

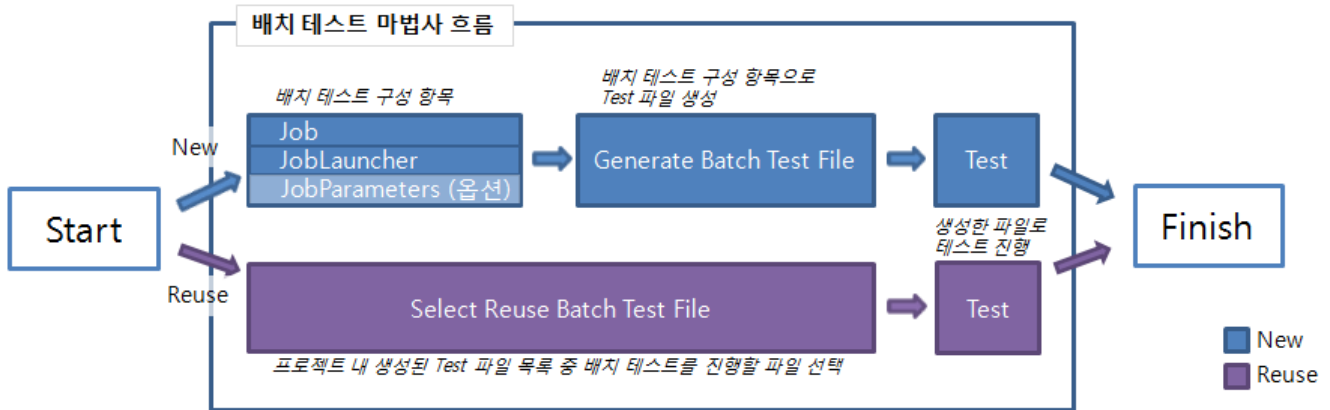
Batch Job Test Wizard

Outline

eGov Standard Framework provides a test wizard that can perform simple tests by using created batch files.

Description

During batch test, create test files by selecting necessary files and perform test via JUnit.

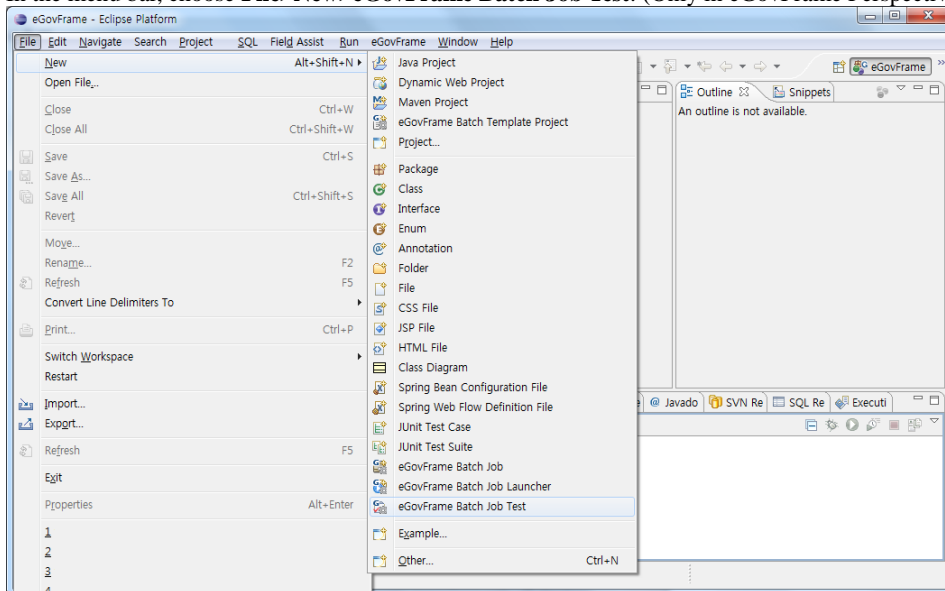


✓ Test files created by the Batch Test Wizard can also be used for separate JUnit tests.

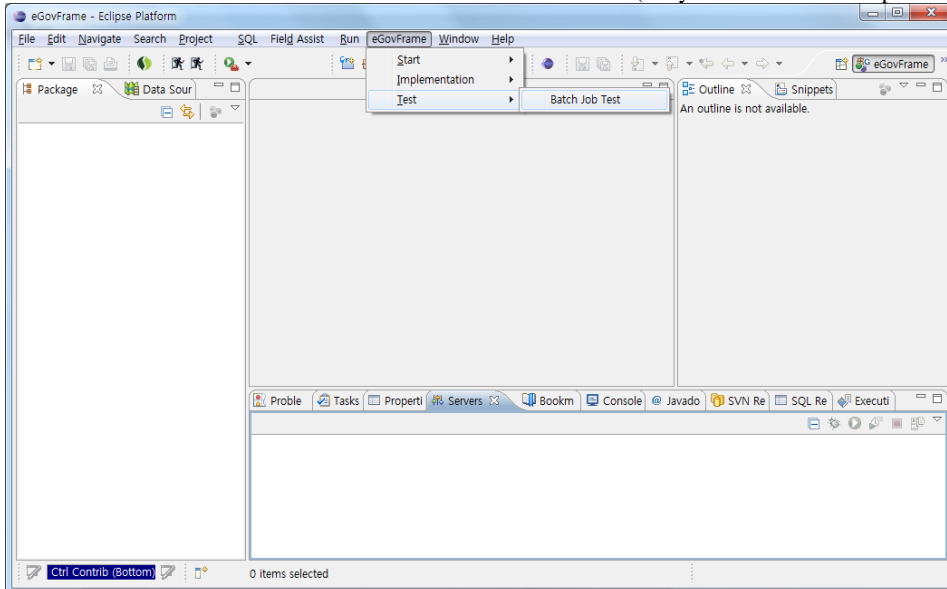
How to Use

1. Starting Batch Test Wizard

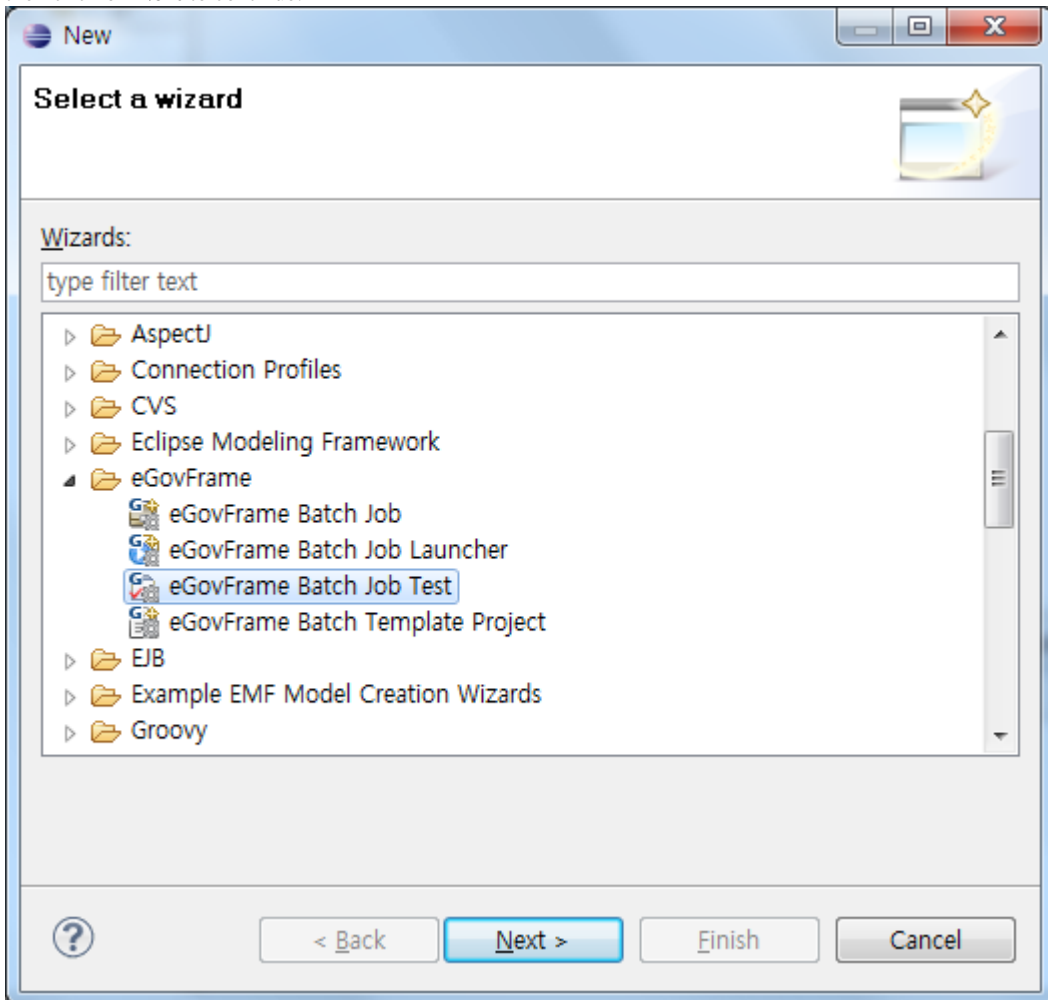
- In the menu bar, choose **File>New>eGovFrame Batch Job Test**. (Only in eGovFrame Perspective)



- Or choose **eGovFrame>Test>Batch Job Test** in the menu bar. (Only in eGovFrame Perspective)

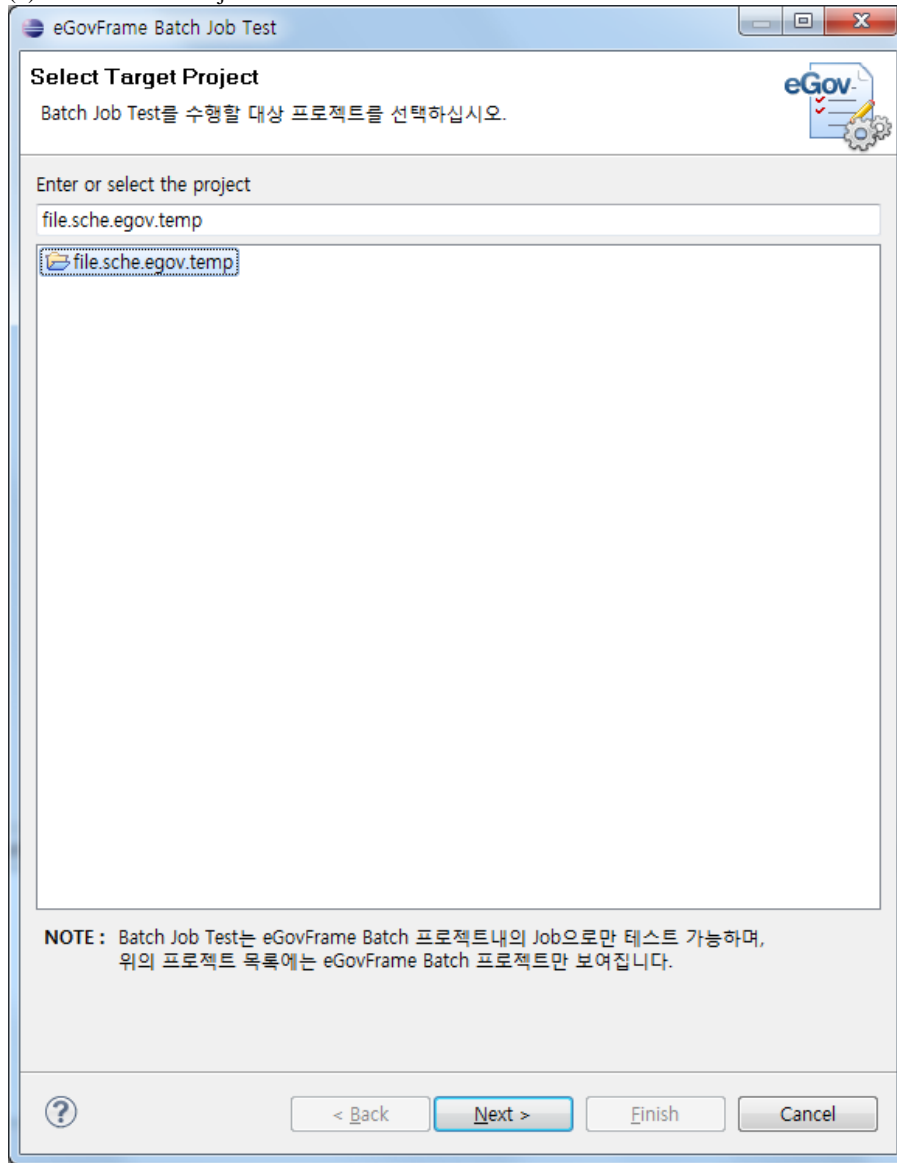


- You can also use the hotkey, **Ctrl+N** to access the "create new" wizard and select **eGovFrame>eGovFrame Batch Job Test** then click on **Next** to continue.

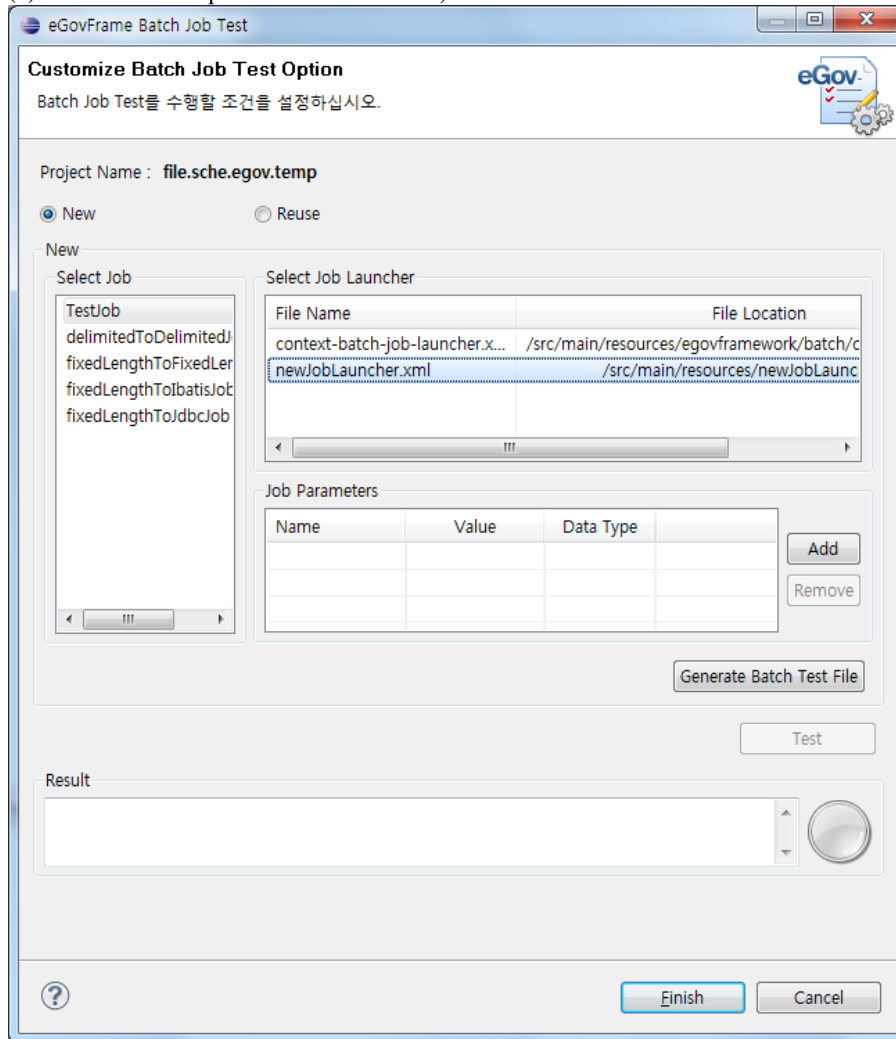


2. Perform batch test by selecting or configuring batch test conditions.
 - Performing a New Batch Test.

- (1) Select a Batch Project that was created based on eGovFrame and click **Next**.

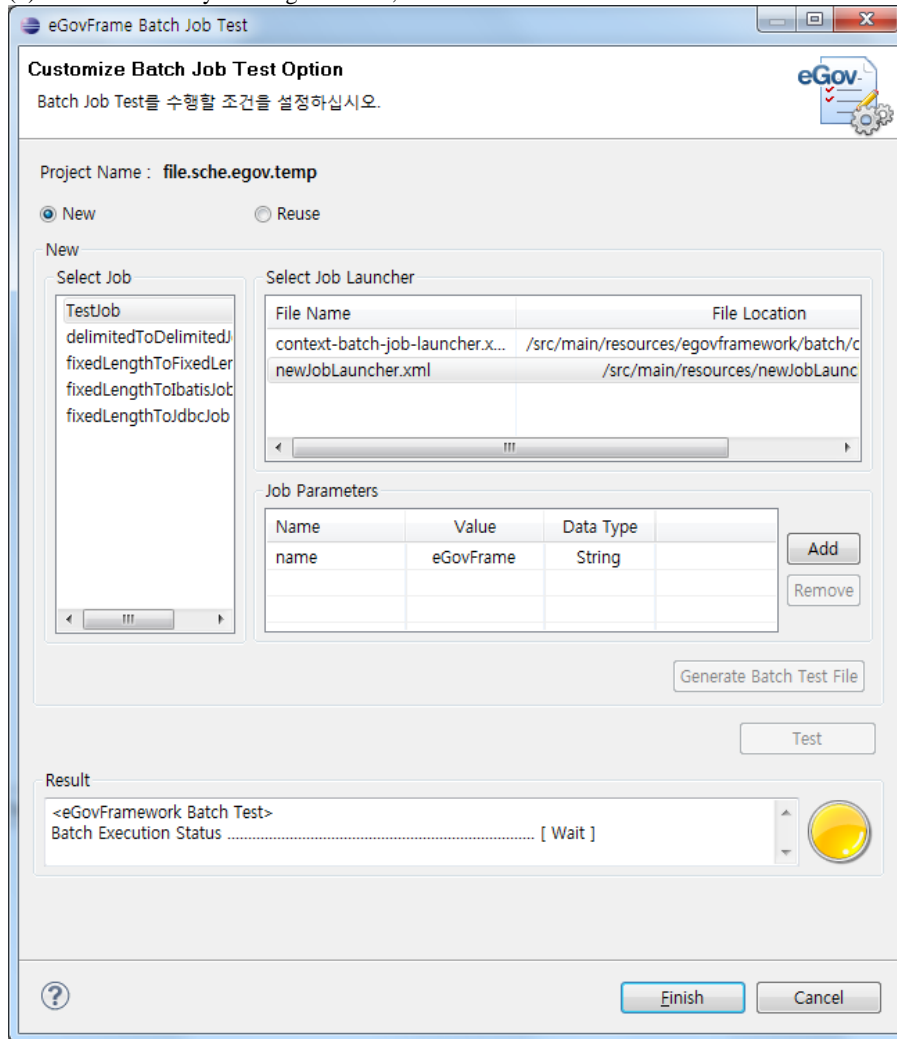


- (2) Select an item to perform test from **Job, Job Launcher Items** that are located within the selected project.

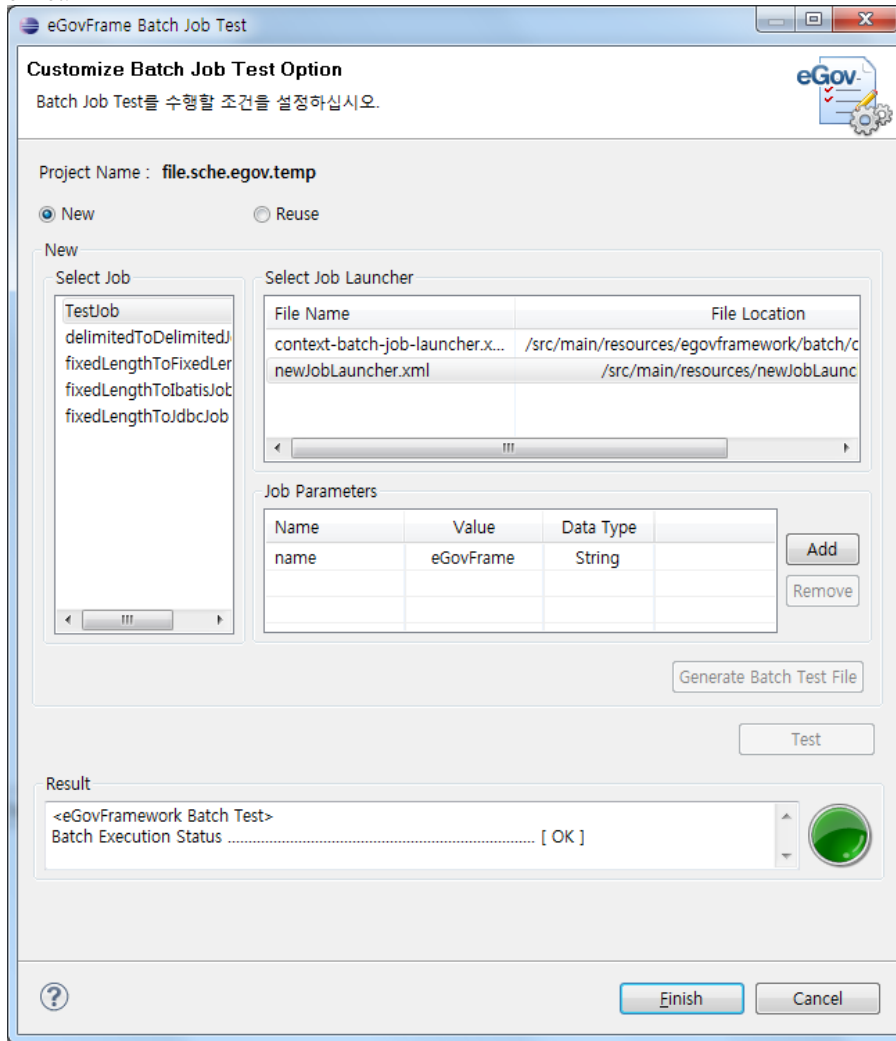


- (3) To add Parameters to the batch test, click on **Add** located on the right side of [Job Parameters](#).
 ✓ When using basic timestamp as a Parameter, there is no need to add an additional item in Job Parameters.

- (6) After successfully creating a test file, click on the **Test** button to test the batch.

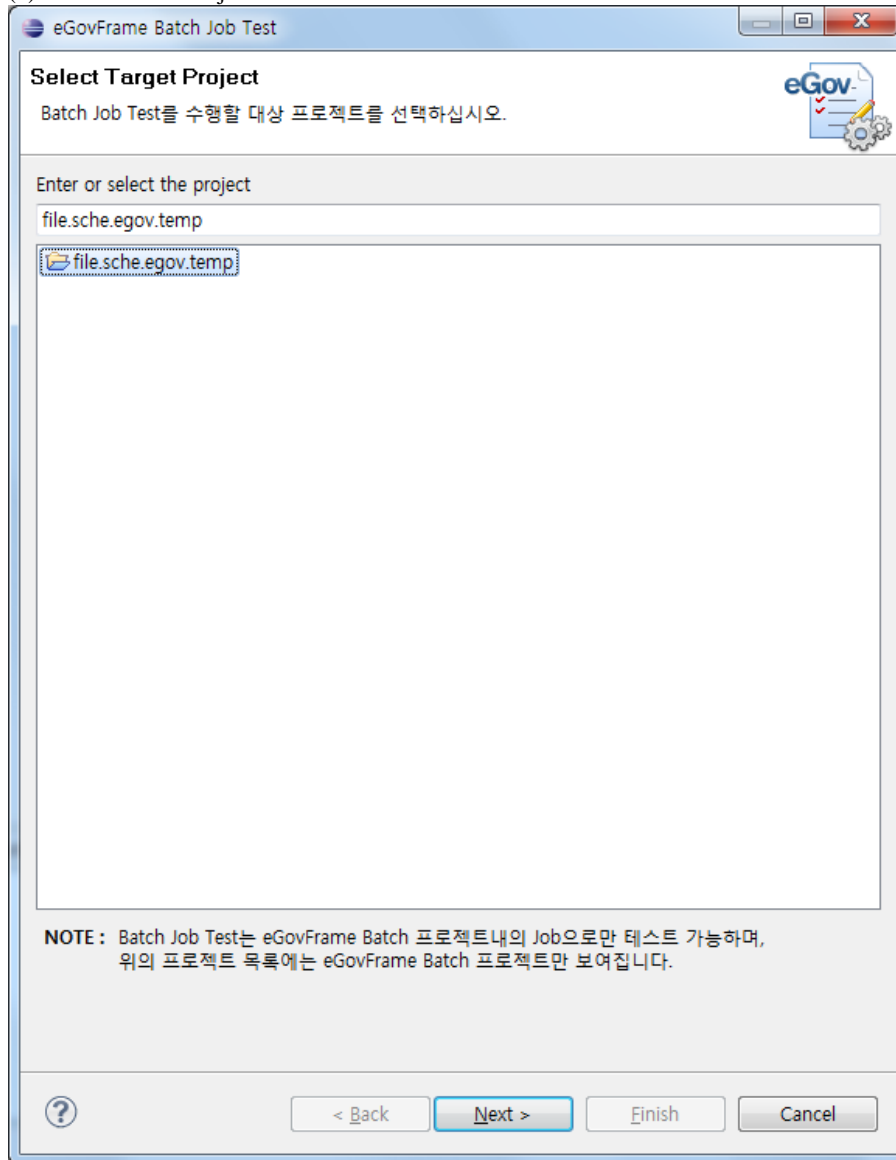


- (7) Click on **Result** located on the bottom of the Batch Test Wizard to check whether the batch test was successful or not.

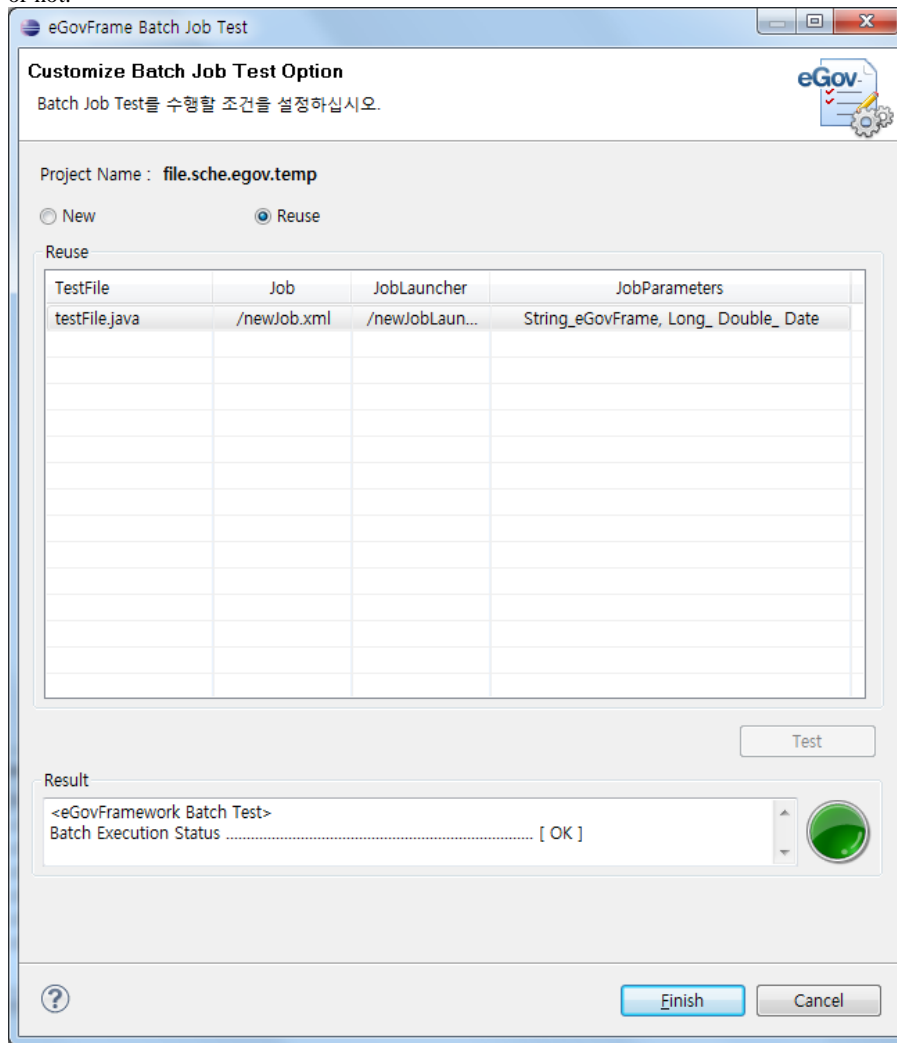


- Performing a Reuse Batch Test.

- (1) Select a Batch Project that was created based on eGovFrame and click **Next**.

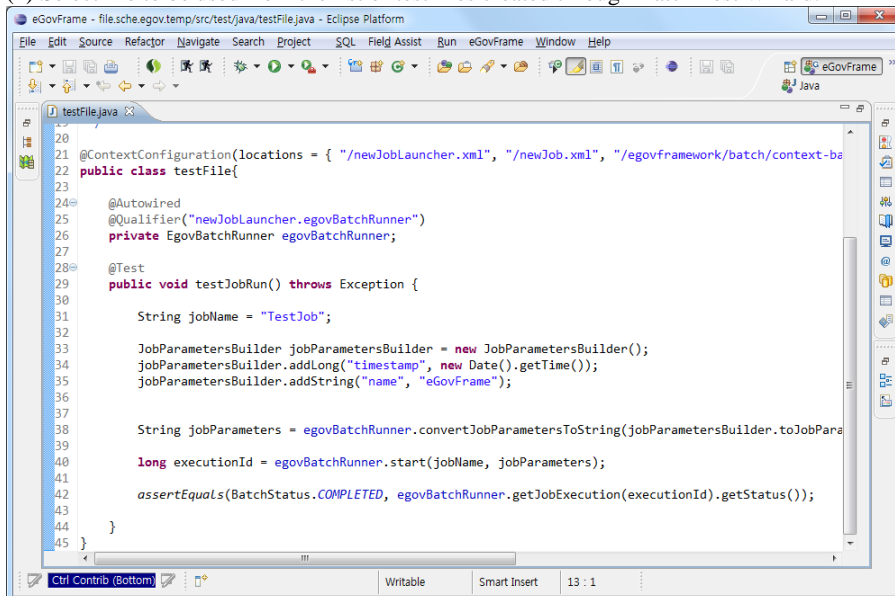


- (5) Click on **Result** located on the bottom of the Batch Test Wizard to check whether the batch test was successful or not.

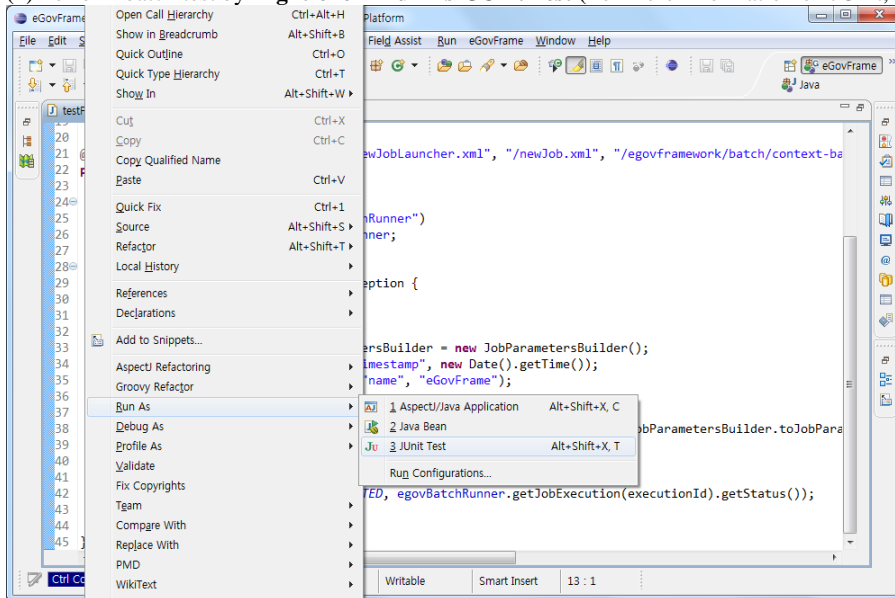


3. Test files created by the Batch Test Wizard can still be used after the Wizard is closed via JUnit. (Detailed information on files can be found in [Remarks](#) located on the below.)

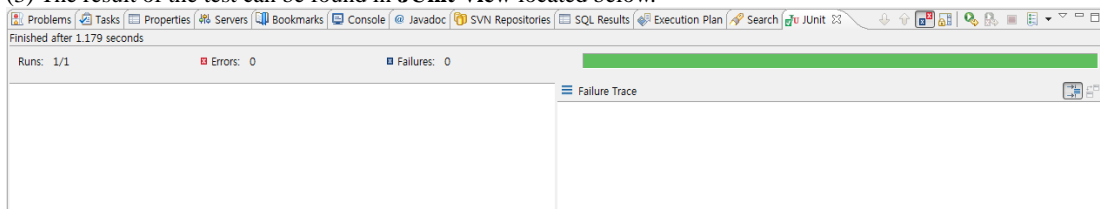
- (1) Select file to be used from the list of test files created through Batch Test Wizard.



- (2) Perform batch test by **Right-click>Run As>JUnit Test** (For more information on JUnit, visit [this link](#).)



- (3) The result of the test can be found in **JUnit View** located below.



Remarks

- Components of the files created through Batch Test Wizard

@RunWith(SpringJUnit4ClassRunner.class)

```
@ContextConfiguration(locations = {"/newJobLauncher.xml", "/newJob.xml", "/egovframework/batch/context-batch-job-launcher.xml", "/egovframework/batch/context-batch-datasource.xml"})
public class TestFile {
```

```
    @Autowired
```

```
    @Qualifier("newJobLauncher.egovBatchRunner")
    private EgovBatchRunner egovBatchRunner;
```

```
    @Test
```

```
    public void testJobRun() throws Exception {
```

```
        String jobName = "TestJob";
```

```
        JobParametersBuilder jobParametersBuilder = new JobParametersBuilder();
        jobParametersBuilder.addLong("timestamp", new Date().getTime());
        jobParametersBuilder.addString("name", "eGovframe");
```

```
        String jobParameters = egovBatchRunner.convertJobParametersToString(jobParametersBuilder.toJobParameters());
```

```
        long executionId = egovBatchRunner.start(jobName, jobParameters);
```

```
        // If an Async configuration exists within the Launcher file, Thread.sleep(10000)
```

```

//
if(egovBatchRunner.getJobExecution(executionId).getExitStatus().toString().contains("UNKNOWN")){
//      Thread.sleep(10000);
// }

    assertEquals(BatchStatus.COMPLETED,
egovBatchRunner.getJobExecution(executionId).getStatus());

}
}

```

Category	Item	Description	Input Value
Synchronous / Asynchronous	@RunWith	Registration for test through JUnit provided by Spring	SpringJUnit4ClassRunner.class
Synchronous / Asynchronous	@ContextConfiguration	Registration of location of components such as Job and Job Launcher for Batch Tests.	{"JobLauncher Directory", "Job Directory", "(JobRepository Directory)", "(Datasource Directory)"}
Synchronous / Asynchronous	@Qualifier	Combines component items such as Job and JobLauncher.	EgovBatchRunner Class Registration Bean ID registered in selected JobLauncher
Asynchronous	Thread.sleep(10000);	Registration for proper functioning if a asynchronous configuration was made in the selected Job item.	10000

✔Thread.sleep(10000) : Exited right after job is complete since it ordinary is run as a Java Application. Therefore, Thread.sleep was added to provide a stable and successful asynchronous Job. The Number 10000 was selected in consideration of ordinary Job performance duration. If the job test process is expected to take more time, the number should be changed accordingly.