

A mapping is a collection of source and target objects linked together by a set of transformations. These transformations consist of a set of rules, which define the data flow and how the data is loaded into the targets.

A mapping consists of following set of objects



- **Source Definition** - Source definition defines the structure and characteristic of the source, its underlying data types, type of the data source, etc.
- **Transformation** - Transformation objects define how the source data is transformed, and various functions can be applied during the process
- **Target Definition** - Target definition defines the final target where the data will be loaded.
- **Links** - Links connect the source definition to different transformations and target tables. It defines how the data flows from source to target and the transformations.

Why do you need Mapping?

Mapping is an object in Informatica with the help of which you can define how the source data is modified before it reaches the destination or target object. Like if you have employee name as "Bill Clinton" in your source system and in the target system the requirement is to have employee name in the format as "Clinton Bill", such operations can be designed at the mapping level. In basic terms, what you do with the source data is defined at the mapping level.

Mapping is the basic Informatica object with the help of which we can define the data transformation details and source/target object characteristics. Mappings help us to define the data transformation at the individual column levels for each row. Even in a single mapping you can handle multiple sources and targets.

Components of Mapping

Basic components of a mapping are

- Source tables
- Mapping parameters and variables
- Target objects
- Mapping transformations

There are various objects that constitute a mapping. A mapping can consist of **sources, targets, mapping parameter and variables, mapplets, various transformations, and user-defined functions.**

There are various objects that constitute a mapping. A mapping can consist of **sources, targets, mapping parameter and variables, mapplets, various transformations, and user-defined functions.**

- **Mapping Source:** Mapping sources are the objects from where you fetch the source data. It can be a database table, flat file, XML source or COBOL file source
- **Mapping target:** Mapping target is our destination objects where final processed data gets loaded. Mapping target can be a relational table of a database, a flat file or XML file. Sources and targets are mandatory in any mapping, their type can differ
- **Mapping Parameters and Variables:** Mapping parameters and variables helps you to create temporary variable objects which will help you to define and store temporary values while mapping data processing. Mapping parameters and variables are optional users defined data types, which can be created for a mapping and can be referenced and updated for a specific requirement. We will learn more about mapping parameters and variables in this section
- **Mapplets:** They are objects which consist of a set of transformation, source or targets. Mapplets are generally created to reuse the existing functionality of a set of transformations. It can be used in any no of mappings.

A workflow is a group of instructions/commands to the integrations service. The integration service is an entity which reads workflow information from the repository, fetches data from sources and after performing transformation loads it into the target.

Workflow - It defines how to run tasks like **session task**, **command task**, **email task**, etc.

To create a workflow

1. You first need to create tasks
2. And then add those tasks to the workflow.

A Workflow is like an empty container, which has the capacity to store an object you want to execute. You add tasks to the workflow that you want to execute. In this tutorial, we are going to do following things in workflow.



Workflow execution can be done in two ways

A Workflow is like an empty container, which has the capacity to store an object you want to execute. You add tasks to the workflow that you want to execute. In this tutorial, we are going to do following things in workflow.



Workflow execution can be done in two ways

- **Sequence** : Tasks execute in the order in which they are defined
- **Event based** : Tasks gets executed based on the event conditions.