# **Understanding change concept**

Change is a difference, found between the ancestor and a contributor. Every change can be accepted or rejected, as well as have dependent changes. Changes can conflict with each other or be equivalent.

Merging begins with building a composite change tree, which consists of a model, diagramming, and non-model changes.

# **Change types**

Read the following definitions to get familiar with different change types.

#### Addition change

If an element has been added to a contributor, an addition change occurs.

# Deletion change

If an element has been removed from a contributor, a deletion change occurs.

#### **Modification change**

If an element property in a contributor has been modified, a modification change occurs.

Selection is Abstract property value of a class in the ancestor had the default value false and the same property value in a contributor has been changed to the same property value in a contributor has been changed to the same property value in a contributor has been changed to the same property value in a contributor has been changed to the same property value in a contributor has been changed to the same property value in a contributor has been changed to the same property value in a contributor has been changed to the same property value in a contributor has been changed to the same property value in the

# rue, a modification change occurs.

There are three types of modification changes:

- · Addition modification change that occurs when a value is added to a property.
- Deletion modification change that occurs when a value is removed from a property.
- Replacement modification change that occurs when one value is replaced with another. This type of modification change occurs only for properties that have multiplicity less or equal to 1.

# Movement change

If an element owner has been changed in a contributor, a movement change occurs.

🕼s say package A contains some class in the ancestor and package B contains the same class in a contributor. This means that the class has been

moved from package A to package B in the contributor. This case is recognized as a movement change.

When case of the movement change is when an attribute or an operation that has been owned by class A in the ancestor, becomes the attribute or an

operation of class B in a contributor.

### Order change

If the order of elements has been changed in a contributor, an order change occurs. Order changes can occur on elements such as attributes, operations, and other ordered elements. Even if a single element in a collection has changed its place, the order change is applied to the entire collection.

Since an element can have several ordered collections, several order changes can occur on a single element.

🐼s say class A has attributed a, b, and c in the ancestor. The attribute c has been moved up and placed above attribute a in a contributor. This means

that the order of attribute collection in class A has changed in the contributor. This is a case of the order change.

#### Turn off order change detection

is the optional phase of the merge procedure. You may not need to perform it.

Order changes can be skipped while merging. For this, you need to specify names of properties wherein order changes should not be detected.

To turn off the order changes detection in specificproperties

- 1. From the Options menu select Environment. The Environment Options dialog opens.
- 2. Find the Do Not Detect Order Changes for the option under the **Merge** category in the **General** options group.
- 3. In the option value cell, specify names of properties wherein order changes should not be detected while merging.

perty names must be written in camel case, for example, ownedAttribute, ownedElement, and so on.

# Change states

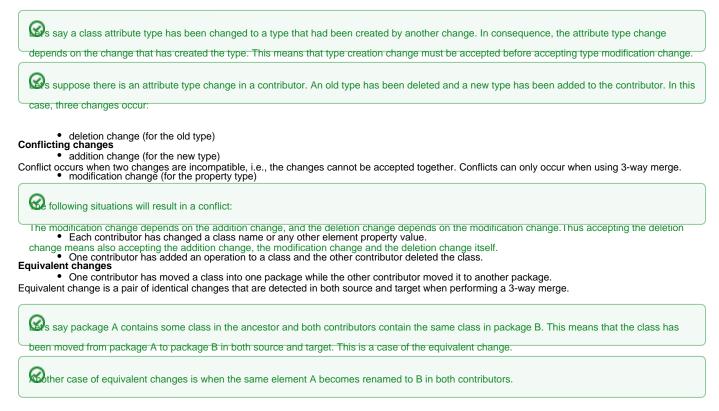
Every change, whether it is addition, modification, deletion, movement, or order change, can be either accepted or rejected. It is a change state.

All accepted changes will be incorporated into the target. Alternatively, they can be rejected and excluded from the target.

## **Dependent changes**

In some cases, other changes have to be accepted or rejected before accepting or rejecting a selected change. In other words, a selected change sometimes depends on other changes and is called a dependent change.

For a better understanding of the concept of dependent changes, study the following examples.



## **Related pages**

- Understanding merge types
- Preparing for merge
- Starting Model Merge
- Analyzing and managing merge results
- Finishing and canceling merge