# MyBatis 1.0

**On this page:**




**Summary:** This document provides basic information about the extension providing **MyBatis (for Java/.NET)** support.


## What's new ?

**Changes in 1.0.0-alpha2:**

- Initial support for MyBatis for .NET
- Support for .NET with Mapper Xml file

**Changes in 1.0.0-alpha1:**

- Initial release for MyBatis with support for Java.

## Description

This extension provides support for **MyBatis for Java / .NET using Mapper XML file containing SQL queries**.

## In what situation should you install this extension?

This extension should be installed when analyzing a Java/.NET project that uses a MyBatis framework, and wanting to view a transaction consisting of MyBatis objects with their corresponding links. This version supports only **MyBatis for Java/.NET** using Mapper XML file containing SQL queries. Links to corresponding database tables can also be resolved, provided that the **MySQL** database has been extracted and DDL has been created.

## Supported MyBatis versions

The following table displays the supported versions matrix:

| Language Supported | Version | Support |
|---|---|---|
| Java | 3.0.2 | Mapper XML file |
| .NET | 4.5.2 | Mapper XML file |

## CAST AIP Compatibility

This extension is compatible with:

| CAST AIP release | Supported |
|---|---|
| 8.3.x | ✓ |
| 8.2.x | ✓ |
| 8.1.x | ✓ |
| 8.0.x | ✓ |
| 7.3.4 and all higher 7.3.x releases | ✓ |

# Supported DBMS servers

This extension is compatible with the following DBMS servers:

| CAST AIP release | CSS | Oracle | Microsoft |
|---|---|---|---|
| All supported releases | ✓ | ✗ | ✗ |

# Prerequisites

| | |
|---|---|
| ✓ | An installation of any compatible release of CAST AIP (see table above) |
| ✓ | MySQL database should already be extracted and DDL created. |

# Dependencies with other extensions

Some CAST AIP extensions require the presence of other CAST AIP extensions in order to function correctly. The **MyBatis** extension requires that the following other CAST extensions are also installed:

- **CAST AIP Internal extension** (internal technical extension)
- SQL Analyzer extension - this extension will handle the MySQL DDL

When using the CAST Extension Downloader to download the MyBatis extension and the **Manage Extensions** interface in CAST Server Manager to install the extension:

- the **CAST AIP Internal extension** will be automatically downloaded and installed for you. You do not need to do anything.
- the SQL Analyzer extension - is not configured as a "dependent" extension and will therefore not be automatically downloaded and installed for you. However:
  - When using **CAST AIP 8.3.x**, the SQL Analyzer extension is installed by default, therefore you do not need to do anything.
  - When using **CAST AIP 8.2.x**, the SQL Analyzer extension is not installed by default, therefore you will need to download and install it manually if you want to handle the SQL source code in your MyBatis application.

# Download and installation instructions

Please see:

- Download an extension
- Install an extension

The latest release status of this extension can be seen when downloading it from the CAST Extend server.

# Packaging, delivering and analyzing your source code

Once the extension is installed, no further configuration changes are required before you can package your source code and run an analysis. The process of packaging, delivering and analyzing your source code is as follows:

## Packaging and delivery

Note that the **MyBatis** extension does not contain any CAST Delivery Manager Tool **discoverers or extractors**, therefore, no "MyBatis" projects will be detected by the DMT. You therefore need to manually create Analysis Units in the CAST Management Studio - this is explained below.

You should ensure that your code is organised as follows:

**Java**

- one folder containing all the **Java** related code and the **XML** related files
- one folder containing the **MySQL DDL** extraction

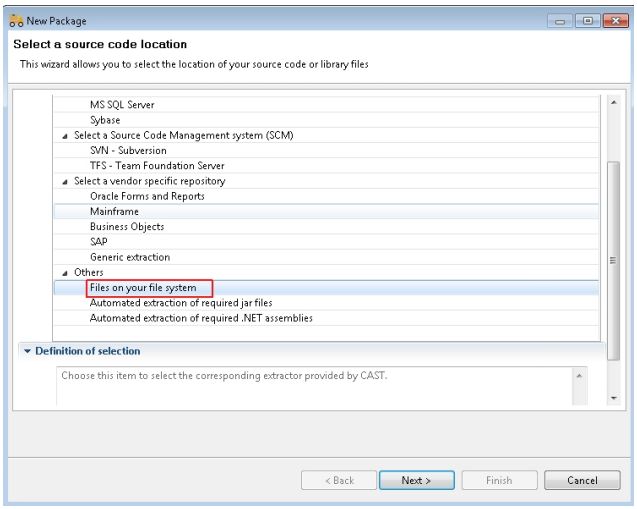
**.NET**

- one folder containing all the **.NET** related code and the **XML** related files
- one folder containing the **MySQL DDL** extraction

Using the CAST Delivery Manager Tool:

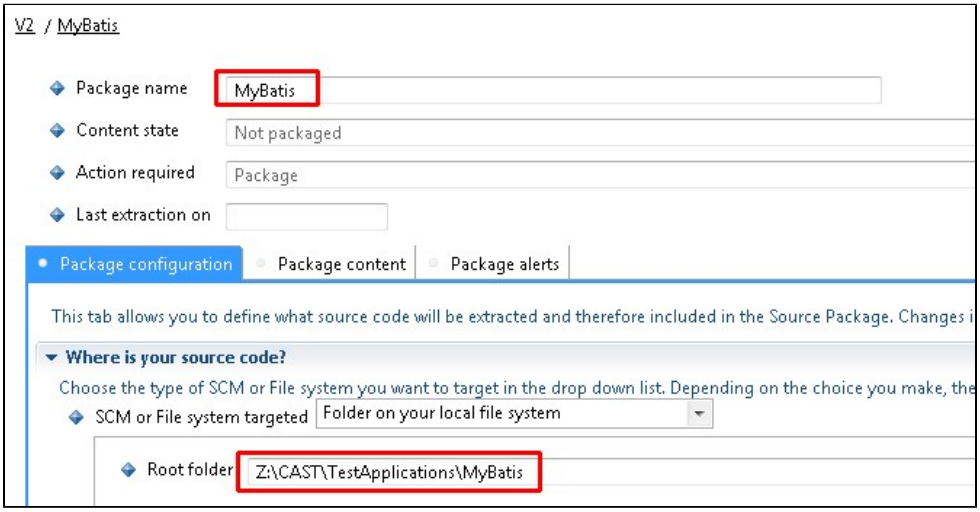
- Create a new **Version**
- Create a new **Package** using the **Files on your file system** option:

*Click to enlarge*




- Define a **name** for the package and the **root folder** of your Application source code - this root folder should contain the folder for the **Java/XML files or .NET/XML files** and the folder for the **MySQL DDL**:

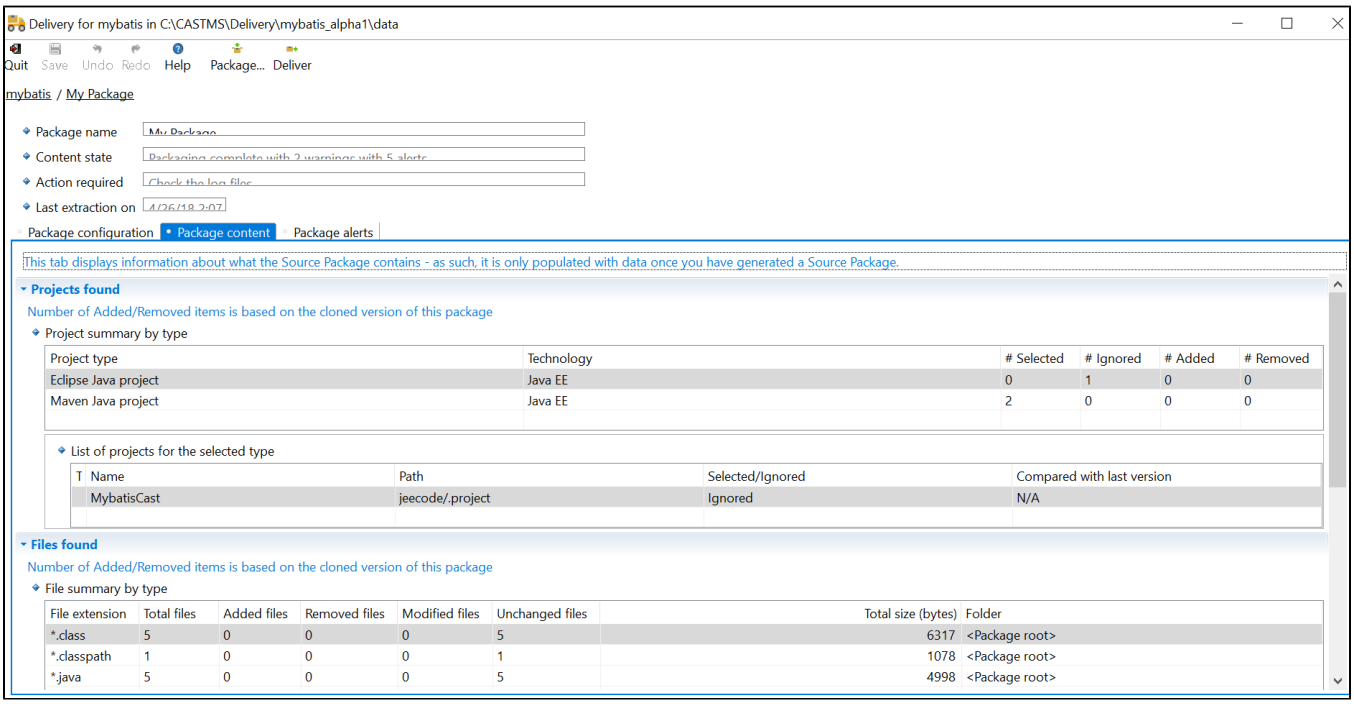
*Click to enlarge*



Note that:

- When working with .NET, the package described above is equivalent to Package no. 1 described in .NET Analyzer - Packaging, delivering and analyzing your source code.
- you may need to create additional packages. For example, for **.NET** an additional package will be required to package the **.NET Assemblies**. Please refer to the .NET Analyzer extension or JEE Analyzer extension for more information.

- Run the **Package action**: the CAST Delivery Manager Tool will find Eclipse/Maven Java or .NET related "projects" associated with the MyBatis application source code - this is the **expected behavior**. However, if your Java/.NET related source code is part of a larger application, then other projects may also be found during the package action:



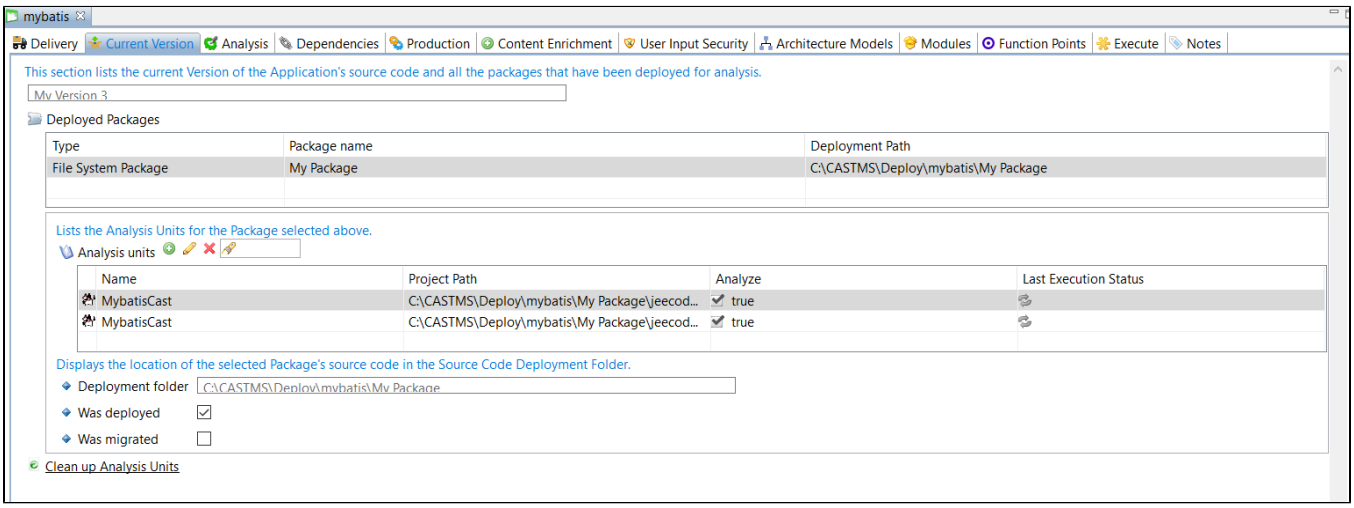

- Finally, **Deliver** the version.

## Analyzing

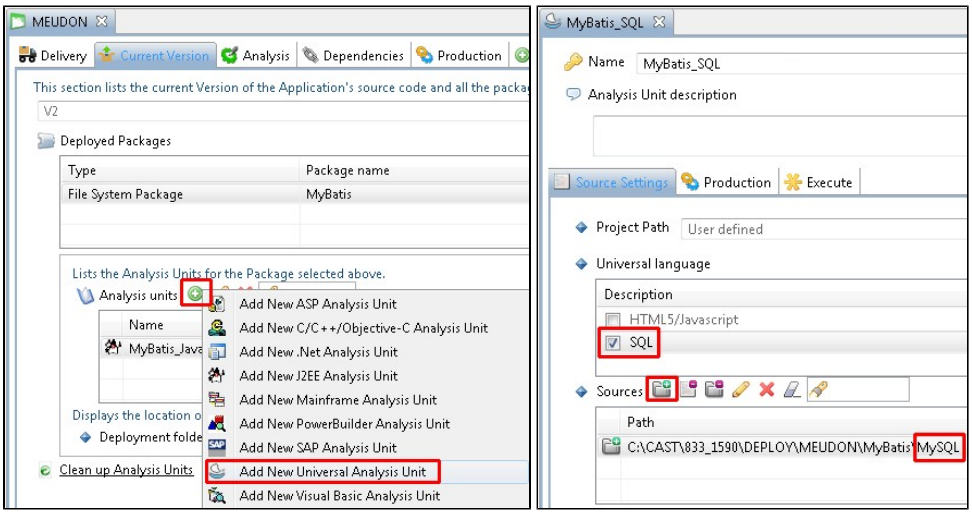
Using the CAST Management Studio:

- Accept and deploy the **Version** in the CAST Management Studio. **JEE / .NET Analysis Units** will be created automatically - this is the **expected behavior**. However, if your MyBatis related source code is part of a larger application, then other Analysis Units may be created automatically:

*Example for JEE*

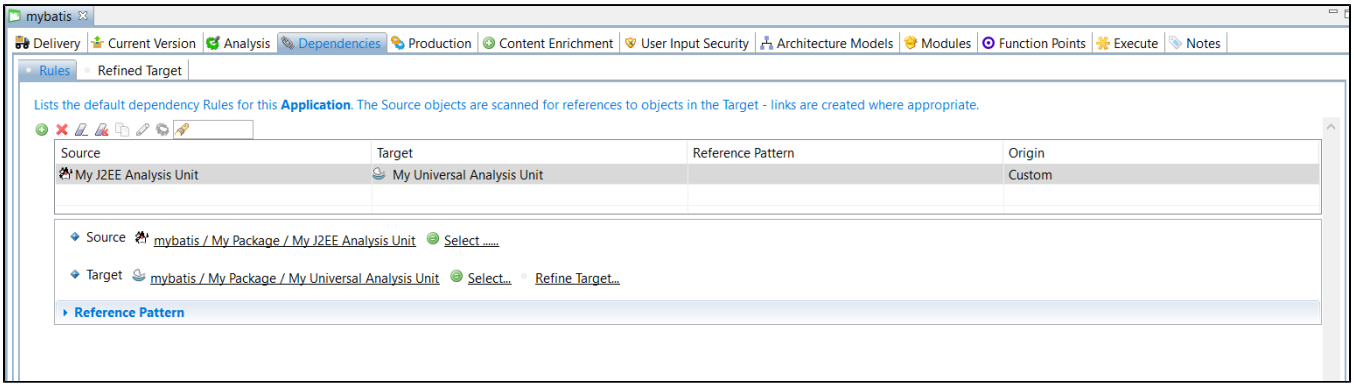


- In the **Current Version tab**, you now need to add manually **one Analysis Units for the  MySQL source code**:
  - Add a **Universal Analyzer Analysis Unit** for your **MySQL source code**, selecting the **Add new Universal Analysis Unit** option and then:
    - ticking the **SQL** option
    - including the folder containing the MySQL DDL:




- Now move into the **Dependencies** tab within the **Application editor**, and create a dependency between the **JEE Analysis Unit** or **.NET Analysis Unit** as **Source** and the **Universal Analysis Unit** as **Target**:

*Example for JEE*

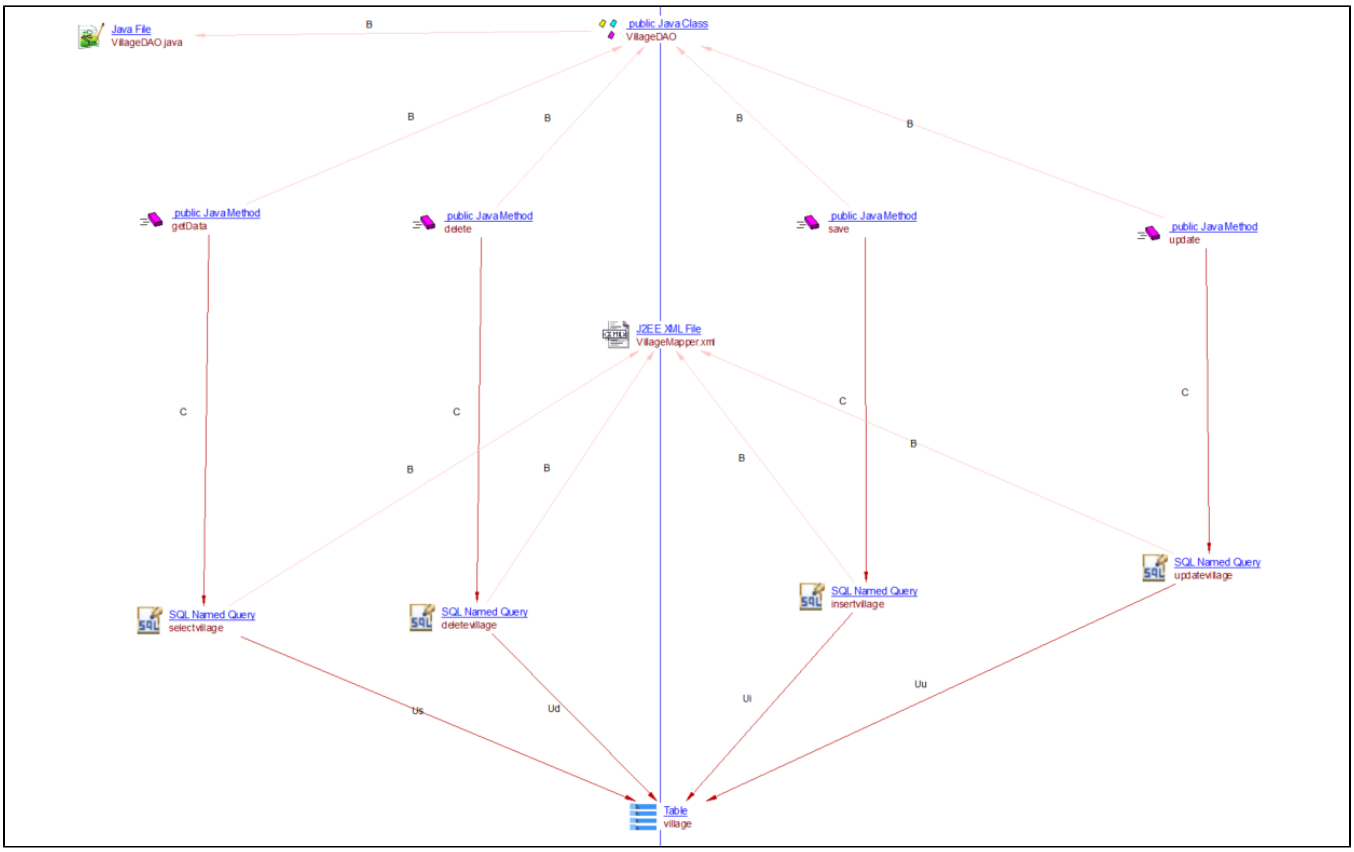



- Finally run a **test analysis** on the Analysis Unit before you generate a **new snapshot**.

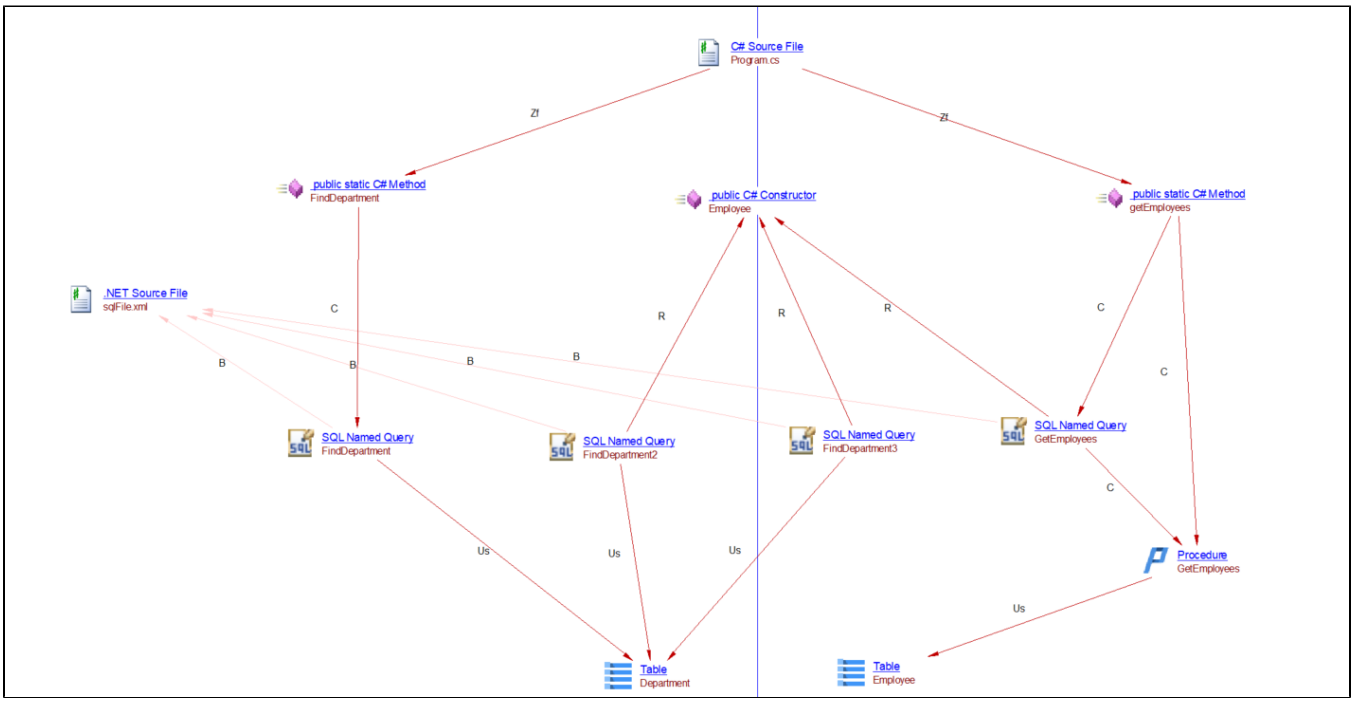
# What results can you expect?

Once the analysis/snapshot generation has completed, you can view the results in the normal manner. Below are the transactions obtained after analysis as shown in CAST Enlighten:

## For Java




For .NET




# Objects

The following specific objects are displayed in CAST Enlighten:

- **Mapper xml file** - this object is created by the JEE analyzer/.NET Analyzer.

- **SQL_Named query** - this object(s) is created by the **MyBatis extension**.

# Links

Various links are created but the following are the links specific to the **Mybatis extension**:

## For Java

- **Call** link between a **Java Method** object and an **SQL_Named query** object.
- **Use** (typed with Select, Delete, Insert, Update) link between an **SQL_Named query** object and a **Table** object.

## For .NET

- **Call link** between a .**NET Method** object and an **SQL_Named query** object.
- **Use** (typed with Select, Delete, Insert, Update) link between an **SQL_Named query** object and a **Table** object.
- **Refer link** between a **.NET Constructor** object and an **SQL_Named query** object.

# Limitations

The following cases are not supported:

- MyBatis for Java using annotations.
- MyBatis for Java using Spring.