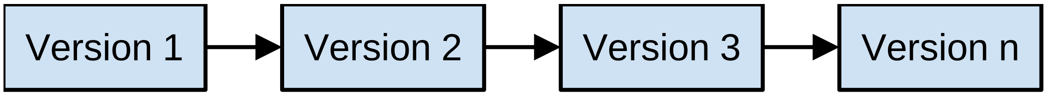
Because the logic for cross-referencing is minimal, this should only be used when the other relationships do not apply.

Version Relationship Type:

There are many cases where there is a newer version of a dataset or document. In many cases, it is still essential to preserve the older version, even if it will not be used as often as the most recent. Thus, it is important that all be discoverable and accessible. Perfect examples include refuge boundaries, annual status reports, protocols and 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.



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, simple spelling changes often (although not always) fall into this category.

In all cases, the following rules apply for versioning:

- The User attempting to create a version link is owns both Reference (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→B→C→A)

Summary

In summary there are two points that are worthy of emphasizing with all of the links defined above. First, it is important to recognize that a Reference can have more than one relationship. Examples include:

- A published report can be part of a report series but also be generated by a project.
- An annual report can supersede older versions and also cross-reference other materials

Second, it is important to use relationship when describing materials created by the FWS. Developing the links ensures that all FWS content is organized consistently and can be grouped and reported on efficiently.