Understanding change types

Change is a difference, found between the base and compared project versions.

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

Addition change
If an element has been added to a compared project version, an addition change occurs.

Deletion change
If an element has been removed from a compared project version, a deletion change occurs.

Modification change
If an element property in a compared project version has been modified, a modification change occurs.

> If the **IsAbstract** property value of a class in the base had the default value *false* and the same property value in a compared project version has been changed to *true*, 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 compared project version, a movement change occurs.

> Let's say package A contains some class in the base and package B contains the same class in a compared project version. This means that the class has been moved from package A to package B in the compared project version. This case is recognized as a movement change.

> Another case of the movement change is when an attribute or an operation that has been owned by class A in the base, becomes the attribute or an operation of class B in a compared project version.

Order change
If the order of elements has been changed in a compared project version, 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.

> Let's say class A has attributes a, b, and c in the base. The attribute c has been moved up and placed above attribute a in a compared project version. This means that the order of attribute collection in class A has changed in the compared project version. This is a case of the order change.

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

To turn off the order changes detection in specific properties

---

1. From the **Options** menu select **Environment**. The **Environment Options** dialog opens.
2. Find the **Do Not Detect Order Changes for** option under the **Merge and Compare** 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.

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

Dependent change

In some cases, changes depend on other changes and are called dependent changes.

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

Let's say a class attribute type has been changed to a type that had been created by another change. In consequence, the attribute type change depends on the change that has created the type.

Let's suppose there is an attribute type change in a compared project version. An old type has been deleted and a new type has been added to the compared project version. In this case, three changes occur:

- deletion change (for the old type)
- addition change (for the new type)
- modification change (for the property type)

The modification change depends on the addition change, and the deletion change depends on the modification change.