

# Model General Principles

## Set up your environment

Considering you already have your own Niagara module. Now you want to create a new connector to import the model from an external service.

You have to import the Active Framework modules needed in the gradle file of your own module:

### modelDemo-rt.gradle

```
niagaraModule {
    preferredSymbol = "modelDemo"
    moduleName = "modelDemo"
    runtimeProfile = "rt"
}

dependencies {
    compile "Tridium:baja:4.6"
    compile "Tridium:nre:4.6"
    compile "BTIB:btibCore-rt:46"
    compile "BTIB:btibStructure-rt:46"
    compile "BTIB:btibConnector-rt:46"
}
```

The content you need from each module is:

- btibCore: it contains all the tools and the basic classes of the Active Framework.
- btibStructure: it contains the model classes (Aspect, Definition, Node) and the engine used to send operations to manage nodes
- btibConnector: it contains the connectors architecture

## Understand the Niagara configuration



Unknown macro: 'drawio'

In Niagara, the components interact as following:

- The Model **Connector** establishes the connection to the service and provides general actions like ping or refresh. It can also contain general methods to communicate with the service.
- The Model **Importer** is dropped within an aspect, it will contain the mapping with definitions and will be in charge of the creation/edit/deletion of the nodes. It reaches the service through the connector and contains only methods dedicated to the model import.

## Understand the class architecture



Unknown macro: 'drawio'



Unknown macro: 'drawio'

Now let's see how to implement your connector and your importer.