Cradle-7

From concept to creation...

3SL

Reusing Information with Adaptations in Cradle

RA004/07   June 2019

http://www.threesl.com
salesdetails@threesl.com
support@threesl.com

Contents

List of Tables

Introduction

Most organisations have **products**, created and evolved in **projects**. Product information is typically managed as a **library** of standard items, often baselined. Each project adds new information and changes existing information. In general, changes do not affect library items or the links between them so other uses of the library do not become invalid. This is especially common in agile projects.

Subject

This paper describes how Cradle's **adaptation** mechanism implements a library and projects that reference that library in **one database**.

It considers two uses of adaptations that will meet the needs of any project:

1. Adaptations of **1** information type. This occurs when a library and project use an item type. The project items are specific to the project, or library references, or are adaptations of the library items.

2. Adaptations of **2** information types. This occurs when project items are linked to library information of a different type, and may then create adaptations of that library information for its own use.

The paper assumes a familiarity with Cradle.

Adaptations

Adaptations are copies of items that inherit links to the items' related information. The adaptation is linked to the original item and the inherited items by user-defined types of cross reference. Adaptations are highlighted by user-defined marks in their names. Adaptations and references are highlighted in the UI and reports in the colour defined for the adaptation and reference link types.

An adaptation creates a project's copy of an item in a library, so the project can change the item for its own purposes.
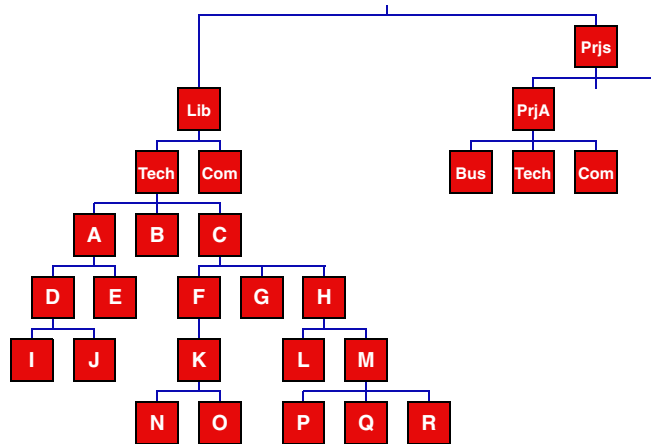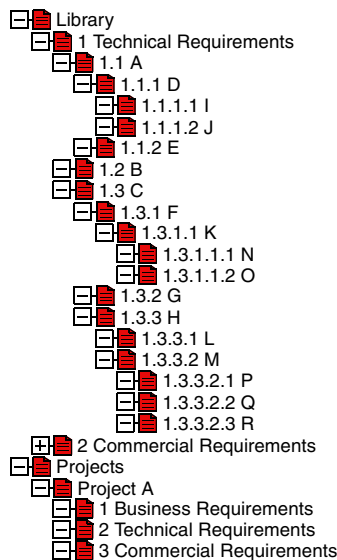
Adapting One Type of Information 1

This is the simplest situation. A hierarchy of items exists in the library and the project will have its own hierarchy of the same type of item. The project will:

• Create its own items
• Reference items from the library
• Create its own adaptations of library items

The important characteristics of this case is that all items are of the same type.

Typical Data 1.1

A library has requirements as **REQ** items. A project has requirements, also as **REQ** items. The project may reference library requirements or adapt them. All other project information will link to requirements *in the project's own hierarchy*. A typical set of requirements could be:

Library
  1 Technical Requirements
    1.1 A
      1.1.1 D
        1.1.1.1 I
        1.1.1.2 J
      1.1.2 E
    1.2 B
    1.3 C
      1.3.1 F
        1.3.1.1 K
          1.3.1.1.1 N
          1.3.1.1.2 O
      1.3.2 G
      1.3.3 H
        1.3.3.1 L
        1.3.3.2 M
          1.3.3.2.1 P
          1.3.3.2.2 Q
          1.3.3.2.3 R
  2 Commercial Requirements
Projects
  Project A
    1 Business Requirements
    2 Technical Requirements
    3 Commercial Requirements

## Schema Setup 1.2

Define three link types with names such as:

• **HAS CHILD**, for the parent-child links in the library's and projects' hierarchies
• **REFERENCES**, for references from projects into the library
• **IS ADAPTATION OF**, for the adaptation links

and set colours for the references and adaptations link types that will be used to colour items in the UI.

Enable the hierarchy options, but do not allow any reordering:

• To become top-level items
• Outside an item's own hierarchy

Enable adaptations for the item type **REQ** and specify the adaptations and references link types.
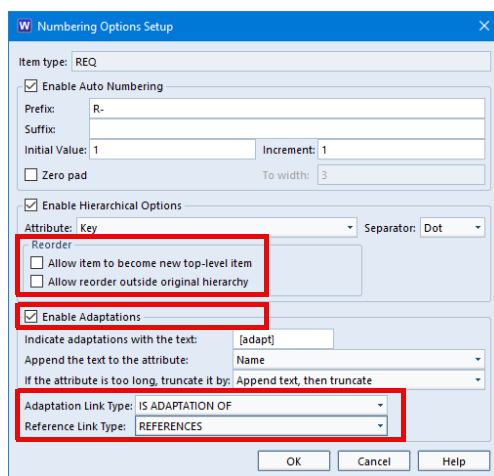
Define the link rules for these three link types for the item type (**REQ** in this case) to be:

Table 1: Link Rules to Adapt 1 Item Type			
From Item	**To Item**	**Link Type**	**Default**
REQ	REQ	HAS CHILD	Y
REQ	REQ or \<any\>	REFERENCES	
REQ	REQ	IS ADAPTATION OF	

Cradle will create these rules automatically for you.

Creating New Items 1.3

The project creates new items by selecting existing **REQ** items and using the Cradle operations:

- **New** → **Child**, to create a child of an existing **REQ** item
- **New** → **Sibling**, to create a sibling of a **REQ** item
- **New** → **Hierarchy**, to create a hierarchy of **REQ** items

All new items will be linked to the existing items by **HAS CHILD** links, as this is the default link type for cross references between **REQ** and **REQ** items.

Use a Library Item 1.4

To use a library requirement inside the project's hierarchy:

1. Open the library hierarchy and find the item to be referenced
2. Open the project hierarchy and find where to place the referenced item
3. Drag & drop the library item onto the project item and choose **Create Reference...**
4. Click **OK** in the next dialog
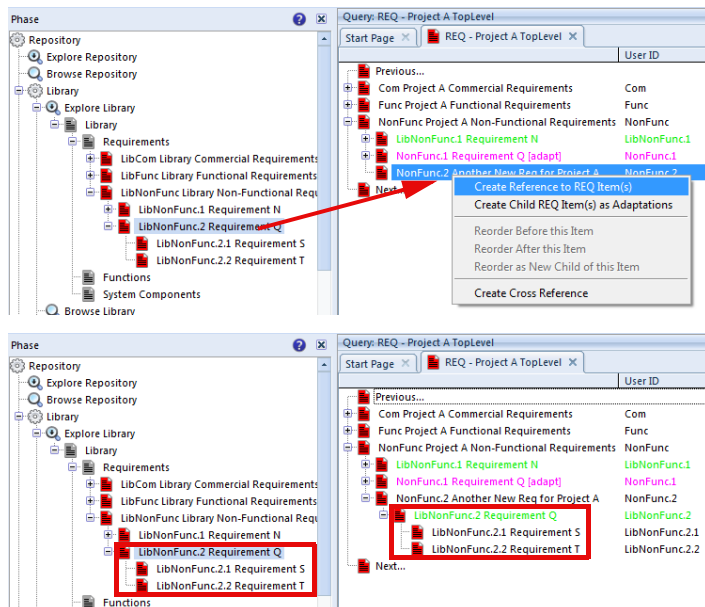
A **REFERENCES** cross reference is created from the project item to the library item. The library item is shown in green to emphasise that it is a reference.

The expansion of the requirement in the library is now part of the project's requirement hierarchy, so in the example, requirements **S** and **T** can be seen in the project's own requirement hierarchy.

In most cases, the referenced requirement and its expansion would be read-only to the user who has referenced them.

Adapt a Library Item 1.5

If the project wants to change an item referenced from the library, to:

- Change the requirement itself
- Change the items that are linked to the requirement
- Change any of the linked requirements

it cannot change the library item because it may be used elsewhere, and it should be read-only to the project. So the project must create an adaptation of the library requirement, by selecting it and:
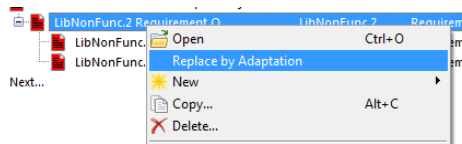
- Right-click and **Replace by Adaptation**, or
- In the **Item** tab, select **Replace by Adaptation**

A dialog will be shown that summarises what links will be created. Select **OK** in this dialog and:

- The requirement will be replaced by an adaptation, a copy of the requirement that is owned by the user
- The item is named and coloured as an adaptation
- The item is linked in the project requirement hierarchy
- The item is linked by references to the same items to which the library requirement is linked
- The adaptation is linked to the library item by an **IS ADAPTATION OF** link
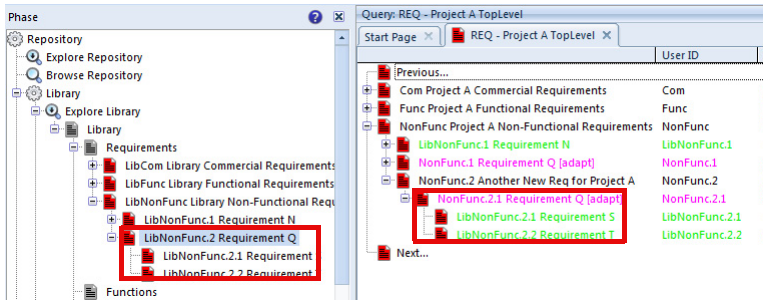
The library requirement **Q** is replaced by an adaptation of **Q** that the project can modify as needed, and the expansion of the library requirement **Q** is now the expansion of the project's own requirement **Q**. So requirements **S** and **T** in the example can be seen in the project's own requirement hierarchy as references.

Browsing 1.6
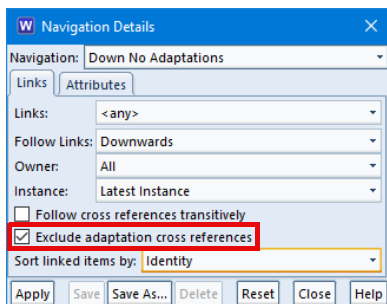
When a database uses adaptations there will be many links between items. When browsing such databases, it is helpful to not follow adaptations' cross references except when you know you want to use them, such as to find library items that have been adapted by projects.

To avoid following adaptations' cross references, always use a navigation which will exclude adaptation cross references, such as the **Down No Adaptations** navigation provided with all Cradle systems. You can exclude adaptation cross references by simply selecting a checkbox in the setup of the navigation, as shown.

Adaptations for Two Types of Item 2

This is the typical case where adaptations are helpful. A hierarchy of items exists in the library and the project has a hierarchy of a different type of item. The project will:

- Create its own items
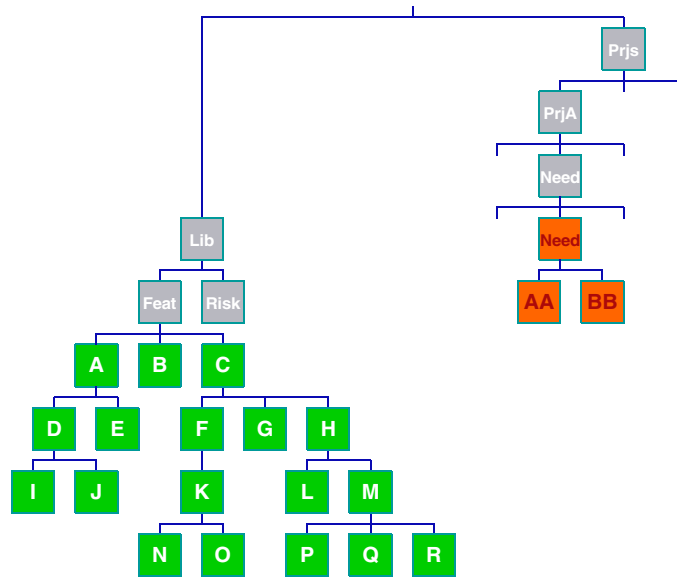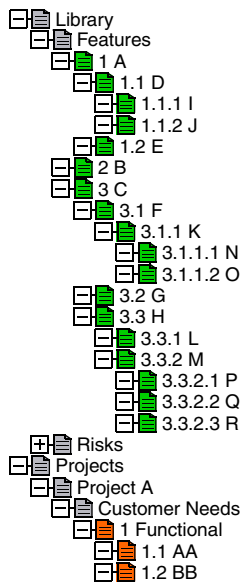- Reference items from the library

- Create its own adaptations of library items

In this case, two item types are involved, and all other project information only links to the project information, not information in the library.

Typical Data 2.1

A project has customer needs as **NEED** items and creates features as **FEATURE** items to satisfy these needs. The features may be derived from a set of standard features held in the library. All other project information will only link to features *in the project's own hierarchy*.

A typical set of information could be:

Schema Setup 2.2

Define four link types with names such as:

- **HAS CHILD**, for the parent-child links
- **REFERENCE TO**, for the references from projects into the library
- **ADAPTATION OF**, for the adaptation links
- **SATISFIED BY**, for links from needs to the features

and set colours in the reference and adaptation link types to colour items in the UI.

Enable hierarchical options and do not allow reordering:

- To become top-level items
- Outside an item's own hierarchy

Enable adaptations for **FEATURE**s and **NEED**s and specify the adaptations and references link types for both item types. The figure to the left shows the setup for **FEATURE** items. The setup
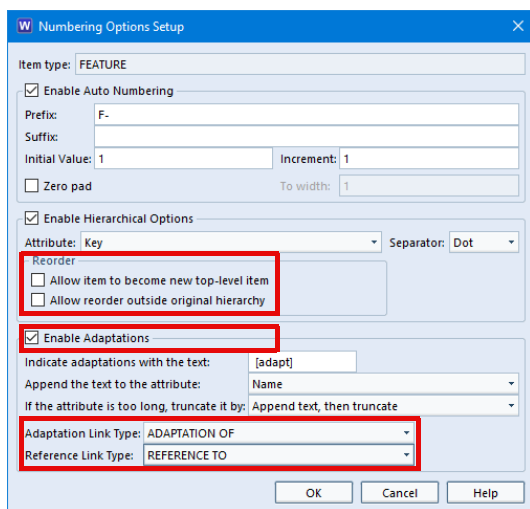
for **NEED**s is the same.

Define the link rules for these items and link types to be:

Table 2: Link Rules to Adapt 2 Item Types			
From Item	**To Item**	**Link Type**	**Default**
FEATURE	FEATURE	HAS CHILD	Y
FEATURE	FEATURE or \<any\>	REFERENCE TO	
FEATURE	FEATURE	ADAPTATION OF	
NEED	NEED	HAS CHILD	Y
NEED	NEED or \<any\>	REFERENCE TO	
NEED	NEED	ADAPTATION OF	
NEED	FEATURE	SATISFIED BY	

Cradle will create these rules automatically for you.

Alternative Schemas 2.3

When items of type **A** are linked to items of type **B**:

- To reference **B**s from **A**s, enable adaptations for **A**s
- To create adaptations of **B**s from **A**s, enable adaptations for the **B**s

If adaptations are enabled for **NEED**s, but not **FEATURE**s, you can create a reference from a **NEED** to a **FEATURE**.

If adaptations are enabled for **FEATURE**s and not **NEED**s, you can create a link from a **NEED** to an adaptation of a **FEATURE**.

If adaptations are enabled for **FEATURE**s and **NEED**s, you can:

- Create a reference from a **NEED** to a **FEATURE**
- Replace a referenced **FEATURE** by an adaptation
- Create a link from a **NEED** to an adaptation of a **FEATURE**, doing the first two of these operations in sequence

You can also create any cross references allowed by the schema's link rules, such as **SATISFIED BY** links from **NEED**s to **FEATURE**s.

Creating New Items 2.4

The project creates new items by selecting existing **NEED** items and using the Cradle operations:

- **New** → **Child**, to create a child of an existing **NEED**
- **New** → **Sibling**, to create a sibling of an existing **NEED**
- **New** → **Hierarchy**, to create a hierarchy of **NEED**s

New items will be linked to the existing items by **HAS CHILD** links, the default link type for cross references between **NEED** items.

Use a Library Item 2.5

To reference a library feature from the project's needs hierarchy:
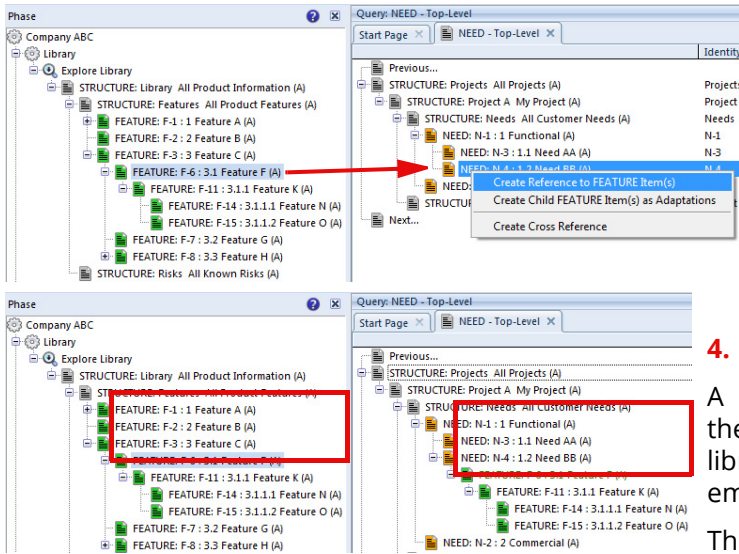
1. Open the library hierarchy and find the item to be referenced
2. Open the project hierarchy and find where the referenced item should be placed
3. Drag and drop the library item onto the project item and choose **Create Reference…**
4. Click **OK** in the next dialog

A cross reference is created from the project item library item. The library item is shown in **green** to emphasise that it is a reference.

The expansion of the library feature is now part of the project's item hierarchy, so features **K**, **N** and **O** in the example can be seen in the project's own hierarchy.

In most cases, the referenced feature and its expansion would be read-only to the user who has referenced them.

Adapt a Library Item 2.6

If the project wants to change an item referenced from the library, to:
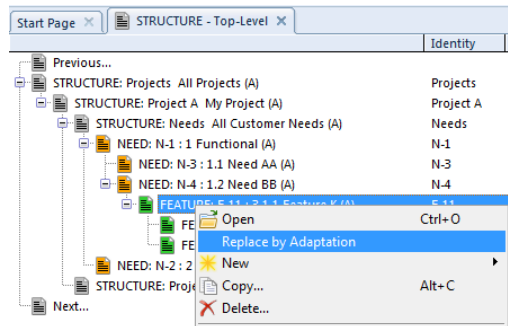
• Change the item itself, the **FEATURE** in this case
• Change the items that are linked to the item
• Change any of the linked items

it cannot change the library item because it may be used elsewhere, and it should be read-only to the project. So the project must create an adaptation of the library item, by selecting it and:

• Right-click and **Replace by Adaptation**, or
• In the **Item** tab, select **Replace by Adaptation**

A dialog will be shown that summarises what links will be created. Select **OK** in this dialog and:

• The feature will be replaced by an adaptation, a copy of the feature that is owned by the user
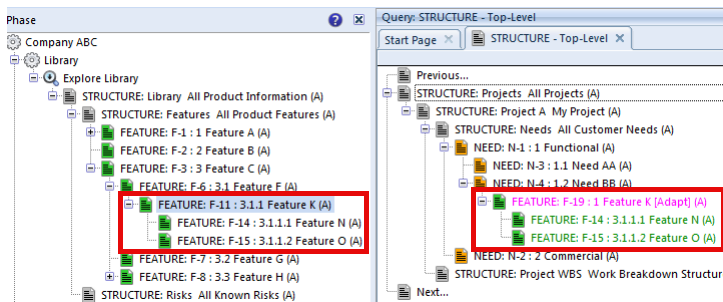
  • The item is named and coloured as an adaptation
  • The item is linked into the project's needs hierarchy
  • The item is linked by references to the same items to which the library feature is linked

So the original library feature **K**

has been replaced by an adaptation of K that the project can modify as needed, and the expansion of the library feature K is now the expansion of the project's feature K. So features N and O in the example can be seen in the project's own hierarchy.

Summary 3

An organisation can reuse information between projects by simply having a set of items in a library and then creating cross references from these into the projects that use them.

However, this creates a problem that if a project wishes to change a library item that it is using, it cannot change that library item:

- Because other projects may be using it
- Because the project should have read-only access to the library

The adaptations mechanism overcomes these problems, making it easy to reference library information and to make project-specific adaptations of this information.

The adaptations mechanism is part of the **Cradle-PDM** (project data management) module, so it is free to use, and is available in all Cradle systems.