

# TestCase Generator

## Summary

TestCase Generator is an Eclipse IDE plug-in for auto-creating test cases, and it provides eGovFrame architecture per-tier test case template creating, and code examples.

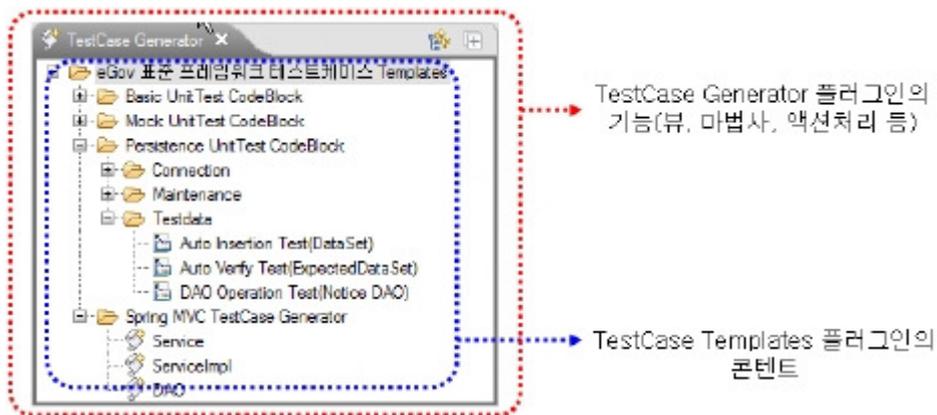
## Description

TestCase Generator is a tree-view-style plug-in, and works side by side with TestCase plug-in.

TestCase Template plug-in is provided as a file system so that independent projects can compose and distribute it.

TestCase Generator comprises as follows.

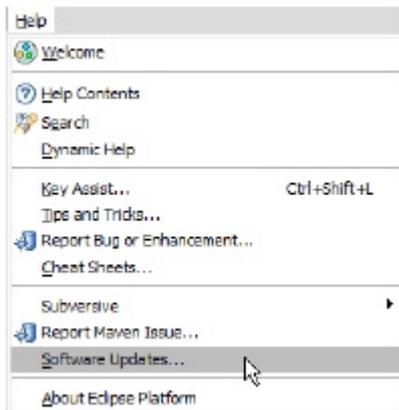
- TestCase Generator View: reads in TestCase Templates plug-in files, and creates templates base on them, as well as viewing code examples.  
Can be run independently, but does not fully work that way because it needs TestCase Templates plug-in's template files in order to display its tree views.
- Template files can be modified and distributed in sync to project schedules, and can be updated regularly through updates.



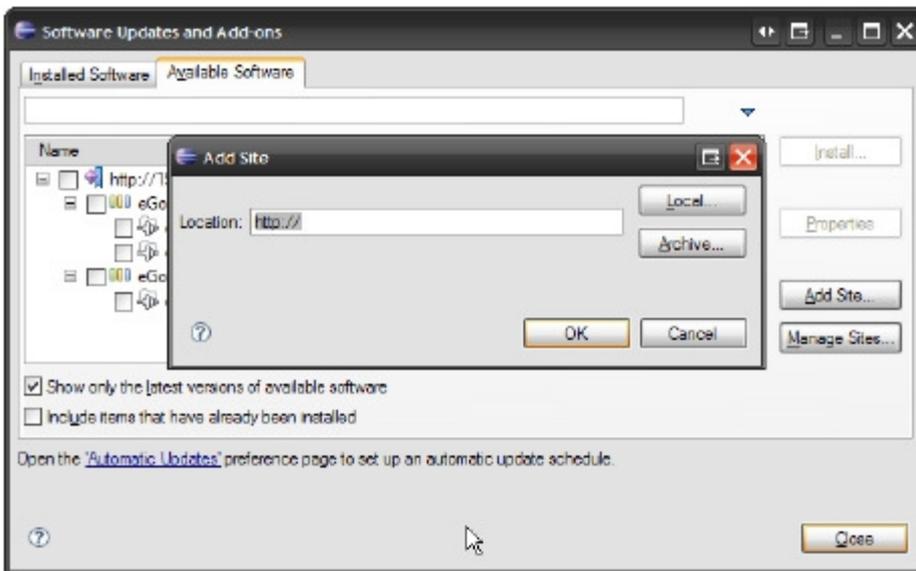
## Environmental settings

Use Eclipse plug-in update to install TestCase Generator.

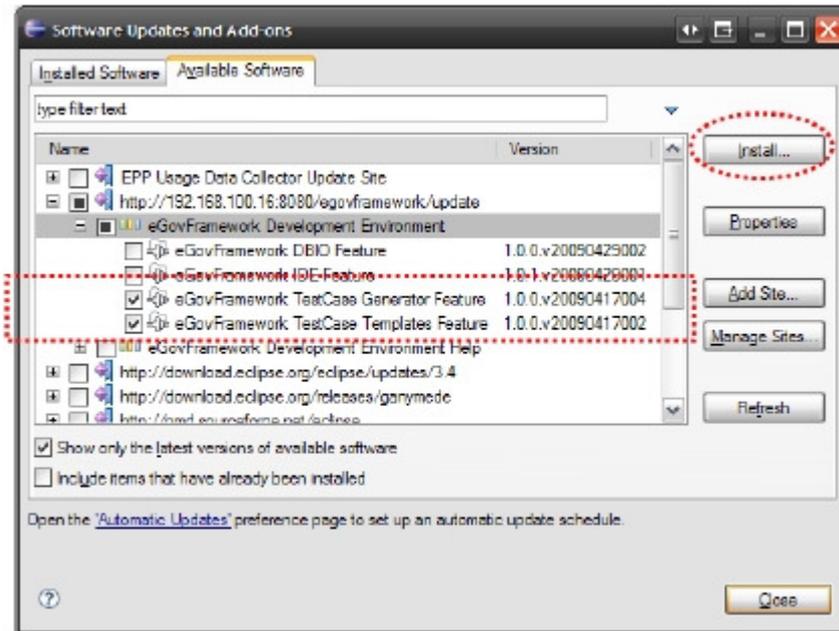
1. Choose Help > Software Updates in the menu.



2. Click on Available Software
3. Choose 'Add Site...' to enter the URL and to update the plug-in (<http://192.168.100.16:8080/egovframework/update> [<http://192.168.100.16:8080/egovframework/update>])



4. In the eGovFrame Plug-in list, choose TestCase Generator plug-in and TestCase Templates plug-in, then click Install... button



5. Follow the installation guide, then restart Eclipse

No further settings are required.

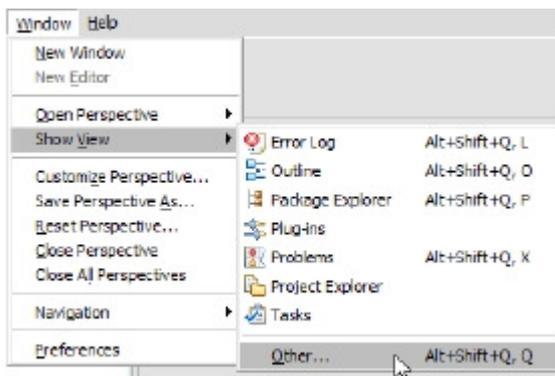
## Manual

### Opening TestCase Generator view

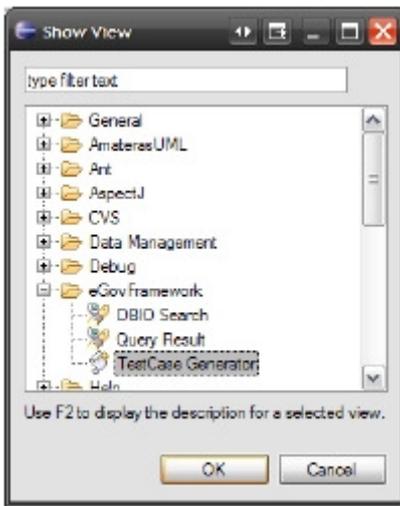
A brief guide on how to open the TestCase Generator in Eclipse IDE, and how to use it.

- Open the TestCase Generator as below.

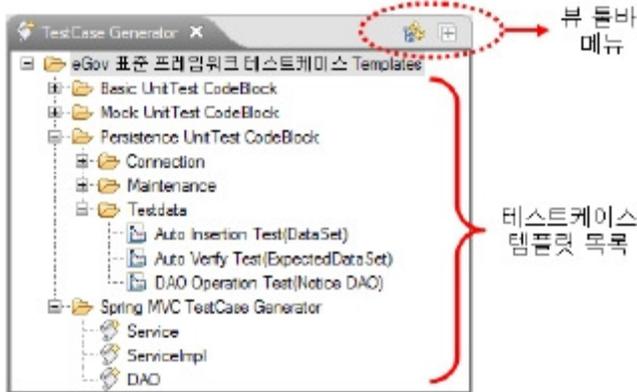
1. In Eclipse, choose Window > Show View > Other...



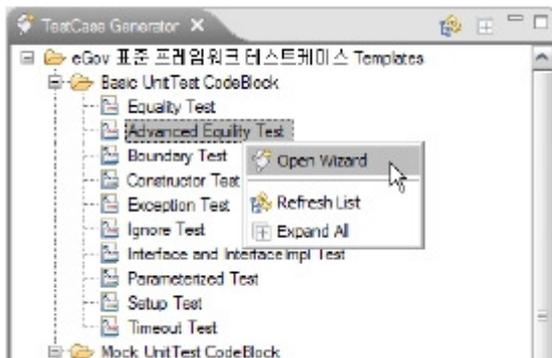
2. In Show View dialog, choose TestCase Generator under the eGovFrame category.



- TestCase Generator view comprises of the toolbar menu and the template list.



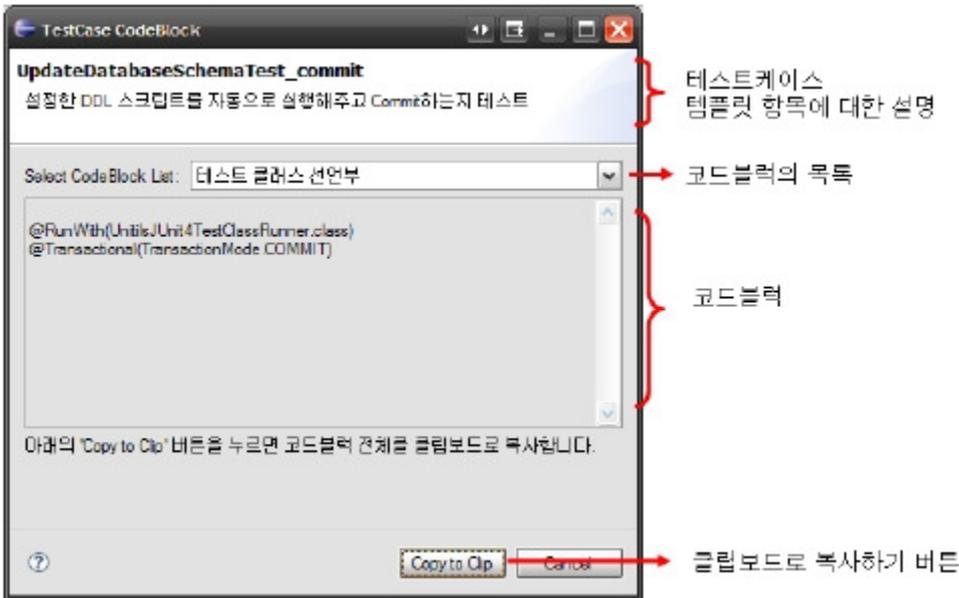
- The toolbar contains Refresh List button,  and Expand All button 
  - Refresh List: Refresh the TestCase template list by loading in the TestCase Templates plug-in from the local file system. Use this to refresh the template list up-to-date when there was a change to the plug-in or the template.
  - Expand the template list./Expand All: Expand all template lists on the test case.
- You can see all the items in an expanded view, and you can double-click for running, or right-click to bring up the context menu.



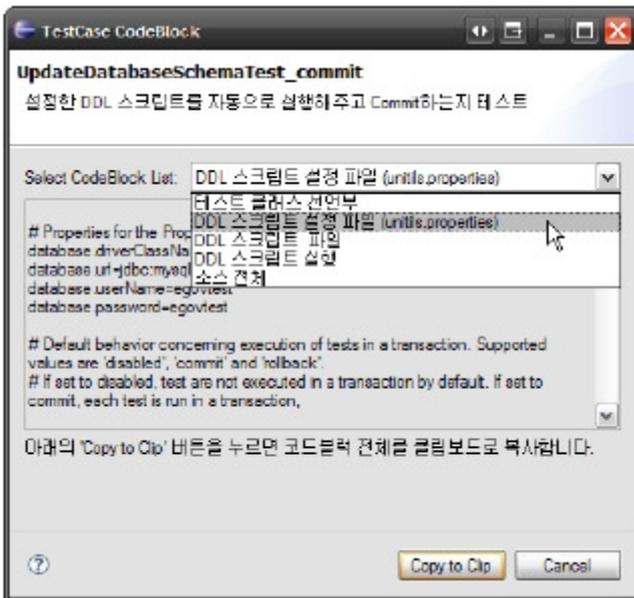
## Opening and using CodeBlocks

In the template list,  shaped icons indicate CodeBlocks, and if you double-click on them or use the context menu to to run Open Wizard, you can open the popup dialog.

The CodeBlock window is composed as following.



This dialog shows you related information on the template item you chose from the dropdown menu, and you can also see the appropriate CodeBlock in the combo box below it. When the change of CodeBlock is selected on the combo box, the Code Block selected in the center text box is indicated.



Use the Copy to Clip button to copy the CodeBlock to the clipboard, then paste into your desired location using Eclipse IDE's editor or other text editors.

## Creating TestCase template

You can double-click on the items with the -shaped icons, or use the context menu, to choose Open Wizard, to run TestCase template creating wizard.

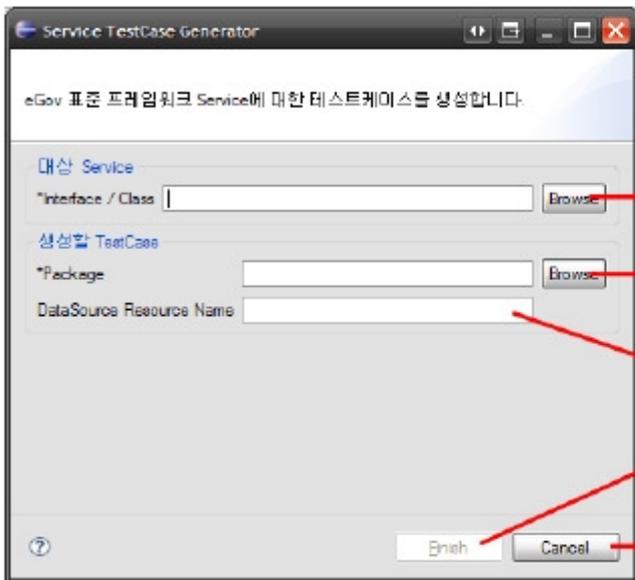
Templates are provided per Springframework's MVC tier 1)One type in each tier is provided and the list is as follows.

- TestCase template for Service interface
- TestCase template for ServiceImple class that implemented the Service interface
- DAO 2) TestCase template for DAO class

Following is the guide on how to create each template types.

### Service interface TestCase template

Service TestCase Generator wizard is as below..



Service 테스트케이스 생성에 대한 간단한 설명

테스트케이스 생성의 대상이 되는 서비스 인터페이스 선택을 위한 텍스트박스와 선택창 버튼

생성할 테스트케이스의 패키지 선택을 위한 텍스트박스와 선택창 버튼

테스트케이스에서 데이터소스를 사용할 경우 데이터소스의 이름 입력을 위한 텍스트박스 (선택 사항)

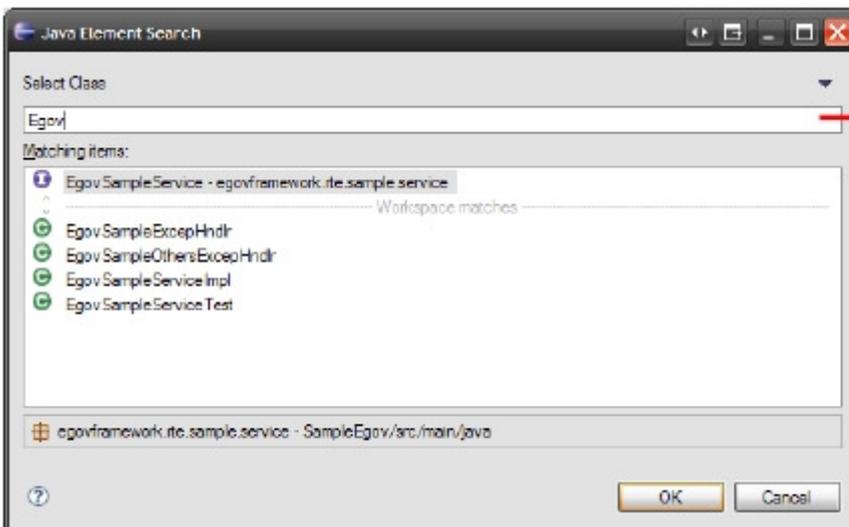
테스트케이스 생성을 위한 종료 버튼

해당 작업 취소를 위한 취소 버튼

Choose target service and test cases as below.

1. Choose target service

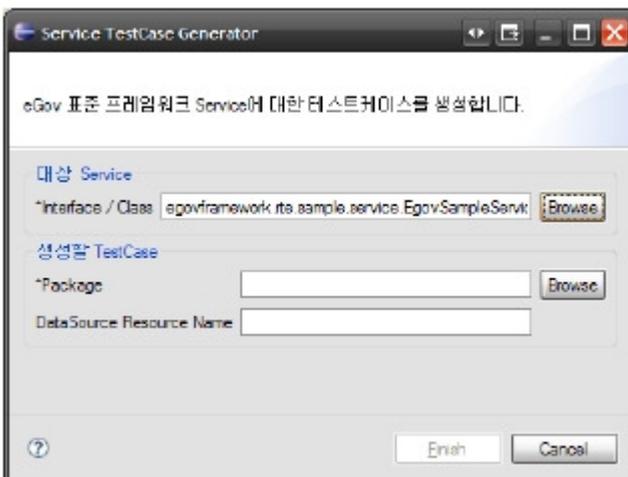
- In the Service TestCase Generator wizard, click the Browse button right of the Interface/Class text field for the desired target Service, and open the Java Element Select dialog. Type in the name of the Service in the Select Class text field.
- Below the text field, search results for Java classes and interfaces will appear.



테스트케이스 생성의 대상이 되는 인터페이스 이름을 타이핑

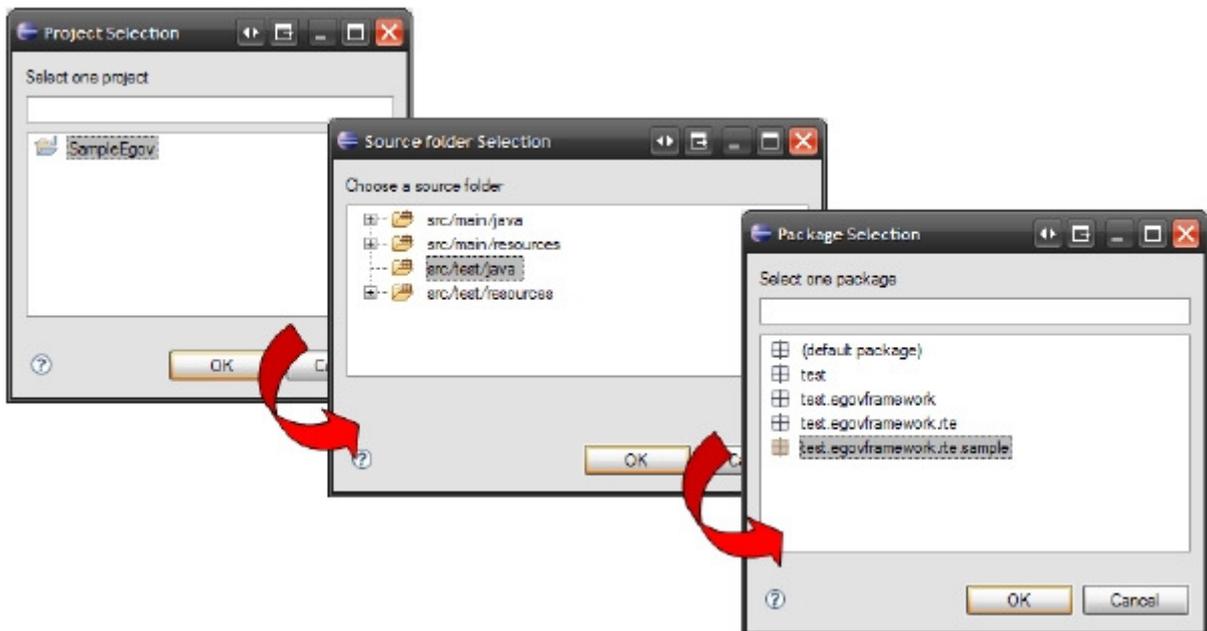
타이핑 결과에 따라 클래스와 인터페이스 들을 검색하여 조회/선택

- Choose an interface from the list, then click OK button to finish.

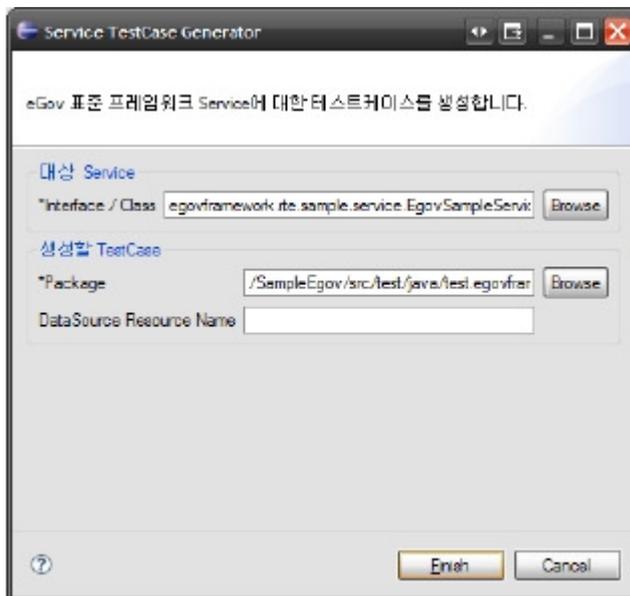


2. Choose package for TestCase Template

- In the Service TestCase Generator wizard, click the Browse button right of the TestCase package text field, and open the Project Selection dialog.
- In the dialog, choose a project, then click OK.
- In the Source folder Selection dialog, choose a directory path, then click OK.



- In the Package Selection dialog, choose a package to create template for, then click OK.



3. Choosing data source name for TestCase Template (Optional)

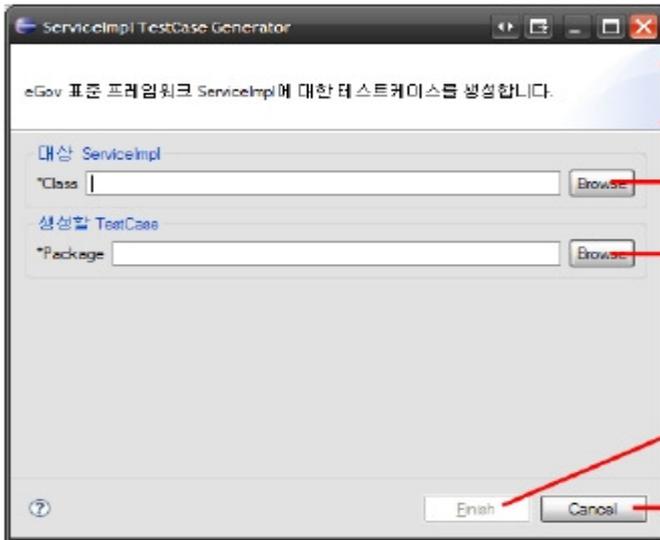
- If using a data source in the test case, enter the name of data source in the DataSource Resource Name text field.

After step 2, the Finish button will be enabled, so click the button to create the test case.

---

### ServiceImpl Class TestCase Template

ServiceImplTestCase Generator wizard is comprised as follows.



ServiceImpl 테스트케이스 생성에 대한 간단한 설명

테스트케이스 생성의 대상이 되는 서비스 클래스 선택을 위한 텍스트박스와 선택창 버튼

생성할 테스트케이스의 패키지 선택을 위한 텍스트박스와 선택창 버튼

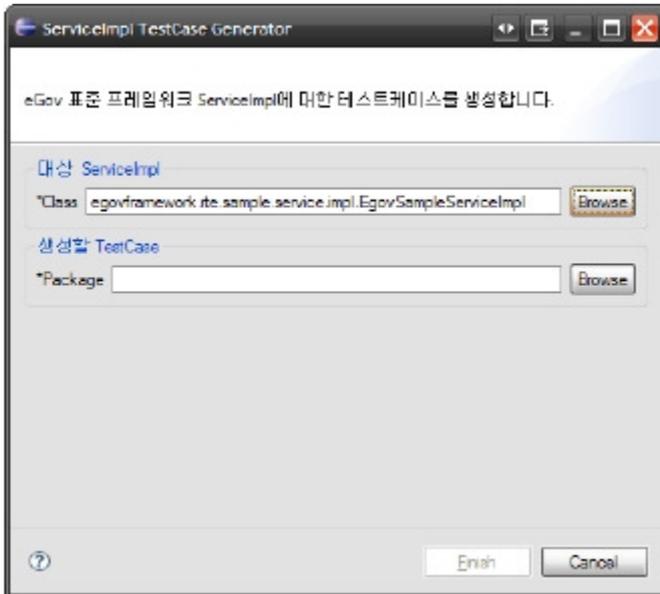
테스트케이스 생성을 위한 종료 버튼

해당 작업 취소를 위한 취소 버튼

Create test case for ServiceImpl class as below.

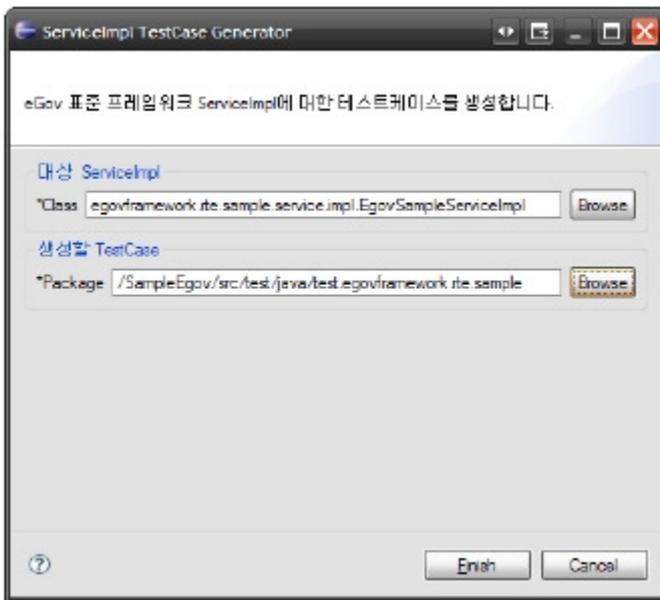
1. Choose target ServiceImpl class

- In the ServiceImplTest Case Generator wizard, click the Browse button on the right of the ServiceImpl text field, and open the Java Element Select dialog. In the dialog, type in the name of the ServiceImpl class.
- Below the text field, search results for Java classes and interfaces will appear.
- Choose the desired ServiceImpl class, then click OK to finish.



2. Choosing package for TestCase Template

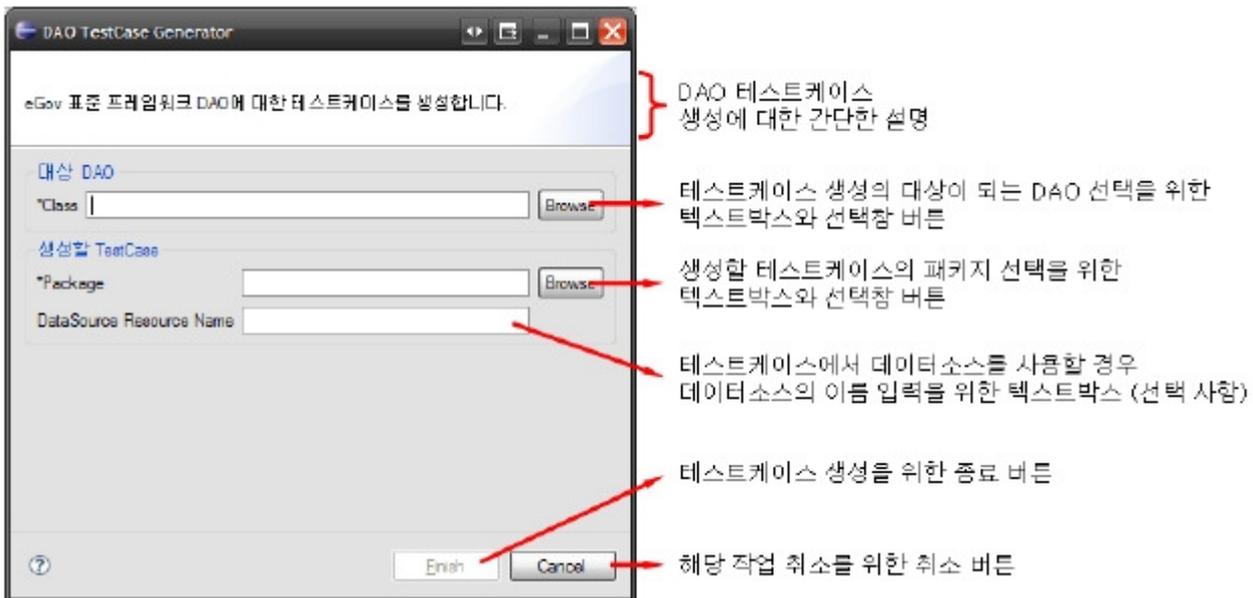
- In the ServiceImplTest Case Generator wizard, click the Browse button right of the TestCase package text field, and open the Project Selection dialog.
- In the dialog, choose a project, then click OK.
- In the Source folder Selection dialog, choose a directory path, then click OK.
- In the Package Selection dialog, choose a package, then click OK to finish..



After step 2, the Finish button will be enabled, so click the button to create the test case.

### DAO Class TestCase Template

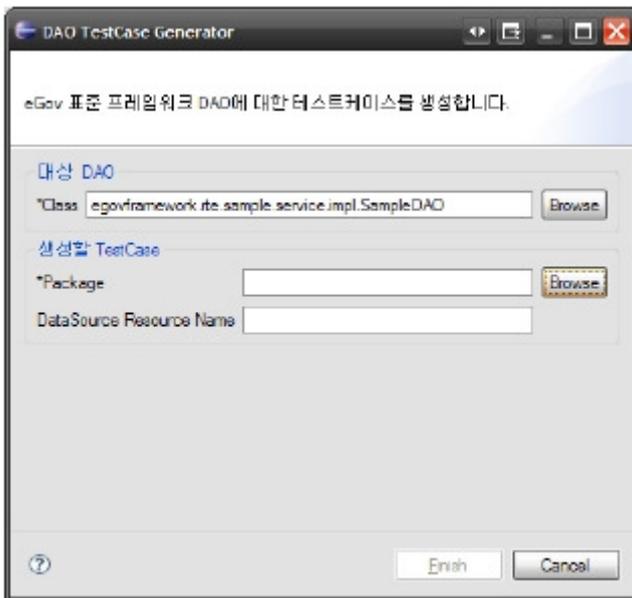
DAO TestCase Generator wizard is comprised of below.



Create DAO test case as below.

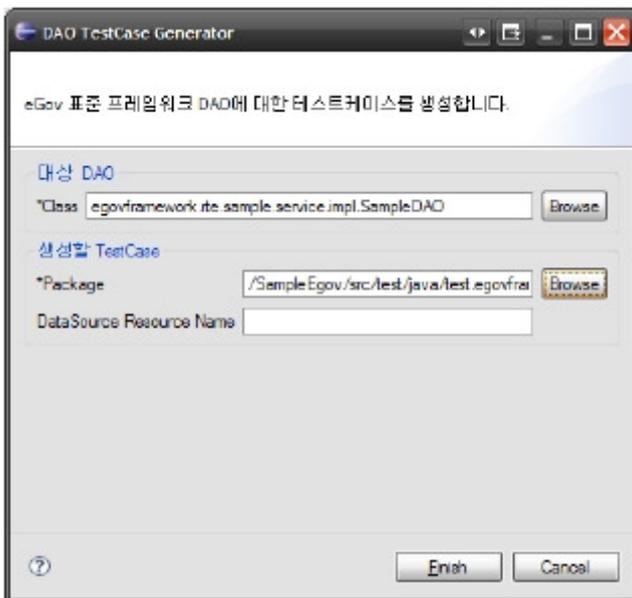
#### 1. Choose target DAO

- In the DAO TestCase Generator wizard, click the Browse button right of the TestCase package text field, and open the Java Element Select dialog.
- Type in the DAO name in the Select Class text field.
- Below the text field, search results for Java classes and interfaces will appear.
- Choose the desired ServiceImpl class, then click OK to finish.



## 2. Choosing package for TestCase Template

- In the DAO TestCase Generator wizard, click the Browse button on the right of the Package text field, and open the Project Selection window.
- In the dialog, choose a project, then click OK.
- In the Source folder Selection dialog, choose a directory for the template, then click OK.
- In the Package Selection dialog, choose a package, then click OK to finish



## 3. Choosing data source name for TestCase Template (Optional)

- If using a data source in the test case, enter the name of data source in the DataSource Resource Name text field.

After step 2, the Finish button will be enabled, so click the button to create the test case.

---

## Sample

## References

- 1) Model-View-Controller: a software architecture design pattern in software engineering, and is designed to seek clear and efficient definition of user interface and business logic.
- 2) Data Access Object: object for abstracting database interfaces.