Crossref Functionality Showcase

Software Bibliography Style

December 26, 2025

Abstract

This document demonstrates all features of the crossref implementation for software bibliography entries. It shows how inheritance chains work, how fields are compared, and how concise crossref formatting is applied. Each section includes examples with citations and the resulting bibliography entries.

Contents

1 Introduction **3**
- 1.1 Key Concepts . . . 3
- 1.2 Inheritance Chains . . . 3

2 Basic Software → Version Examples **4**
- 2.1 Base Software Entry . . . 4
- 2.2 Version with Different Date . . . 4
- 2.3 Version with Different Version Number . . . 4
- 2.4 Version with Different Title . . . 4
- 2.5 Version with All Fields Equal (Warning Case) . . . 4

3 Module Inheritance Examples **5**
- 3.1 Base Setup . . . 5
- 3.2 Module Inheriting Version . . . 5
- 3.3 Module with Own Version . . . 5
- 3.4 Module with Different Author . . . 5

4 Fragment Inheritance Examples **6**
- 4.1 Base Setup . . . 6
- 4.2 Fragment from Version . . . 6
- 4.3 Fragment from Module . . . 6

5 Multi-level Inheritance **7**
- 5.1 3-Level Chain: Software → Version → Module . . . 7
- 5.2 4-Level Chain: Software → Version → Module → Fragment . . . 7
- 5.3 Deep Chain with Module Version . . . 7
- 5.4 4-Level Chain: CGAL Voronoi Module → Fragment . . . 8

6 Edge Cases and Special Scenarios **9**
6.1 Editor Field Comparison 9
6.2 Date Precision Differences 9
6.3 Multiple Authors 9

7 Complete Bibliography **10**

8 Summary **11**

1 Introduction

The crossref functionality allows software bibliography entries to reference parent entries, creating inheritance chains where child entries inherit fields from their parents. This enables concise bibliography formatting by showing only the differences between child and parent entries.

1.1 Key Concepts

- **Inheritance**: Child entries automatically inherit fields from parent entries via the `crossref` field
- **Field Comparison**: The system compares child fields with parent fields to determine what to display
- **Concise Format**: Only differing fields are shown, with a citation reference to the parent
- **Version/SWHID Requirement**: For Software → Version relationships to make sense, each `@softwareversion` entry MUST have either:
  - An explicit `version` field, OR
  - A `swhid` id

  Entries without either will not use the "version X of [parent]" format.
- **Relationship Types**: Different entry types use different relationship text:
  - Versions: "version X of [citation]"
  - Modules: "part of [citation]"
  - Fragments: "from [citation]"
- **Duplicate Detection**: When all key fields are equal, a warning is issued and the full entry is shown

1.2 Inheritance Chains

The following inheritance chains are supported:

- **2-level**: Software → Version
- **3-level**: Software → Version → Module
- **3-level**: Software → Version → Fragment
- **4-level**: Software → Version → Module → Fragment

2 Basic Software → Version Examples

This section demonstrates basic crossref scenarios between software and softwareversion entries, using the real-world Scilab example.

2.1 Base Software Entry

Scilab is a software for numerical computation. Here is the base software entry [22].

2.2 Version with Different Date

When only the date differs (but the version field is present), the output shows the version, date, and citation. The version field is required for proper Software → Version inheritance.

Example: [23]

The output format is: *Version 1.1, Jan. 1994 of [parent citation].* The version entry shows the version number and date difference (1994 vs 1994-01) while inheriting author, title, and institution. It also includes SWHID and file information.

2.3 Version with Different Version Number

When only the version number differs, the output shows the version and date with a reference to the parent. Note that the `@softwareversion` entry must have an explicit `version` field or `swhid` id for this format to be used.

Example: [21]

The output format is: *Version 2.0, 1995 of [parent citation].*

2.4 Version with Different Title

When the title differs, it is shown along with the citation.

Example: [24]

The output shows the overridden title "Scilab Extended" along with the version and citation reference.

2.5 Version with All Fields Equal (Warning Case)

When a version entry has all fields identical to its parent, the system emits a warning and shows the full entry. This indicates a potential duplicate that should be reviewed.

Example: [14]

Note: Check the compilation log for a warning message about duplicate entries. This synthetic example is kept for pedagogical demonstration of duplicate detection.

3 Module Inheritance Examples

Modules are components of software that can have their own versions or inherit versions from their parent. This section uses the real-world CGAL (Computational Geometry Algorithms Library) example throughout.

3.1 Base Setup

We start with the CGAL base software entry [20] and its version [18].

3.2 Module Inheriting Version

When a module does not have an explicit version field, it inherits the version from its parent but does not display it. Only the author, subtitle, and "part of" relationship are shown.

Example: [11]

The output format is: *M. Karavelas, "2D Voronoi Diagram Adaptor", part of [parent citation]* (no version shown, inherits from parent). The module also shows its license and URL, properly referencing the parent version.

3.3 Module with Own Version

When a module has its own explicit version that differs from the parent, both the version and the "part of" relationship are shown, along with the author.

Example: [13]

The output format is: *S. Pion, "3D Convex Hull", version 4.0 part of [parent citation].*

3.4 Module with Different Author

Modules can have different authors while still referencing the parent.

Example: [10]

The output shows the different author (A. Fabri) along with the subtitle and "part of" relationship.

4 Fragment Inheritance Examples

Code fragments represent specific code sections extracted from larger software components. This section uses the real-world Parmap library example throughout.

4.1 Base Setup

We start with the Parmap base software entry [19] and its version [8].

4.2 Fragment from Version

A fragment can reference a version directly. The output shows the subtitle, date, and a reference to the parent version.

Example: [9]

The output format is: *"Core mapping routine", 2020 from [parent citation]* (includes SWHID information with path and line numbers).

4.3 Fragment from Module

Fragments can also reference modules, creating a 4-level chain: Software → Version → Module → Fragment.

Example: [6]

This demonstrates a complete 4-level inheritance chain where the fragment references a module, which in turn references a version, which references the base software entry.

5 Multi-level Inheritance

This section demonstrates deep inheritance chains with multiple levels, using real-world examples.

5.1 3-Level Chain: Software → Version → Module

A complete 3-level chain showing how modules inherit from versions. This example uses the real-world CGAL library.

- Base software: [20]
- Version: [18]
- Module: [11]

The module properly references its parent version in the bibliography, showing only the fields that differ from the parent (author, subtitle, license, URL).

5.2 4-Level Chain: Software → Version → Module → Fragment

The deepest supported chain demonstrates complete inheritance. This example uses the real-world Parmap library.

- Base software: [19]
- Version: [8]
- Module: [7]
- Fragment: [6]

Each level correctly references its parent, showing the complete inheritance chain. The fragment shows its subtitle and SWHID information, referencing the module which references the version which references the base software.

5.3 Deep Chain with Module Version

A module can have its own version in a deep chain. This example uses CGAL with a module variant that has an explicit version.

- Base: [20]
- Version: [18]
- Module with version: [13]
- Fragment: (fragment from module with version can be created if needed)

The module with its own version (4.0) is shown along with the "part of" relationship to the parent version.

5.4 4-Level Chain: CGAL Voronoi Module → Fragment

A complete 4-level chain using the CGAL Voronoi Diagram Adaptor module, showing a fragment extracted from the module.

- Base software: [20]
- Version: [18]
- Module: [11]
- Fragment: [12]

This demonstrates a fragment ("Print Endpoint") extracted from the CGAL Voronoi Diagram Adaptor module, creating a complete 4-level inheritance chain. The fragment shows its subtitle and SWHID information with path and line numbers, referencing the module.

6 Edge Cases and Special Scenarios

This section demonstrates how the system handles special cases. These examples use synthetic entries specifically designed to test boundary conditions that may not exist in real-world data.

6.1 Editor Field Comparison

Entries with editor fields are compared correctly.

- Base with editor: [1]
- Version with same editor: [2]
- Version with different editor: [3]

6.2 Date Precision Differences

The system handles different date precisions (year, year-month, full date).

- Base (year only): [15]
- Version (year-month): [16]
- Version (full date): [17]

6.3 Multiple Authors

Entries with multiple authors are handled correctly.

- Base with 3 authors: [4]
- Version with 4 authors: [5]

7 Complete Bibliography

The following bibliography shows all entries with their crossref relationships and concise formatting.

Complete Showcase Bibliography

[1] [Software] M. Author, *Edited Software* (Coordinated by C. Editor), 2021. Publishing House.

[2] [Software Release] Version 1.0, 2021 of [1].

[3] [Software Release] version 1.1 of [1], (Coordinated by D. Editor), 2021.

[4] [Software] A. First, A. Second, and A. Third, *Collaborative Software* 2021. Collaboration Inc.

[5] [Software Release] A. First, A. Second, A. Third, and A. Fourth, version 2.0 of [4], 2021.

[6] [Software excerpt] "Work distribution algorithm", 2020 from [7], SWHID: ⟨swh:1:cnt:example9876543210fedcba9876543210fedcba98765432⟩.

[7] [Software Module] R. Di Cosmo, "Parallel mapping core", part of [8].

[8] [Software Release] Version 1.1.1, 2020 of [19], SWHID: ⟨swh:1:rel:373e2604d96de4ab1d505190b654c5c4045db773⟩.

[9] [Software excerpt] "Core mapping routine", 2020 from [8], SWHID: ⟨swh:1:cnt:43a6b232768017b03da934ba22d9cc3f2726a6c5⟩.

[10] [Software Module] A. Fabri, "2D Arrangements", part of [18]. URL: https://doc.cgal.org/5.0.2/Manual/packages.html#PkgArrangement2.

[11] [Software Module] M. Karavelas, "2D Voronoi Diagram Adaptor", part of [18]. URL: https://doc.cgal.org/5.0.2/Manual/packages.html#PkgVoronoiDiagram2.

[12] [Software excerpt] "Print Endpoint", 2020 from [11], SWHID: ⟨swh:1:cnt:8451c216bded7e6769381cb49c5ebc1c7ca3f274⟩.

[13] [Software Module] S. Pion, "3D Convex Hull", version 4.0 part of [18]. URL: https://doc.cgal.org/5.0.2/Manual/packages.html#PkgConvexHull3.

[14] [Software Release] J. Smith and J. Doe, *Example Software* version 1.0, 2020. Example University. URL: https://example.com/software.

[15] [Software] D. Tester, *Date Test Software* 2020. Test Lab.

[16] [Software Release] Version 1.0.1, June 2020 of [15].

[17] [Software Release] Version 1.0.2, June 15, 2020 of [15].

[18] [Software Release] Version 5.0.2, 2020 of [20], SWHID: ⟨swh:1:rel:636541bbf6c77863908eae744610a3d91fa58855⟩. URL: https://docs.cgal.org/5.02.

[19] [Software] R. Di Cosmo and M. Danelutto, *The Parmap library* 2012. Inria, University of Paris, and University of Pisa.

[20] [Software] The CGAL Project, *The Computational Geometry Algorithms Library* (Coordinated by CGAL Editorial Board), 1996. URL: https://cgal.org/.

[21] [Software Release] Version 2.0, 1995 of [22], SWHID: ⟨swh:1:rel:example1234567890abcdef1234567890abcdef12345678⟩.

[22] [Software] F. Delebecque, C. Gomez, M. Goursat, R. Nikoukhah, S. Steer, and J.-P. Chancelier, *Scilab* 1994. Inria.

[23] [Software Release] Version 1.1, Jan. 1994 of [22], SWHID: ⟨swh:1:dir:1ba0b67b5d0c8f10961d878d91ae9d6e499d746a⟩.

[24] [Software Release] *Scilab Extended* version 1.5 of [22], June 1994.

8 Summary

This showcase demonstrates:

- Basic crossref functionality between software and versions (using Scilab)
- Module inheritance with and without explicit versions (using CGAL)
- Fragment inheritance from versions and modules (using Parmap)
- Multi-level inheritance chains (3 and 4 levels) using real-world examples
- Edge cases including editor fields, date precision, and multiple authors
- Duplicate entry detection and warnings

All examples show how the crossref system creates concise bibliography entries by showing only the differences between child and parent entries, making bibliographies more readable and avoiding redundant information.