

형상관리와 이슈 관리를 사용하는

프로젝트 개발 환경 SETUP GUIDE

Version 1.0

*본 문서는 형상관리와 이슈관리를 **개발 환경에 쉽고 빠르게 구축하는 것**을 목적으로 합니다. 따라서 상세화된 수준의 Customizing에 대해서는 설명하고 있지 않습니다.

채수원

doortts.tistory.com

문서의 목표	3
SW 형상관리(SCM)와 이슈 트래커(Issue Tracker) 툴 선정과 이유	3
SW 형상관리(Subversion) 설정	3
1. Subversion 소개.....	3
2. Subversion 서버 설치.....	3
3. 이클립스와 Subversion 서버 연결하기.....	9
4. 저장소(Repository)로 기반 소스 체크인 하기.....	13
이슈 트래커 (Trac) 설정	18
1. Trac 소개.....	18
2. Trac 설치하기.....	18
3. Trac Project Home 밑에 SVN 과 연동된 Trac 프로젝트 만들기.....	20
4. Trac 사용자 기본 환경 설정하기 (Trac Customizing).....	24
5. Trac 사용자 관리자(Admin) 페이지 설정하기.....	28
6. Trac 티켓 업무흐름도.....	37
7. Trac 활용 화면 샘플.....	38
8. Trac FAQ.....	43
[부록] Trac 추가 기능 활용예	46
1. Trac Wiki.....	46
2. Source Syntax Highlight.....	46
3. Eclipse Plugin 연동.....	47

문서의 목표

본 문서는 형상관리와 이슈관리를 프로젝트에서 사용할 수 있도록 **최대한 쉽고 빠르게 구축하는 것을 목적으로** 합니다. 따라서 대상이 되는 서버는 윈도우 환경으로 가정 하였습니다. (프로젝트 내에서만 사용할 것이기 때문이 굳이 유닉스/리눅스 서버를 고집할 필요가 없습니다.)

SW 형상관리(SCM)와 이슈 트래커(Issue Tracker) 툴 선정과 이유

SW형상관리

Subversion - CVS의 개량 버전이며 사실상의 업계표준(De facto Standard).

비교 대상 : CVS

이슈 트래커

Trac - 무료.심플한 인터페이스와 사용하기 쉬운 환경 제공. Subversion 과의 통합. Wiki 문서 작성 제공.

비교대상: Mantis, Bugzilla, JIRA(상용이라 고려시점 초반에 제외됨)

SW 형상관리(Subversion) 설정

1. Subversion 소개

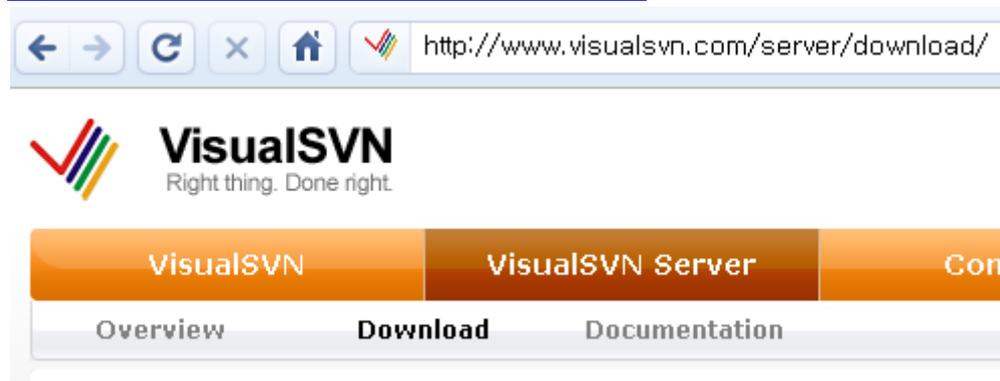
SW개발에서 중요자산인 소스(Source)의 변경과 이력을 관리하는 것을 SW형상관리 (Software Configuration Management, SCM) 라고 부른다. Subversion은 SCM을 지원해 주는 Tool의 일종으로 CVS 를 개선/대체하기 위해 만들어진 오픈소스 제품으로, SCM 분야에서 사실상의 업계 표준 툴이 되었다.

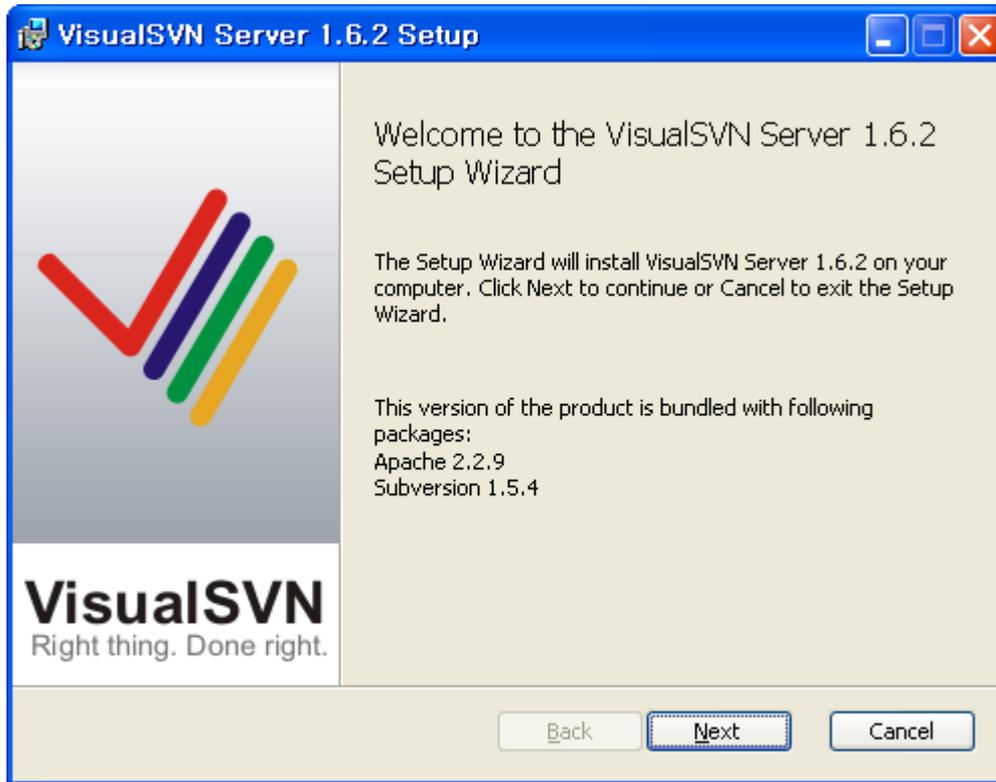
프로젝트에서 소스 관리를 위해 CVS를 사용할 것인가 아니면 Subversion을 쓸 것인가 라는 질문은 무의미 하며, 특수한 경우를 제외하고는 Subversion을 쓴다고 생각하면 되겠다.

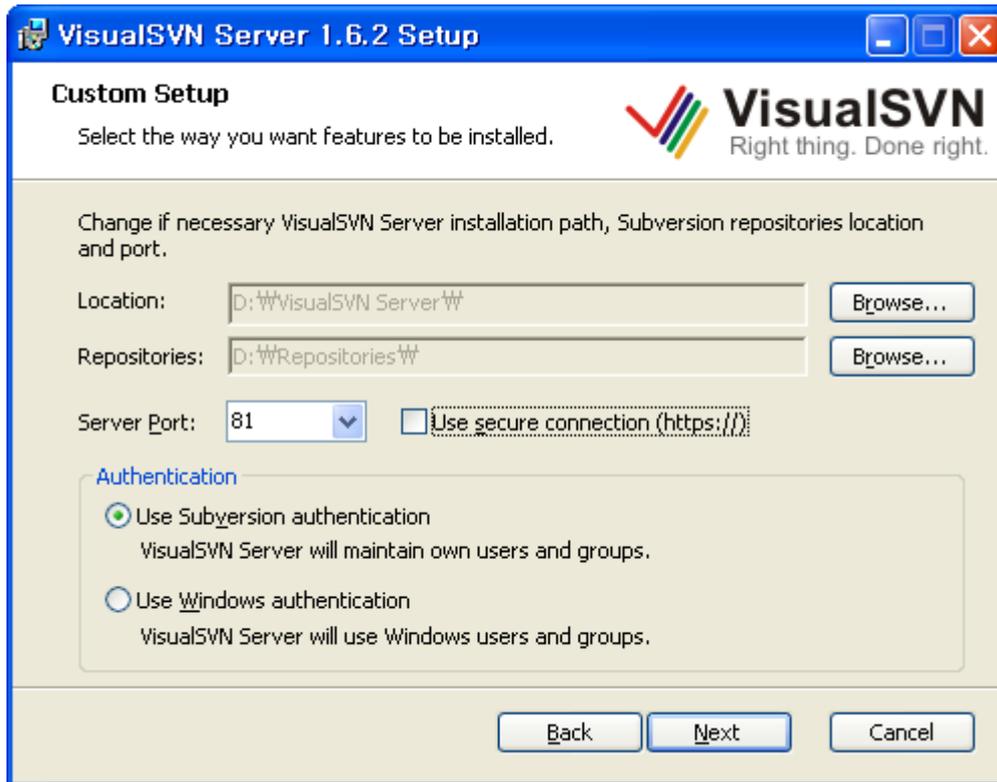
2. Subversion 서버 설치

A. VisualSVN 내려받기

<http://www.visualsvn.com/visualsvn/server/download>

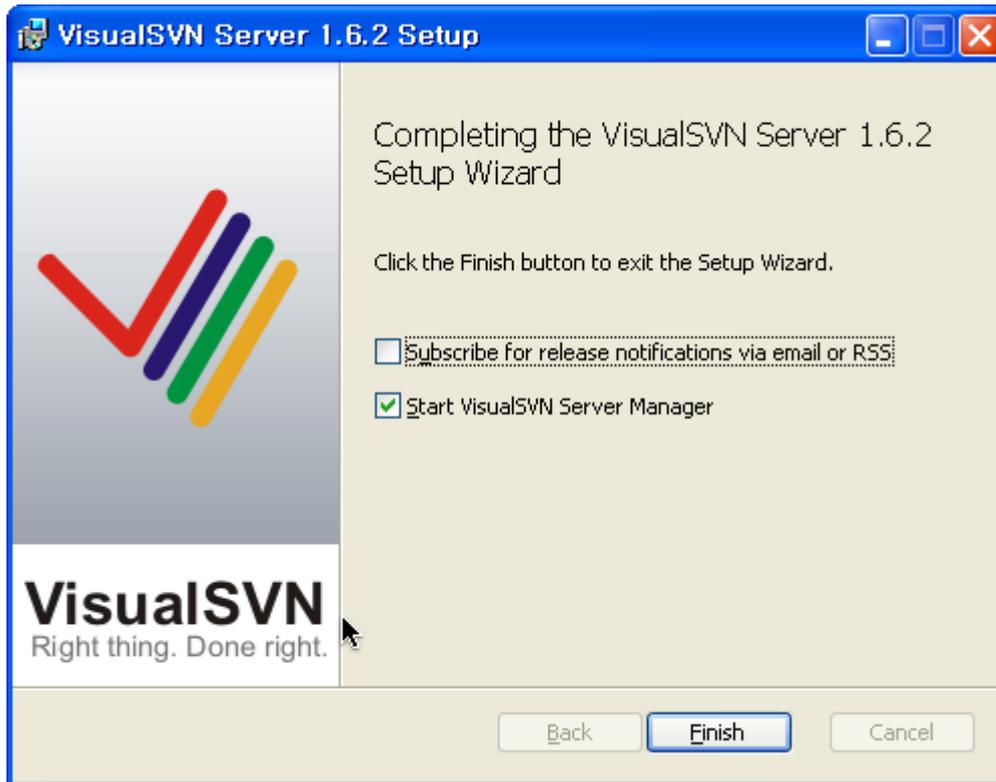


B. 원하는 폴더로 설치(install) – 본 문서에서는 d:\visual



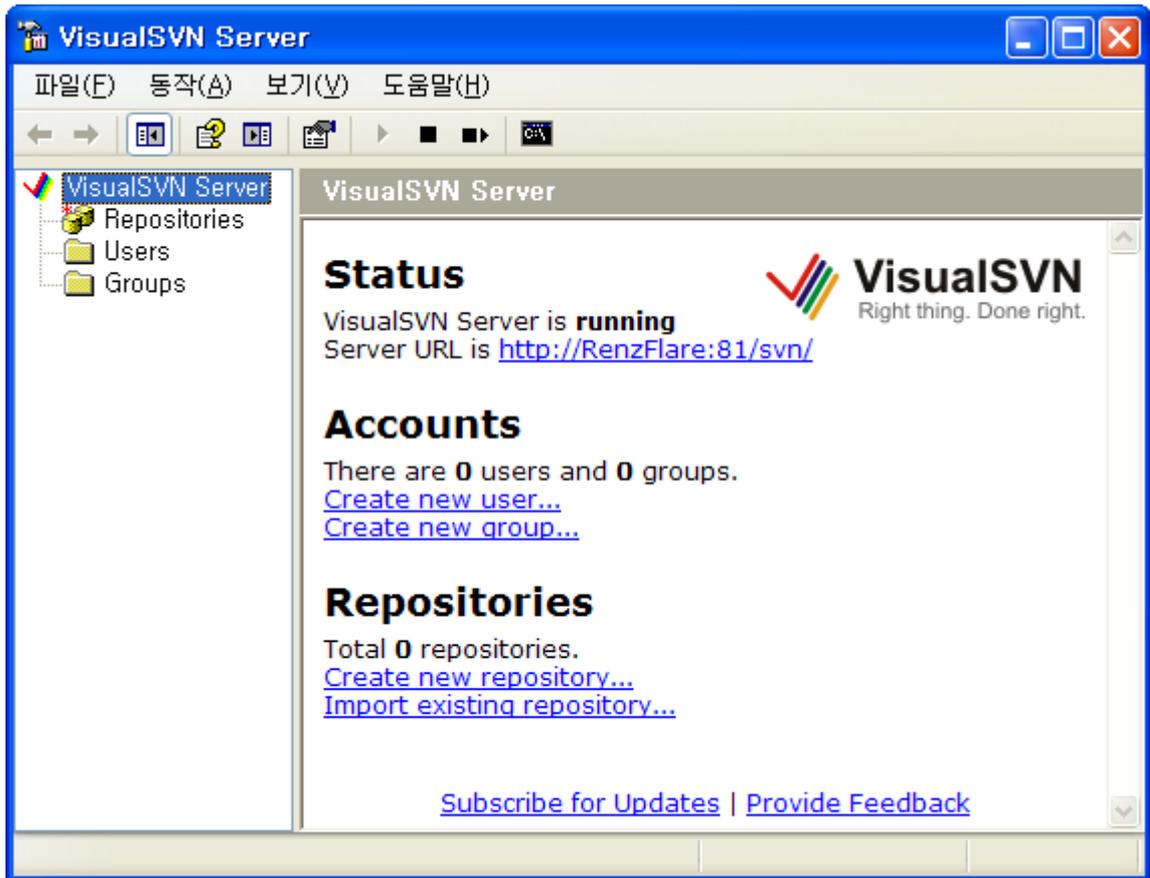
원하는 경로로 **VisualSVN Server** 와 **Repository** 경로 설정. **https** 사용여부. 인증방식, **server Port** 를 정한다.

항목	본 예제에서의 설정값
VisualSVN Server	D:\VisualSVN Server
Repository 경로	D:\Repositories
https 사용여부 결정	Uncheck
Server Port	81
Authentication	Use Subversion...

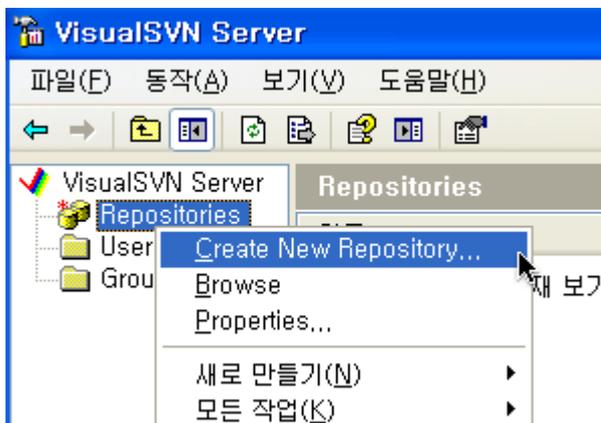


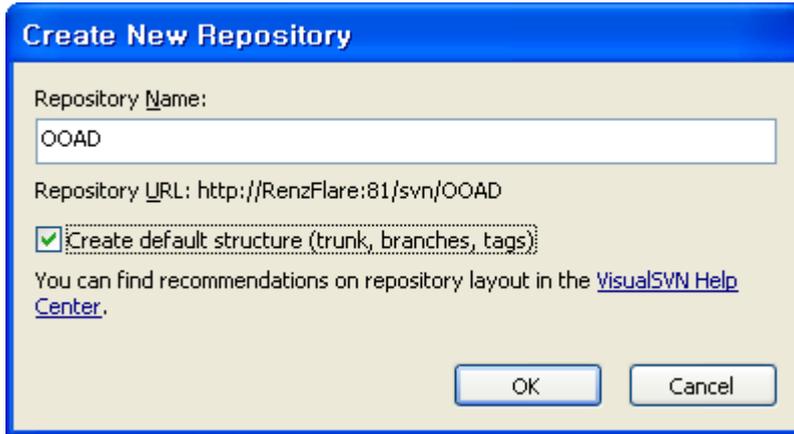
정상 완료 화면

C. 기동된 서버 매니저 확인



D. 저장소(Repository) 생성



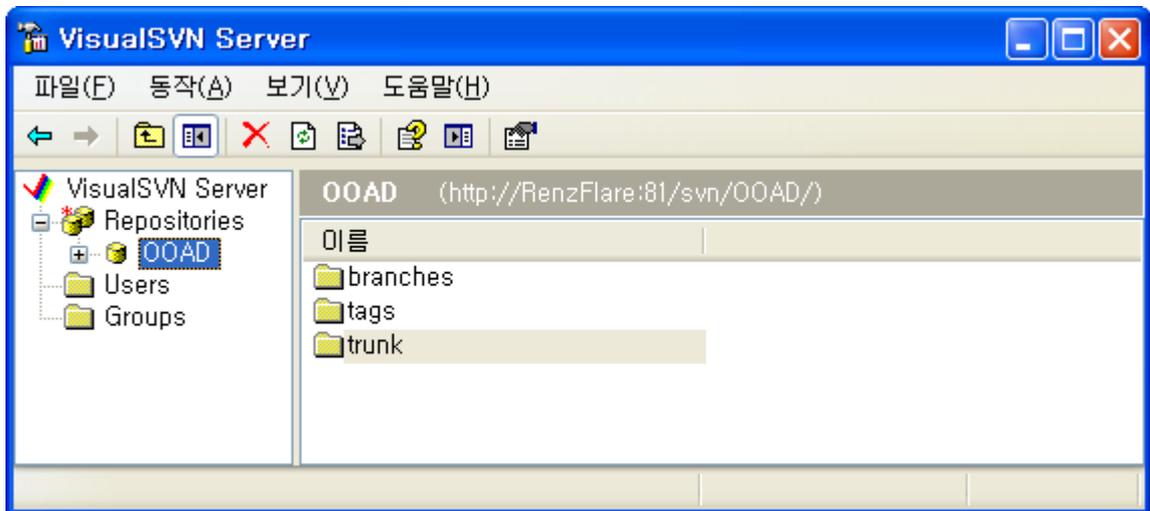


드물게 경우에 따라서는 trunk, branches, tags 를 만들지 않을 수 있다. 하지만, 가급적 만들어서 각 폴더의 의미를 간략하게 알려주는 것이 좋다.

trunk: Main Stream Source

branches: 특정한 이슈에 의해 갈라져 나온 버전. 가급적 빠른 시일 내에 trunk 와 병합필요

tags: 소스의 특정 시점을 긋고자 할 때 사용함 (ex. OpenBeta1, 1.0 Stable Release 등등)



E. 필요한 유저 계정 생성



<당연한 이야기이지만, 경우에 따라 그룹을 생성해서 그룹으로 관리할 수도 있다>

3. 이클립스와 Subversion 서버 연결하기

A. Eclipse 용 Subversion plugin (subclipse) 설치

<http://subclipse.tigris.org/> 접속하여 eclipse update site 주소 확인



Eclipse 3.2/Callisto, 3.3/Europa, 3.4/Ganymede +

Subclipse 1.4.7 is now available for Eclipse 3.2+!

See the [changelog](#) for details. Existing Subclipse users should read the [u](#) changes you to need to make to your Eclipse preferences to see the new

Support for Subclipse 1.2.x is now deprecated as the 1.4.x releases supp requires Subversion 1.5.0 version of JavaHL/SVNKit

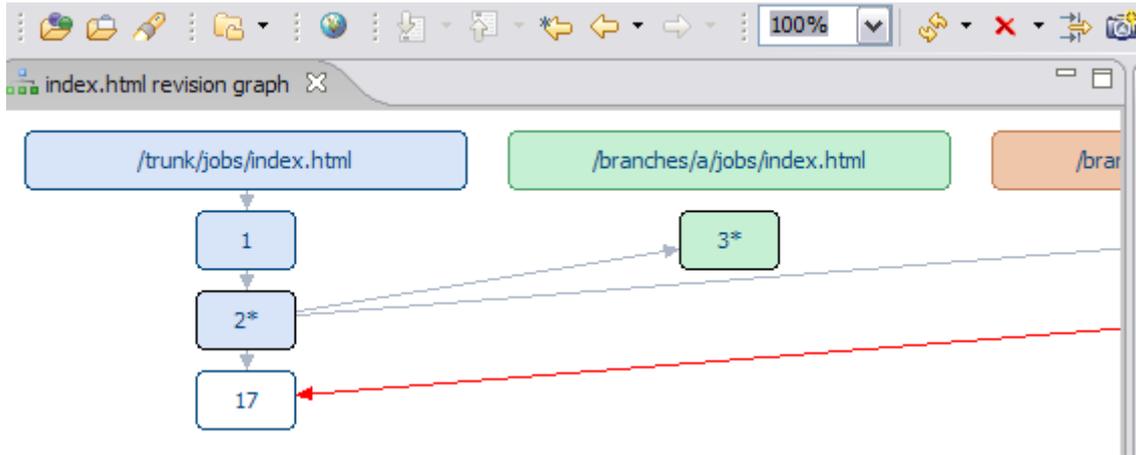
Links for 1.4.x Release:

Changelog: http://subclipse.tigris.org/subclipse_1.4.x/changes.html

Eclipse update site URL: http://subclipse.tigris.org/update_1.4.x

Zipped downloads: <http://subclipse.tigris.org/servlets/ProjectDocumentLi>

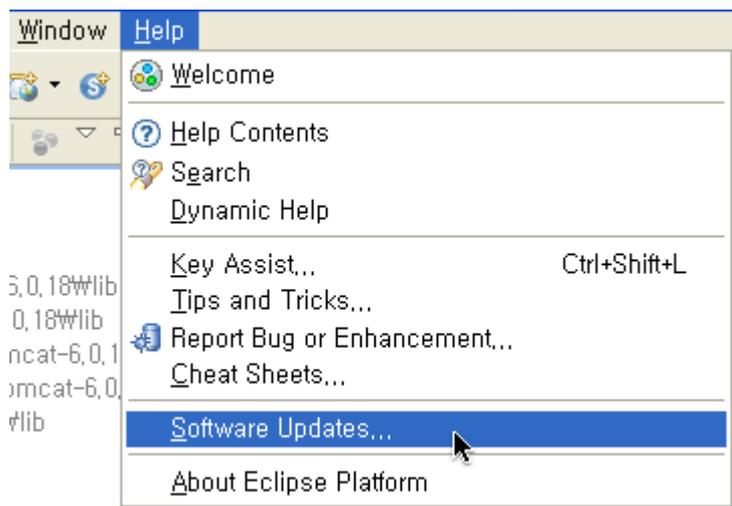
이클립스 플러그인 사이트 주소 http://subclipse.tigris.org/update_1.4.x



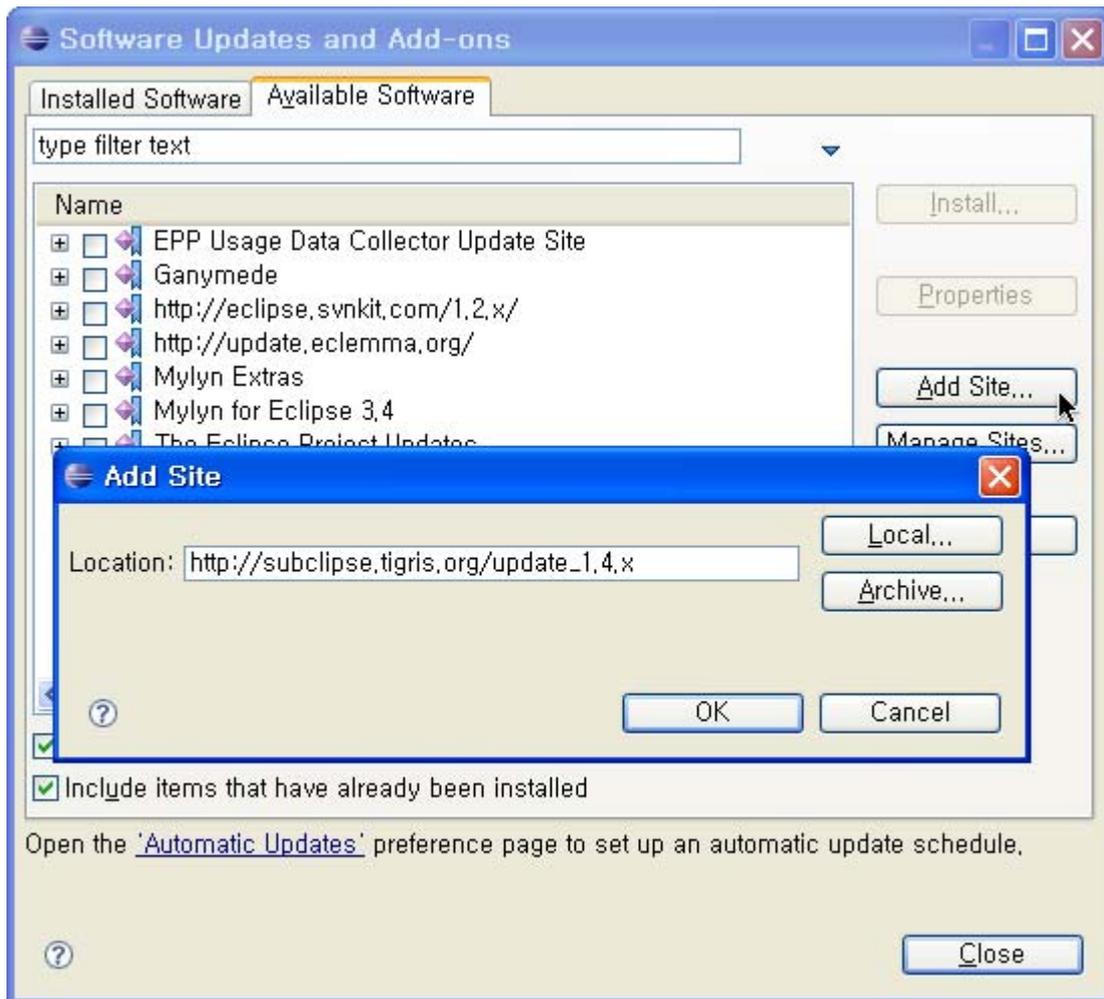
<Subclipse 선정이유 중 하나인 revision graph 기능>

B. 이클립스 기동

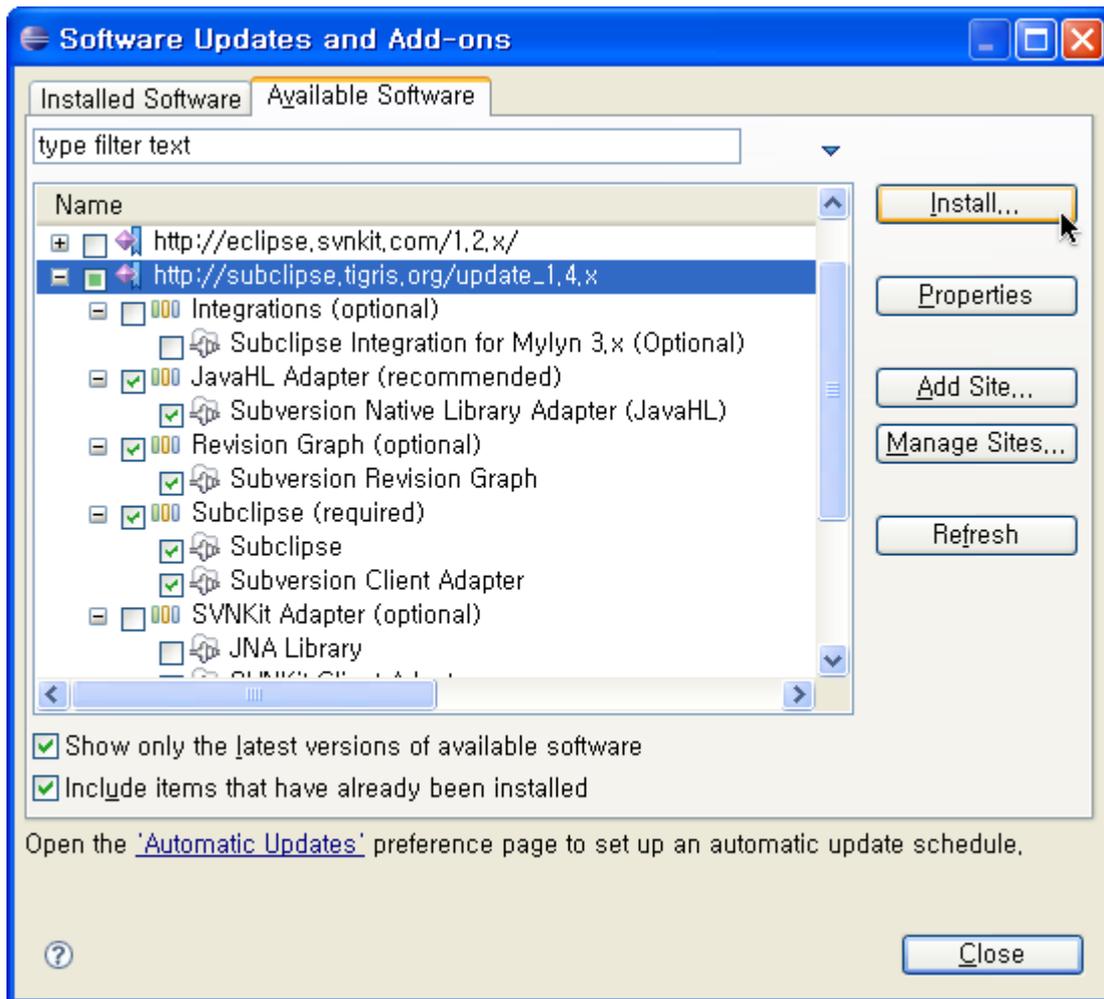
C. 메뉴에서 Help → Software Updates... 선택



D. Available Software 탭을 누른 후 Add Site 선택 → location 에 위에서 찾은 subclipse 의 update site 주소를 입력한다.

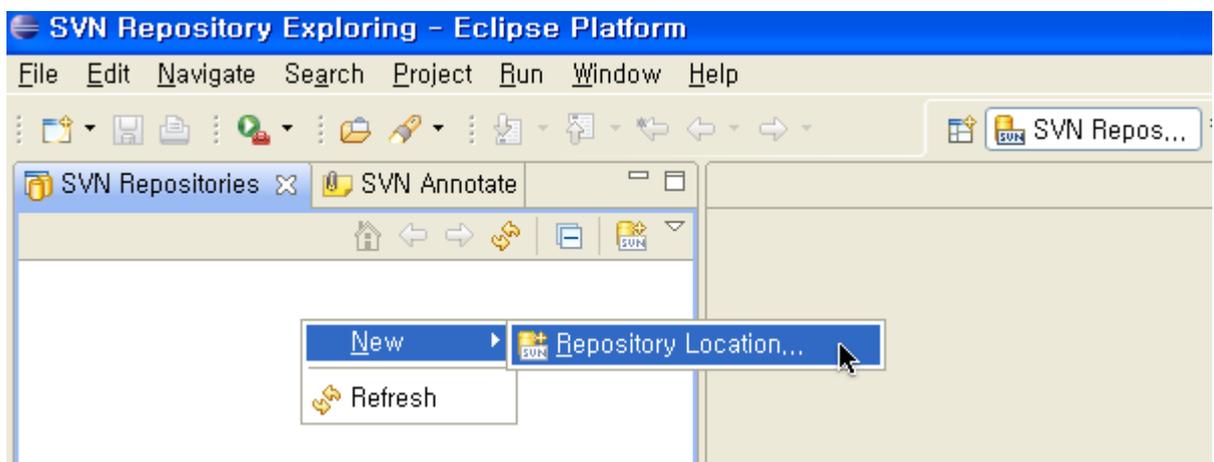


E. 인스톨 할 항목을 선택한 다음 install 을 눌러 끝까지 진행한다.

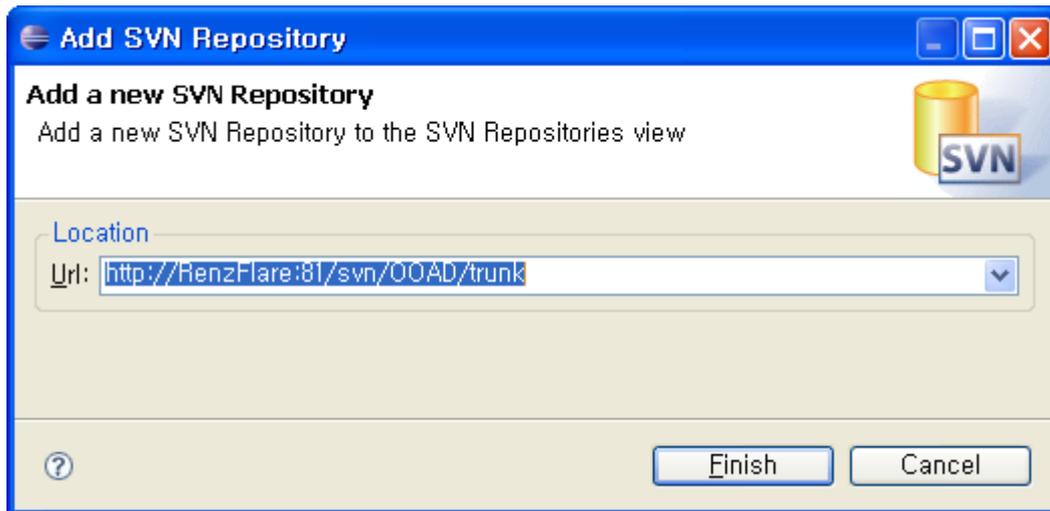


<설치 항목 선택 화면. 참고로 SVNKit Adapter 는 Subversion 자체를 제어하기 위한 라이브러리이다>

- F. 이클립스의 SVN Repository Perspective 를 선택한다.
- G. SVN Repositories View 에서 마우스 오른 버튼 클릭 → New → Repository Location 선택



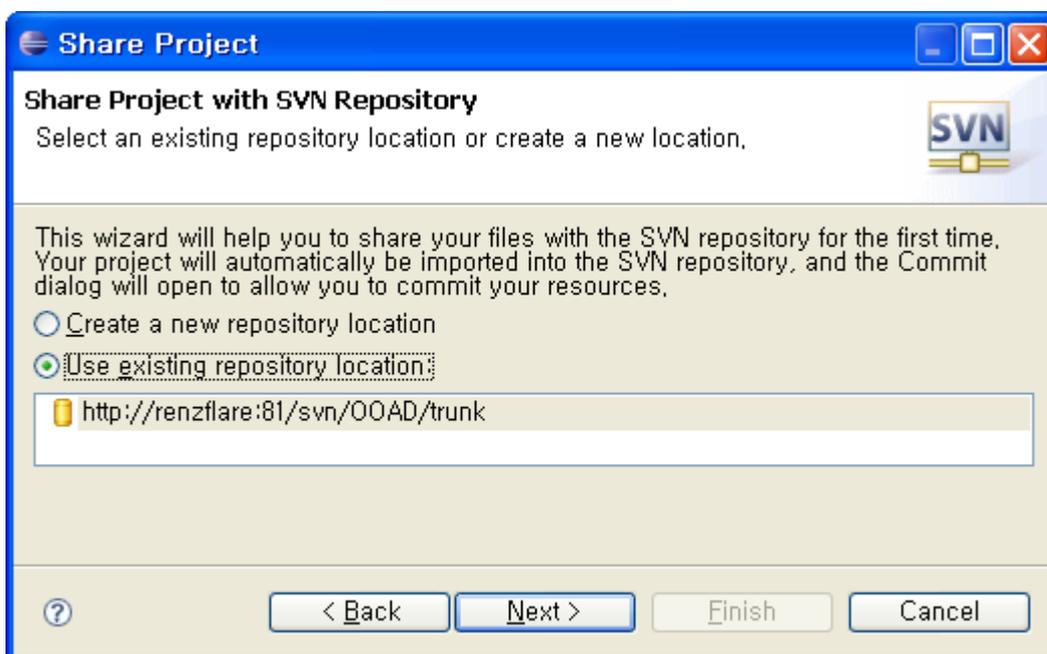
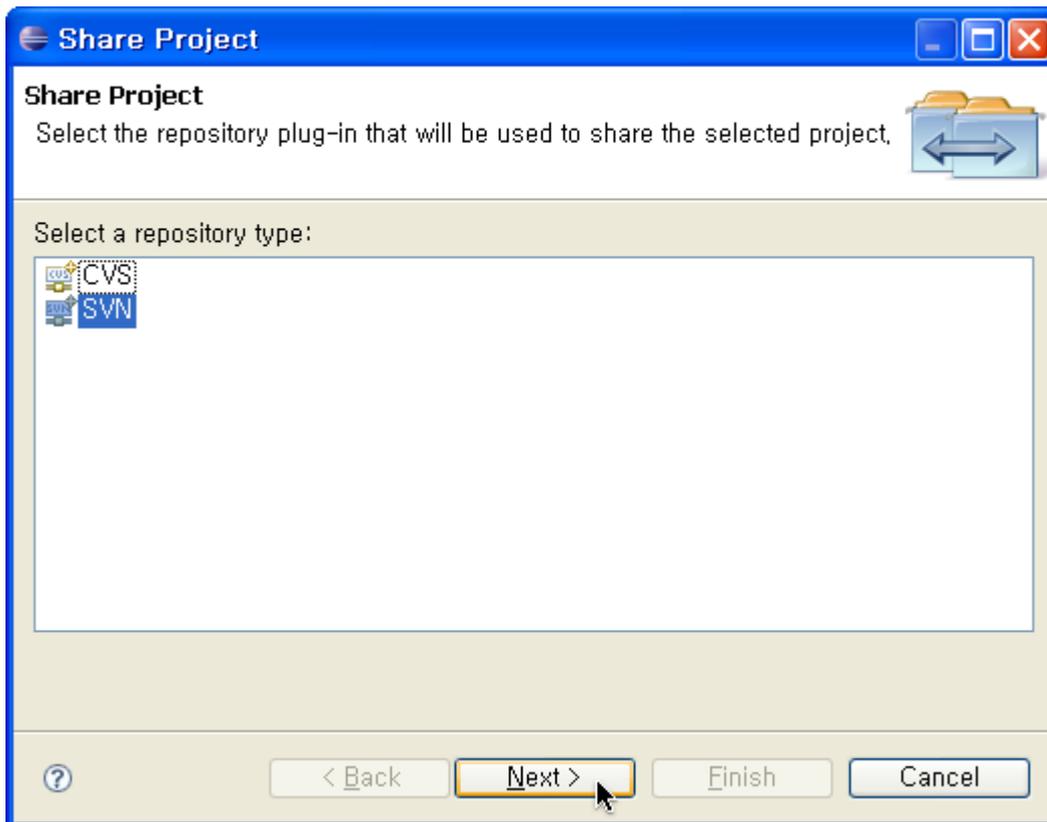
- H. Subversion 서버 저장소 URL 을 입력한다. (trunk 를 붙여준다)

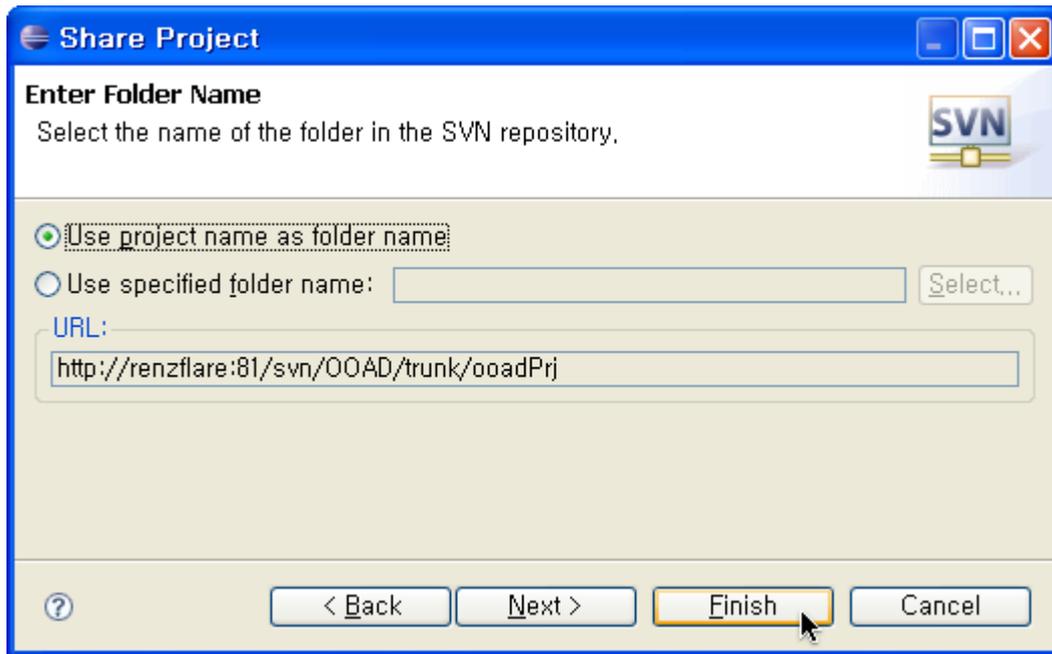


4. 저장소(Repository)로 기반 소스 체크인 하기

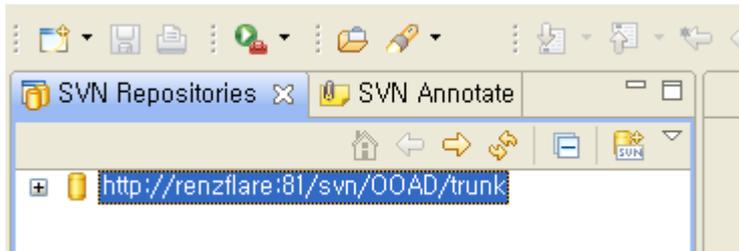
A. 프로젝트를 선택한 다음 마우스 오른 버튼 → Team → Share Project 선택





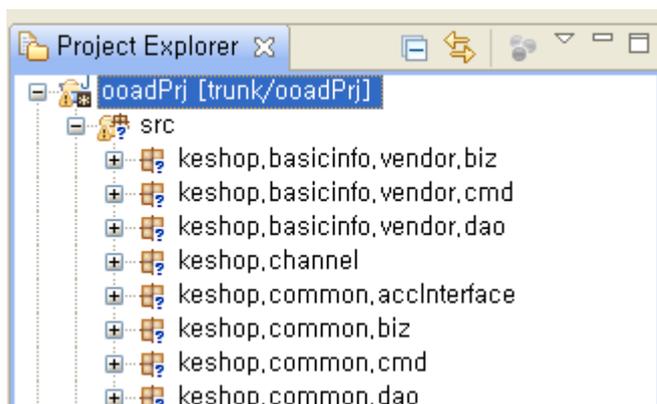


특별한 일이 없으면 프로젝트 이름을 그대로 폴더 이름으로 사용한다.



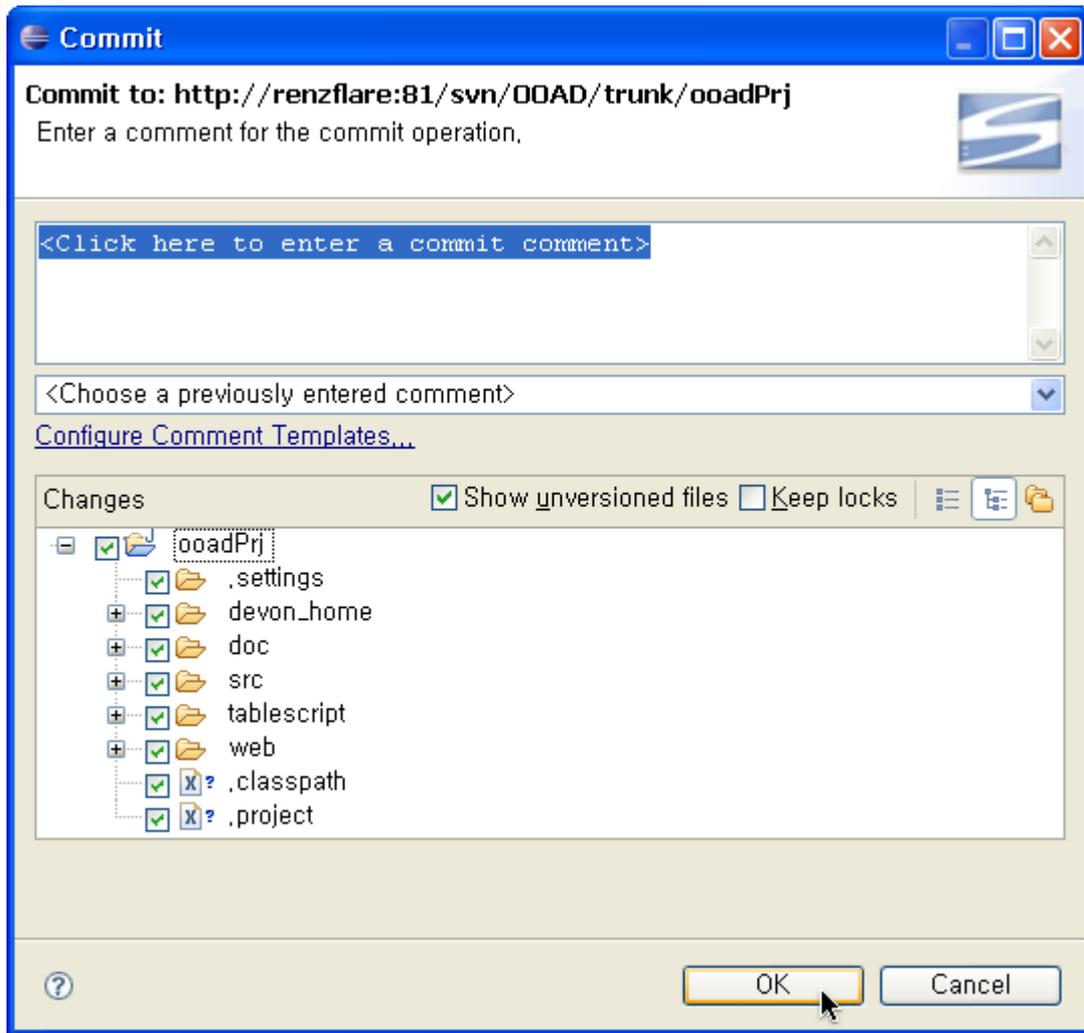
SVN repository의 위치가 등록된 화면

B. 개발 소스 최초 check-in 작업

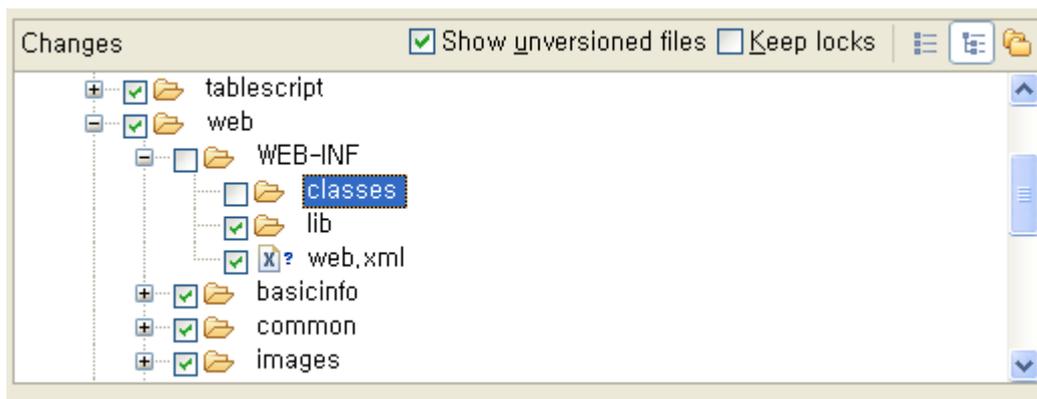


<현재는 소스가 under control 상태가 아니라서 ? 표시가 붙는다>

Java Perspective 에서 프로젝트를 선택해서 마우스 오른 버튼 → Team → commit

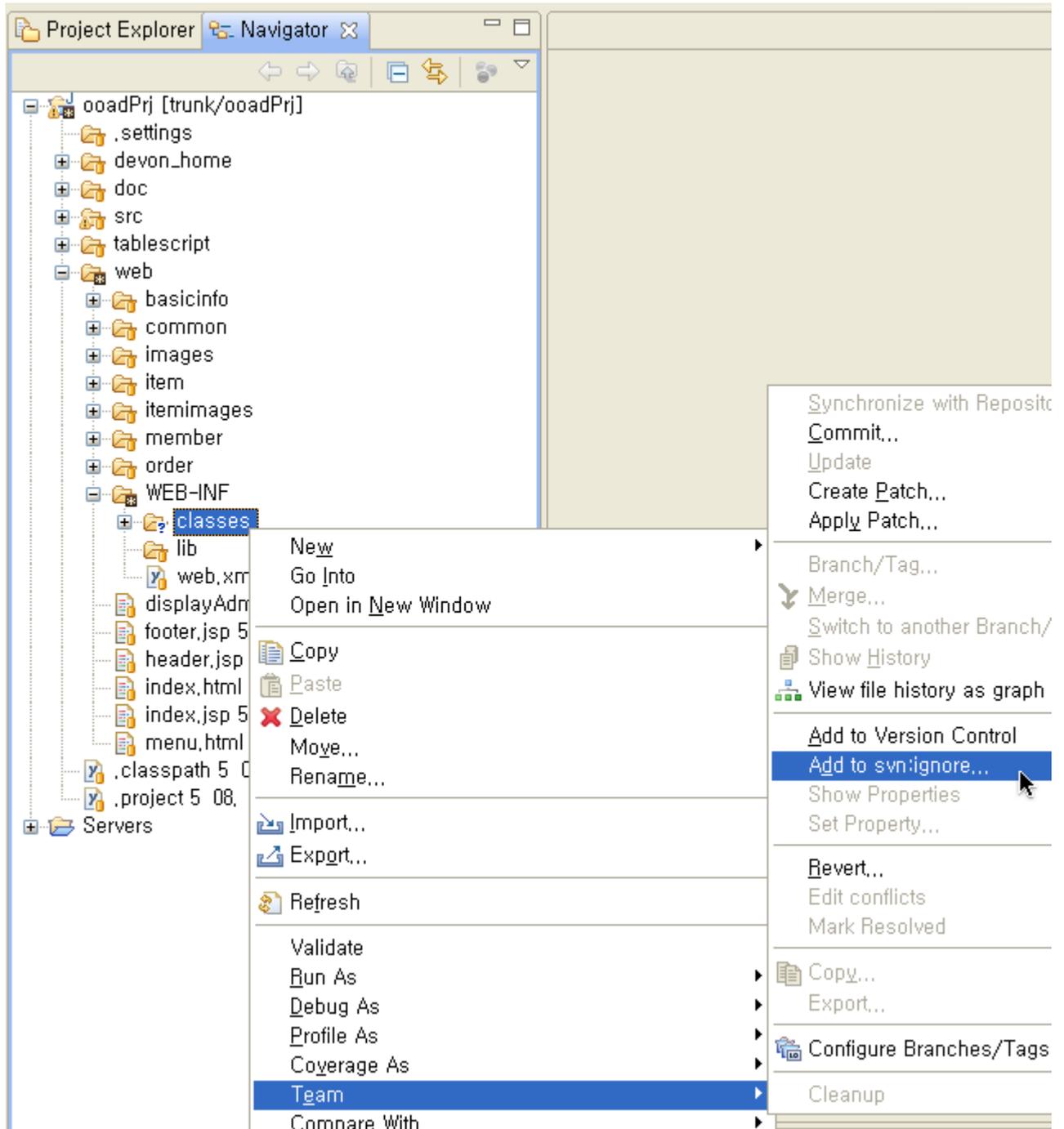


소스 컨트롤로 관리할 파일을 선택한다.

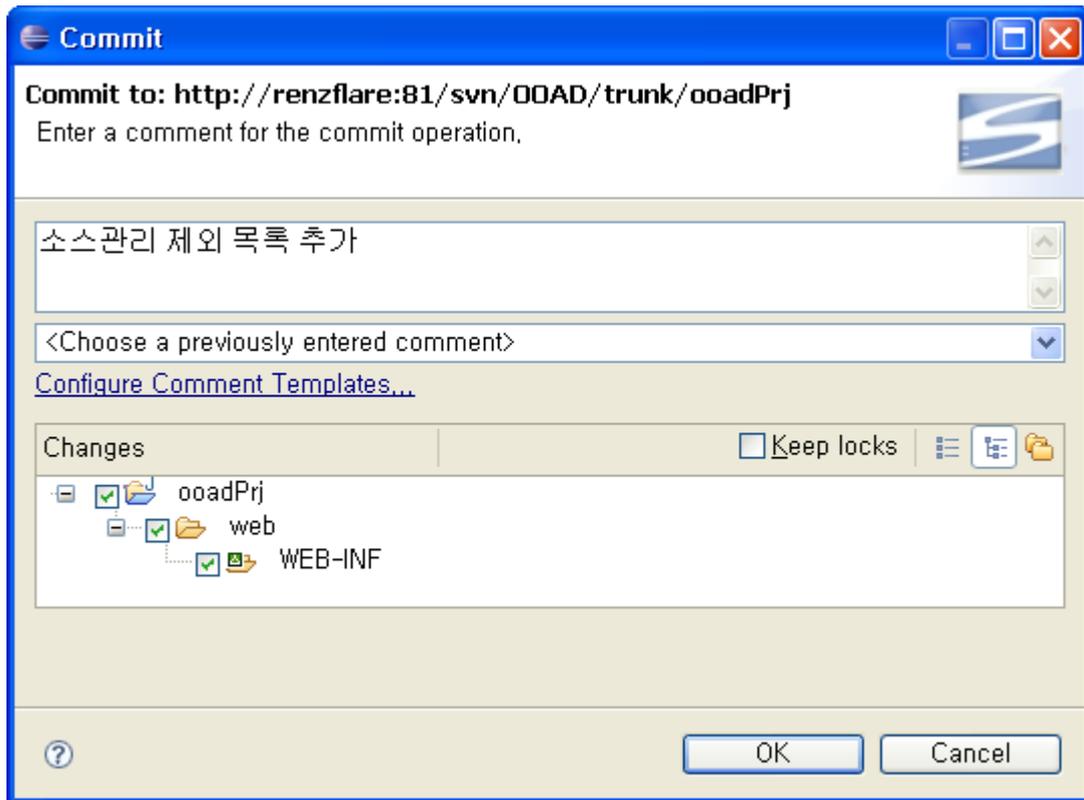


예를 들면 웹 컨테이너가 클래스 파일들을 생성할 폴더 같은 경우에는 관리에서 제외한다

C. 제외 파일이나 폴더를 Subversion 으로 관리하지 않게 만들기.



Navigator View 를 선택해서 여전히 '?'로 남아있는 폴더나 파일을 선택해서 Team → Add to svn:ignore 를 선택해서 commit (check-in) 시에 매번 확인하지 않도록 만듭니다.



다시 commit 을 수행해서 관리하지 않을 콘텐츠들을 저장한다.

이슈 트래커 (Trac) 설정

1. Trac 소개

Trac은 이슈의 관리와 할당/추적을 지원하는 오픈소스 제품이다. 기존의 툴들(버그질라, 맨티스)이 버그 추적 (Bug Tracking)에 중점을 두었다면, Trac 은 업무(task or issue)와 버그를 통합한 형태로 관리한다. 한가지 독특한 점은 trac에서는 버그나 이슈를 티켓(Ticket)이라는 형태로 통칭해서 부르고 있다는 점이다. 한글에서 동음 반복문이 좋은 표현이 아니라는 것처럼 issue the issue 라는 표현을 재치있게 issue the ticket(티켓을 발행)이라는 형태로 표현하였다.

2. Trac 설치하기

A. 목표 환경 설명

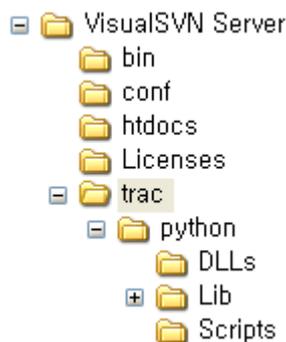
Trac Install Home	D:\VisualSVN Server\trac
OOAD 프로젝트용 Trac Project Home	D:\Trac\OOAD

B. VisualSVN 용 Trac 을 다운로드 한다.

<http://www.visualsvn.com/server/trac/>

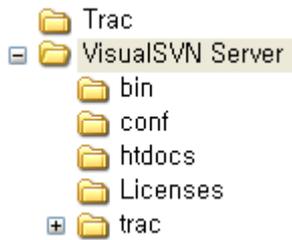


C. 다운로드 받은 압축 파일을 VisualSVN 이 설치된 폴더 아래에 푼다.



앞으로는 D:\VisualSVN Server\trac 을 Trac Install Home 이라고 칭한다.

D. Trac Project Home 폴더를 Trac 이라는 이름으로 원하는 위치에 만든다. (본 예제에서는 D:\Trac 으로 만들었다고 가정한다.)



B와 C 작업을 한 후의 모습

3. Trac Project Home 밑에 SVN 과 연동된 Trac 프로젝트 만들기

A. 트랙 인스톨 홈 폴더(D:\VisualSVN Server\trac) 에서 trac-admin 을 이용해 프로젝트를 생성한다.

i. trac-admin 커맨드를 이용하여 프로젝트 생성

```
D:\VisualSVN Server\trac>dir
D 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: D896-8DD1

D:\VisualSVN Server\trac 디렉터리

2008-12-22 오후 04:44 <DIR>          .
2008-12-22 오후 04:44 <DIR>          ..
2008-12-22 오후 04:45 <DIR>          python
2008-08-02 오후 06:03          148 trac-admin.bat
                1개 파일          148 바이트
                3개 디렉터리 137,929,154,560 바이트 남음

D:\VisualSVN Server\trac>trac-admin D:\Trac\OOAD initenv
```

문법> trac-admin Trac_Project_Home\프로젝트이름 initenv

예제> trac-admin D:\Trac\OOAD initenv

ii. 프로젝트 이름 설정

```
Please enter the name of your project.
This name will be used in page titles and descriptions.

Project Name [My Project]> 분석/설계 실무자 프로젝트
```

iii. Db 는 내장 DB 를 사용한다.

```
Please specify the connection string for the database to use.
By default, a local SQLite database is created in the environment
directory. It is also possible to use an already existing
PostgreSQL database (check the Trac documentation for the exact
connection string syntax).
```

```
Database connection string [sqlite:db/trac.db]> 
```

그냥 Enter

- iv. 버전 컨트롤 시스템은 기본값인 SVN (=Subversion) 을 이용한다.

```
Please specify the type of version control system,
By default, it will be svn.

If you don't want to use Trac with version control integration,
choose the default here and don't specify a repository directory.
in the next question.
```

```
Repository type [svn]>
```

마찬가지로 그냥 Enter

- v. Subversion 의 프로젝트 소스 저장소(Repository) 위치를 지정한다.

```
Please specify the absolute path to the version control
repository, or leave it blank to use Trac without a repository.
You can also set the repository location later.
```

```
Path to repository [/path/to/repos]> D:\Repositories\OOAD
```

VisualSVN Server 에서 정한 폴더 위치를 적는다.

- vi. 정상적인 종료 메시지 확인

```

-----
Project environment for '분석/설계 실무자 프로젝트' created.

You may now configure the environment by editing the file:

    D:\Trac\OOAD\conf\trac.ini

If you'd like to take this new project environment for a test drive,
try running the Trac standalone web server `tracd`:

    tracd --port 8000 D:\Trac\OOAD

Then point your browser to http://localhost:8000/OOAD.
There you can also browse the documentation for your installed
version of Trac, including information on further setup (such as
deploying Trac to a real web server).

The latest documentation can also always be found on the project
website:

    http://trac.edgewall.org/

Congratulations!

```

과연 마냥 축하할 일인가...??

B. VisualSVN Server 내장 아파치 서버에서 Trac URI 인식하게 만들기

- i. D:\VisualSVN Server\httpd-wrapper.bat 파일을 열어서 맨 윗줄에 "set PYTHONHOME=%~dp0\Trac\python" 을 입력한다.

```
@echo off
```

```
rem We have to use this wrapper to set path to our own version of Subversion
set PYTHONHOME=%~dp0\Trac\python
```

```
set PATH=%~dp0\bin;%PATH%
"%~dp0\bin\VisualSVNServer.exe" %*
exit %ERRORLEVEL%
```

- ii. D:\VisualSVN Server\conf 아래에 있는 httpd-custom.conf 파일을 열어 아래 내용을 추가한다.

```

LoadModule python_module "trac/python/mod_python.so.pyd"
LoadModule authz_user_module bin/mod_authz_user.so
<Location /trac>
    SetHandler mod_python
    PythonInterpreter main_interpreter
    PythonHandler trac.web.modpython_frontend
    PythonOption TracEnvParentDir D:\Trac

```

```

PythonOption TracUriRoot /trac
AuthName "Trac"
AuthType Basic
AuthBasicProvider file
AuthUserFile "D:/Repositories/htpasswd"
Require valid-user
</Location>

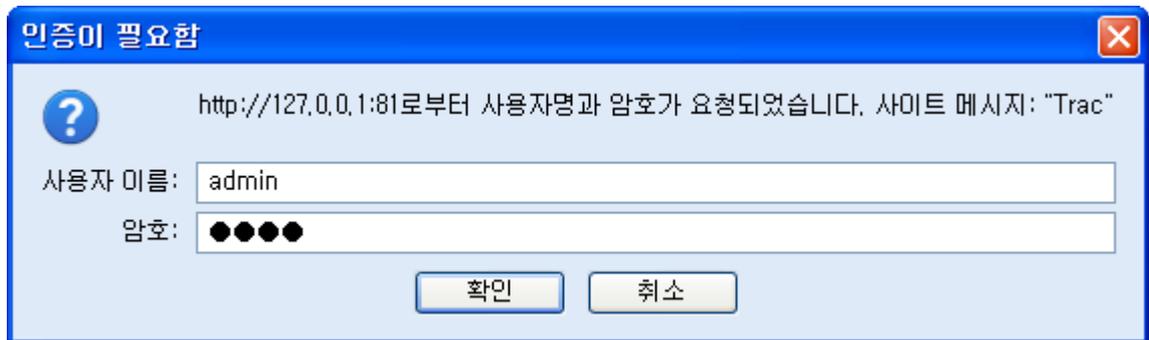
```

굵게 표시된 부분을 유의해서 경로를 맞춰준다. 단, **VisualSVN 서버 인스톨 시점에 Subversion Authentication** 을 선택하지 않았을 경우는 좀 다르다. 이 때에는

<http://www.visualsvn.com/server/trac/> 여기를 참조한다.

- iii. **VisualSVN Server** 를 사용해서 **Subversion Server** 를 재기동한 후 웹 브라우저로 접속해 본다.

URL 예> <http://127.0.0.1:81/trac/>



위에서 만든 계정과 password 를 입력한다.



Available Projects

- [분석/설계 실무자 프로젝트](#)

원한다면 본 가이드의 3.A 단계를 여러 번 실행해서 복수개의 프로젝트를 구성하는 것도 가능하다. 본, 가이드에서는 1 개만 사용한다.



(please configure the [header_logo] section in trac.ini)



Welcome to Trac 0.11

Trac is a **minimalistic** approach to **web-based** management of **so** enhancements and overall progress.

All aspects of Trac have been designed with the single goal to **help** possible on a team's established process and culture.

환영 메시지가 나오면 성공!

4. Trac 사용자 기본 환경 설정하기 (Trac Customizing)

본 설정에서는 Trac에서 자주 쓰이는 기본 설정위주로만 설명한다. 기타, 자세한 내용은 Trac 메뉴의 help/guide 항목을 참조하기 바란다.

A. 프로젝트 로고 설정하기

- i. 이미지 파일을 만들어서 Trac Project Home 에 해당하는 D:\Trac\OOAD 의 하위 디렉터리인 htdocs 폴더에 넣어 준 다음 D:\Trac\OOAD\conf 폴더의 trac.ini 파일에서 header_logo 항목을 수정한다.

```
[header_logo]
alt = 분석/설계 실무자 프로젝트
height = -1
link = /trac/
src = site/header_logo.png
width = -1
```

본 예제에서는 header_logo.png 라는 파일을 사용하였다.

참고로, 위 설정에서 src 부분에 site 라고 치면 D:\Trac\OOAD\htdocs 폴더로 자동 인식한다.

- ii. 재시작 없이 브라우저에서 [F5] 키를 눌러서 로고가 정상적으로 보이는지 확인한다.

GailCooper Project

Welcome to Trac 0.11

Trac is a **minimalistic** approach to **web-based** management c
enhancements and overall progress.

이제 로고가 정상적으로 보인다.

B. 시작 페이지 꾸미기

- i. **Edit this page** 를 선택한다.

Starting Points

- [TracGuide](#) -- Built-in Documen
- [The Trac project](#) -- Trac Op
- [Trac FAQ](#) -- Frequently Aski
- [TracSupport](#) -- Trac Support

For a complete list of local wiki page

- ii. **Wiki** 문법을 이용하거나 **rich edit** 버튼을 이용해 메시지를 바꾼다.

Editing WikiStart



= 이슈관리 홈 입니다 =

* 공지사항

- 이슈를 등록시에 담당자를 지정하지 마세요.
- [http://127.0.0.1:81/trac/00AD/prefs, Preferences]에서 이메일을 등록해 주세요.

이슈관리 HOME 입니다

* 공지사항

- 이슈를 등록시에 담당자를 지정하지 마세요.
- [Preferences](#)에서 이메일을 등록해 주세요.

Edit this page

Attach file

등록된 화면

C. 메뉴 이름 바꾸기

- Trac.ini 파일을 열어 [mainnav] 라는 항목을 만들어서 “메뉴.label = 새 메뉴이름” 형태로 저장한다.

```
[mainnav]
```

```
wiki.label = Home
```

wiki 메뉴의 이름은 Home 으로 바꾸고, tickets(View Tickets 메뉴)를 클릭하면 1 번 레포트가 자동으로 보여지게 만든다.

- 변경 전

	Wiki	Timeline	Ro
--	------	----------	----

메인메뉴 Wiki

- 변경 후

	Home	Timeline	Ro
--	------	----------	----

Wike → Home 으로 바뀌었다.

D. 기본 Report 메뉴 선택해서 보여주기

- trac.ini 파일을 열어 [mainnav] 라는 항목에 “메뉴.href = 경로” 형태로 내용을 추가 저장한다

```
[mainnav]
wiki.label = Home
tickets.href = /report/1
```

view tickets 선택하면 1 번 report 를 보여주도록 설정

ii. 변경 전



Available Reports

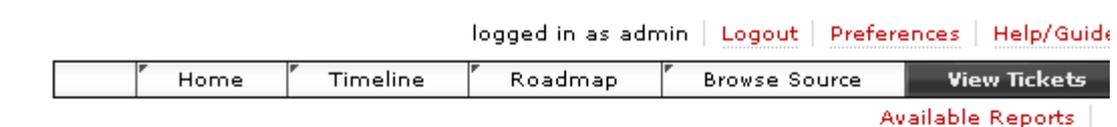
This is a list of available reports.

Report	Title
{1}	Active Tickets
{2}	Active Tickets by Version
{3}	Active Tickets by Milestone
{4}	Accepted, Active Tickets by Owner
{5}	Accepted, Active Tickets by Owner (Full Description)
{6}	All Tickets By Milestone (Including closed)
{7}	My Tickets
{8}	Active Tickets, Mine first

Note: See [TracReports](#) for help or

View Tickets 는 **Report** 목록 보여주고 있다

iii. 변경 후



{1} Active Tickets

- List all active tickets by priority.
- Color each row based on priority.

No matches found.

View Tickets 선택 시 자동으로 1 번 Report 가 보인다.

E. 첨부파일 사이즈 변경하기

```
[attachment]
```

```
max_size = 1024000
```

```
render_unsafe_content = true
```

F. 특정 링크(메뉴) 감추기

- i. Metanav 항목을 추가하여 disable 로 변경한다.

```
[metanav]
```

```
about = disabled
```

```
logout = disabled
```

- ii. 변경 전

logged in as admin	Logout	Preferences	Help/Guide	About Trac
Browse Source	View Tickets	New Ticket	Search	
	Start Page	Index	History	Last Change

de

- iii. 변경 후

logged in as admin	Preferences	Help/Guide	
View Tickets	New Ticket	Search	
Start Page	Index	History	Last Change

- G. 기타 상세한 설정 방법은 매뉴얼을 참고하기 바란다. Trac 이 설치되었으므로 메뉴의 Help/Guide 를 선택해서 로컬에서 확인 할 수 있다.

- i. 참조 페이지: <http://127.0.0.1:81/trac/ooad/wiki/TracInterfaceCustomization>

5. Trac 사용자 관리자(Admin) 페이지 설정하기

A. 특정 계정에 관리자 권한 부여

- i. 본 예제에서는 admin 유저가 관리자 라고 가정한다.
- ii. cmd 창을 열어서 trac-admin.bat 파일이 있는 trac install home 위치로 이동한다. (2.A 참조)
- iii. admin 계정에 관리자 권한을 부여한다.

```
trac-admin D:\WTrac\WOOAD permission add admin TRAC_ADMIN
```


- iii. Logging 메뉴는 로그를 남길지 여부를 결정한다. 특별한 이슈가 없다면 로그를 남기도록 만들자.

Administration



Logging

Configuration

Type:
File

Log level:
DEBUG

Log file:
trac.log

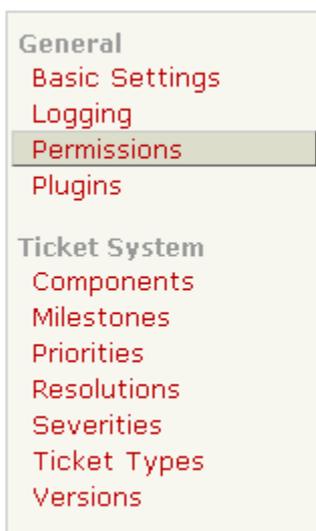
If you specify a relative path, #oad#log).

Apply changes

자세히 보면 D:\WTrac\woad\log 에 로그가 쌓인다고 설명되어 있다. 설정 후 Apply changes 를 누르는 것을 잊지 말자.

- iv. 권한(Permissions) 항목은 주체(Subject)에 권한을 라벨(label)형태로 주거나 뺏을 수 있다.

Administration



Manage Permissions

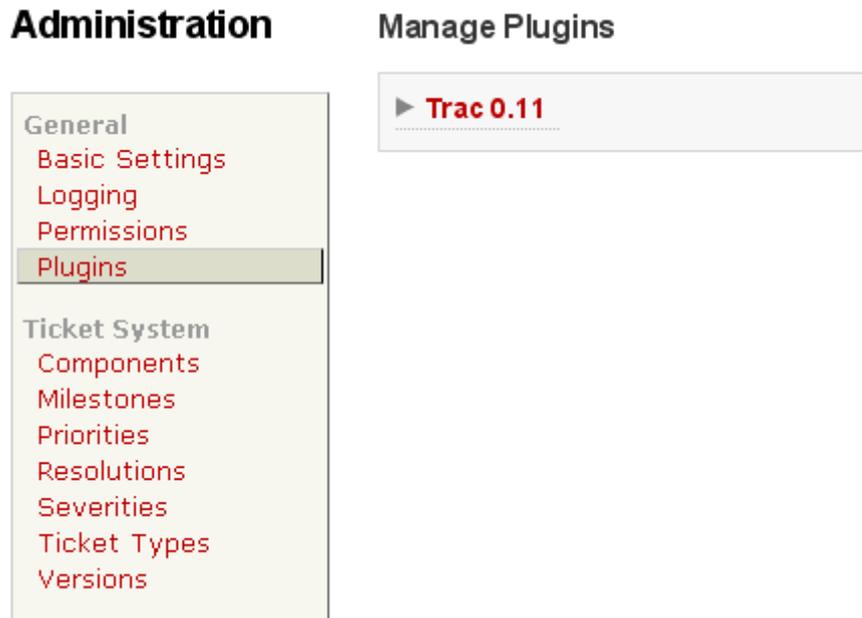
Note that *Subject* or *Group* names can't be all upper cased, as this is reserved for permission names.

Subject	Action
admin	<input type="checkbox"/> TRAC_ADMIN
anonymous	<input type="checkbox"/> BROWSER_VIEW
	<input type="checkbox"/> FILE_VIEW
	<input type="checkbox"/> MILESTONE_VIEW
	<input type="checkbox"/> REPORT_VIEW
	<input type="checkbox"/> SEARCH_VIEW
	<input type="checkbox"/> TIMELINE_VIEW
	<input type="checkbox"/> WIKI_VIEW
authenticated	<input type="checkbox"/> TICKET_CREATE
	<input type="checkbox"/> WIKI_CREATE
	<input type="checkbox"/> CHANGESSET_VIEW
	<input type="checkbox"/> LOG_VIEW
	<input type="checkbox"/> REPORT_SQL_VIEW
	<input type="checkbox"/> ROADMAP_VIEW
	<input type="checkbox"/> TICKET_VIEW
	<input type="checkbox"/> WIKI_MODIFY
	<input type="checkbox"/> WIKI_MODIFY

Remove selected items

주체와 라벨이라... MAC 모델인가? -,-;;

- v. 트랙은 플러그인 형태로 확장을 지원한다. 이를테면 xml-rpc 플러그인을 통해서 eclipse 플러그인(Mylyn)과 연동하는 것이 가능하다. 이에 대해서는 기회가 닿으면 다루어 보도록 하겠다.



C. Admin 페이지를 이용한 trac 프로젝트 티켓 시스템(Ticket System) 설정

티켓 시스템(Ticket System) 설정은 이슈 관리를 운영하는 데에 있어 중요한 부분이다. 워크 플로우(work flow)와도 관계가 있으며, 프로젝트 상황에 맞는 환경 구성이 필요하다. 따라서 업무 구성에 대한 규칙이 선행되어야 한다.

i. 이슈 관리 규칙 확인

본 예제에서는 아래의 표와 같다고 가정한다. 가상의 시스템인 관계로 프로젝트 설정 값이나 이름 자체엔 너무 집착하진 않길 바란다.

항목	프로젝트 설정 값	설명
모듈 구성(Component)	구매 등록 입고 처리 배송 처리	모듈의 단위를 통해 전체 프로젝트의 사이즈를 결정한다. 이를테면 모듈을 유즈케이스 단위로 정할 것인지, 업무 파트 단위로 할 것인지, 아니면 다른 어떤 것으로 할 것인지를 결정해야 한다.
마일 스톤(Milestones)	0.9 베타 릴리즈 1.0 정식 릴리즈	일반적으로 SW 프로젝트는 메이저 버전(1.0, 2.0 등등)이 있고 각 버전의 정식 버전(Stable Ver.)에 이르기까지 필요로 하는 요구사항(혹은 기능)을

		<p>만족시켜야 하는 마일스톤이 있다. 각종 방법론에서 이야기 하는 이터레이션(iteration)과 관련지어서 생각할 수 도 있지만, 그것보다는 흔히 말하는 alpha, beta, RC(release candidate), RTM 등과 비슷한 맥락으로 생각하면 된다. 예제에서는 간단하게 두 단계만 사용하였다.</p>
이슈 중요도(Priorities)	<p>Critical</p> <p>Major</p> <p>Minor</p>	<p>흔히 많이 사용되는 형태로는 아래와 같다.</p> <p>이슈는 항상 문제제기에만 사용되는 것이 아니므로, 결함과 기능, 두 가지 측면에서 볼 것을 권장한다. 전통적인 형태의 분류는 아래와 같다.</p> <p>blocker: 해결되지 않으면 더 이상의 업무진행이 불가. Show Stopper 라고도 부른다.</p> <p>critical: 금전관계나 무결성 훼손, 심각한 오류로 인한 고객 상실 야기의 수준. 제품의 핵심기능과 관련됨.</p> <p>major: critical 을 제외한 설계상의 오류나 기능의 실패. SW 의 주요 업무 진행에 관련됨.</p> <p>minor: 주 업무 진행에 관여되진 않았지만, 고객이나 사용자의 불만이 될 수 있는 오류. 혹은 사용을 편리하게 만들어 주는 추가 기능</p> <p>trivial: 사소한 오타나, 눈에 잘 띄지 않는 오류. 우선 순위가 상당히 낮은 기능 요구 사항</p>
해결(Resolution)	<p>수정 완료</p> <p>재검토 요망</p> <p>기타</p>	<p>해결 방식에 대해서는 프로젝트에 맞게 적절하게 선정한다. 참고로, 기타라는 항목은 자칫 독이 될 수 있으므로 만들지 여부를 신중히 고민하고, 기타로 분류된 내용들의 케이스를 분류해서 향후 추가 카테고리를 만들 때 이용하는 수준으로 사용하길 권장한다.</p> <p>참고로 전통적인 버그 트랙에서는 해결을 다음과</p>

		<p>같은 식으로 분류한다.</p> <p>Fixed: 수정됨</p> <p>Invalid: 데이터가 불충분함</p> <p>Duplicate: 이미 보고된 문제임</p> <p>Wontfix: 수정할 필요가 없음</p> <p>WorksForMe: 재현 불가능</p>
심각도(Severities)	설정하지 않음	<p>심각도는 이슈 중요도(Priorities)와 상당부분 겹쳐질 수 있으므로, 좀 더 엄격한 관리가 필요할 경우에만 사용할 것을 권장한다. trac 에서도 기본값은 설정하지 않음이며, 이슈 등록할 때 아예 표시가 되지 않는다.</p> <p>* 항목이 없으면 표시조차 되지 않는 건 trac 모든 항목메뉴들의 공통사항이다</p>
티켓 종류(Ticket types)	<p>결함</p> <p>모호한 표현</p> <p>기타</p>	<p>티켓의 종류도 이슈 중요도와 마찬가지로 단순히 오류상황만을 보는 것이 아니라, 경우에 따라 크게 업무(task), 기능개선(enhancement), 결함(Defect) 등으로 나누어서 생각할 수 있다. 본 예에서는 trac 을 결함관리 위주로 사용하였기에 '결함, 모호한 표현, 기타'의 세 가지로 나누었다.</p>
버전(Versions)	설정하지 않음	<p>출시된 제품을 관리하고자 할 경우에는 마일스톤(milestone) 만으로 충분하지 않을 수 있다. 이런 경우 출시된 버전(stable)과 마일스톤을 함께 사용해서 관리 할 수 있다.</p>

ii. 모듈 구성(Component)

Administration

General
Basic Settings
Logging
Permissions
Plugins
Ticket System
Components
Milestones
Priorities
Resolutions

Manage Components

	Name	Owner	Default
<input type="checkbox"/>	구매 등록	user01	<input checked="" type="radio"/>
<input type="checkbox"/>	배송 처리		<input type="radio"/>
<input type="checkbox"/>	입고 처리		<input type="radio"/>

Remove selected items

Apply changes

You can remove all items from this list to completely hide this field from the user interface.

모듈 담당자(owner)를 지정해 줄 수도 있다. 기본 선택 값을 '구매 등록'으로 선택하였다.

iii. 마일스톤(Milestones)

Manage Milestones

	Name	Due	Completed	Default
<input type="checkbox"/>	0.9 베타 릴리즈	2008-12-31		<input checked="" type="radio"/>
<input type="checkbox"/>	1.0 정식 릴리즈	2009-01-31		<input type="radio"/>

Remove selected items

Apply changes

You can remove all items from this list to completely hide this field from the user interface.

Add Milestone: _____

Name:

Due:

Forma

YYYY-MM-DD

Add

마감(Due)을 지정해 놓으면 RoadMap 및 기타 화면에서 남은 일정을 표시해 준다.

Roadmap

Milestone: 0.9 베타 릴리즈

Due in 2 days (2008-12-31)

iv. 이슈 중요도(Priorities)

Manage Priorities

	Name	Default	Order
<input type="checkbox"/>	critical	<input type="radio"/>	1 ▼
<input type="checkbox"/>	major	<input checked="" type="radio"/>	2 ▼
<input type="checkbox"/>	minor	<input type="radio"/>	3 ▼

v. 해결(Resolution)

Manage Resolutions

	Name	Default	Order
<input type="checkbox"/>	수정 완료	<input checked="" type="radio"/>	1 ▼
<input type="checkbox"/>	재검토 요망	<input type="radio"/>	2 ▼
<input type="checkbox"/>	기타	<input type="radio"/>	3 ▼

vi. 심각도(Severities)

Manage Severities

As long as you don't add any items to the list, this field will remain completely hidden from the user interface.

심각도는 설정하지 않았다.

vii. 티켓 종류(Ticket types)

Manage Ticket Types

	Name	Default	Order
<input type="checkbox"/>	결함	<input checked="" type="radio"/>	1 ▼
<input type="checkbox"/>	모호한 표현	<input type="radio"/>	2 ▼
<input type="checkbox"/>	기타	<input type="radio"/>	3 ▼

viii. 버전(Versions)

Manage Versions

	Name	Released	Default
<input type="checkbox"/>	2.0		<input type="radio"/>
<input type="checkbox"/>	1.0		<input checked="" type="radio"/>

버전(version)은 출시 되고 나서 해당 버전을 관리하는 형태이다. 출시된 이후에 관리할 때 쓰인다. 시/분/초는 입력하지 않도록 한다. 본 예제에서는 관리포인트 최소화를 위해 아래처럼 Version 자체를 관리하지 않았다.

Manage Versions

As long as you don't add any items to the list, this field will remain completely hidden from the user interface.

ix. 설정을 마친 후의 이슈(=티켓) 등록 화면

Create New Ticket

Properties

Summary:

Reporter:

Description: **B** *I* **A**      

Assign to:

Priority:

Component:

Cc:

Type:

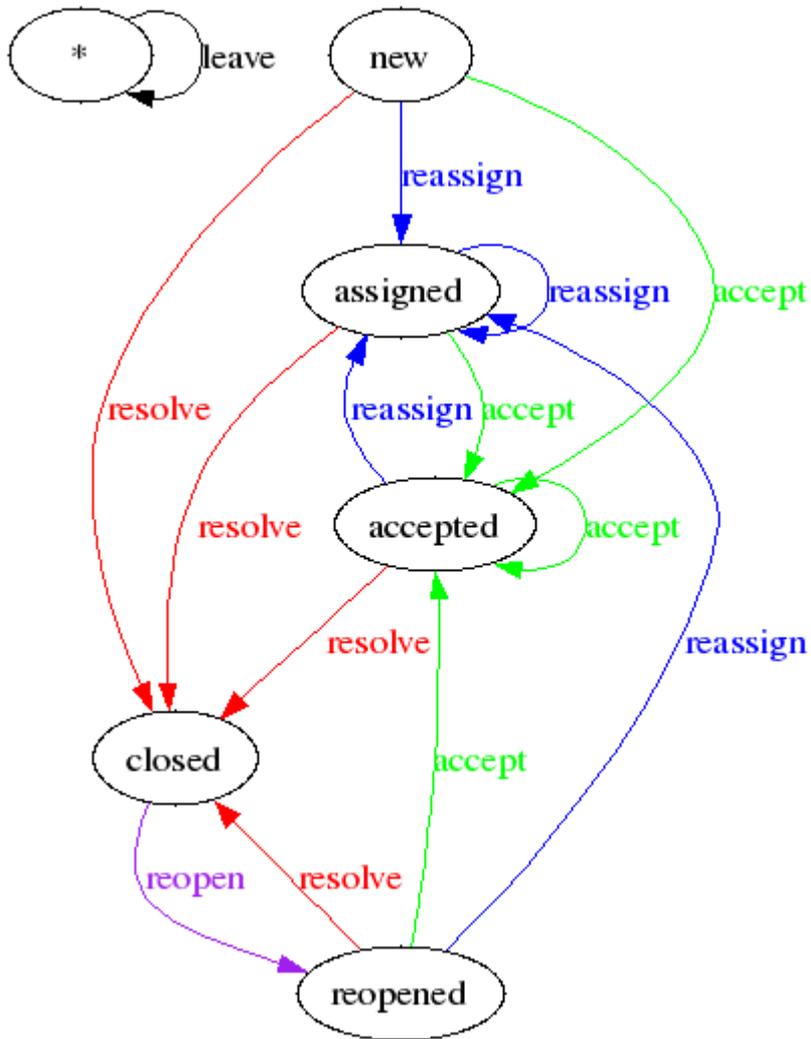
Milestone:

Keywords:

I have files to attach to this ticket

6. Trac 티켓 업무흐름도

- A. 티켓은 발생(new)과 함께 업무 흐름을 갖게 된다. 아래, 그림은 단순히 티켓들의 상태를 표시할 뿐이며, 실제로 상태 중간 중간에 업무역할(개발자, Tester, QA 등등)이 위치하게 된다.



7. Trac 활용 화면 샘플

A. 텍스트 큐브 개발 환경 (권장)

- i. <http://dev.textcube.org/wiki>
- ii. <http://blog.tattertools.com/153> (디자이너의 사용후기)

B. 교육용 환경 샘플

logged in as user01 | [Preferences](#) | [Help/Guide](#)

[Home](#) | [Timeline](#) | [Roadmap](#) | [Browse Source](#) | [View Tickets](#) | [New Ticket](#) | [Search](#)

[Start Page](#) | [Index](#) | [History](#) | [Last Change](#)

분석/설계 실무자 과정[5조 B.S.M팀]

본 페이지는 이슈 트래커인 Trac 을 사용한 Project Home 입니다.
프로젝트 이슈 관리 및 소스관리, 일정등을 관리합니다.

[Edit this page](#) | [Attach file](#)

Download in other formats:
Plain Text

 Powered by [Trac 0.11](#)
By [Edgewall Software](#).

분석 설계 실무자 과정 이슈 관리 페이지

logged in as admin | [Preferences](#) | [Help/Guide](#)

[Home](#) | [Timeline](#) | [Roadmap](#) | [Browse Source](#) | [View Tickets](#) | [New Ticket](#) | [Search](#) | [Admin](#)

Roadmap

Milestone: 0.9 베타 릴리즈

Due in **13 hours** (2008-12-20)



Closed tickets: 15 | *Active tickets: 0* / *Total tickets: 15*

단위 테스트 수행용 릴리즈

Show already completed milestones

[Update](#)

Milestone: 1.0 릴리즈

No date set

	Home	Timeline	Roadmap	Browse Source	View Tickets	New Ticket	Search	Admin
--	----------------------	--------------------------	-------------------------	-------------------------------	-------------------------------------	----------------------------	------------------------	-----------------------

Custom Query (16 matches)

▶ **Filters**

▶ **Columns**

Group results by Component descending Show under each result: Description Max items per page

Component: 상품구매 (1 match)

Ticket	Summary	Owner	Type	Priority ▲	Component	Resolution
#4	[상품구매] 티켓	상품구매	결함	major	상품구매	수정 완료

Component: 입고 (4 matches)

Ticket	Summary	Owner	Type	Priority ▲	Component	Resolution
#15	[입고] 구매내역 조회 query 정보 누락	user03	결함	major	입고	수정 완료
#6	입고처리 화면설계서	user03	기타	minor	입고	수정 완료
#10	[입고] 쿼리의 컬럼 누락	user03	결함	minor	입고	수정 완료
#13	[입고] 메소드의 query문 검토요청	somebody	기타	minor	입고	수정 완료

Component: 출고 (6 matches)

Ticket	Summary	Owner	Type	Priority ▲	Component	Resolution
#3	클래스설계서에서 jsp 부분 확인요청드립니다.	somebody	결함	major	출고	수정 완료
#9	출고관련	user02	결함	major	출고	수정 완료
#11	[출고] 화면과 쿼리 및 LData의 항목 상이	user02	결함	major	출고	수정 완료
#12	[출고] 출고수량 항목 누락, 출고확인버튼 클릭시 함수호출 관련	somebody	결함	major	출고	수정 완료
#14	[출고] 항목누락, 매개변수관련	user02	결함	major	출고	수정 완료
#2	출고(배송) _ 클래스 설계서 (예외처리관련)	somebody	결함	minor	출고	수정 완료

Component: 쿠폰 (6 matches)

Ticket	Summary	Owner	Type	Priority ▲	Component	Resolution
#5	쿠폰 화면항목의 설명이 ERD의 column과 상이	user04	결함	major	쿠폰	수정 완료
#8	쿠폰취득_상세페이지 Hidden값 누락	user04	기타	major	쿠폰	수정 완료

Ticket #4 (closed 결함: 수정 완료)

[상품구매] 티켓		Opened 21 hours ago	
		Last modified 21 hours ago	
Reported by:	guest	Owned by:	상품구매
Priority:	major	Milestone:	0.9 베타 릴리즈
Component:	상품구매	Keywords:	
Cc:	jinsunkim@lgcns.com		
Description			
<p>1. GoodsDao의 retrievePurchasablGoodsListDao 메소드 명이 retrievePurchasableGoodsList을 의도했던 것인지 궁금합니다.</p> <p>2. 위의 메소드에서 사용하는 SQL문에서 GOODS_UNITPRICE와 PCHS_AMT는 DB에 존재하지 않는 항목명인데 각각 어떤 것으로 대체해서 사용해야하는지 궁금합니다.</p>			Reply

Attachments

Attach file

Change History

Changed 21 hours ago by user01	
<ul style="list-style-type: none"> ▪ status changed from <i>new</i> to <i>closed</i> ▪ resolution set to 수정 완료 <p>1. retrievePurchasablGoodsListDao의 오타였습니다.;;</p> <p>2. 쿼리 수정하여 업데이트하였습니다.</p>	Reply

Add/Change #4 ([상품구매] 티켓)

	Home	Timeline	Roadmap	Brow
--	------	----------	---------	------

Changeset 93

Timestamp: 2008-12-19 오전 10:55:23 (less than one hour ago)

Author: user03

Message: 구매 Form 중 불필요한 year 삭제, inhousePmntYmd 추가

Files:  1 modified

 [oodPrj/web/purchase2/requestPurchase.jsp \(1 diff\)](#)

Unmodified Added Removed

oodPrj/web/purchase2/requestPurchase.jsp		
r89	r93	
119	119	<TD align=center><INPUT TYPE=INPUT NAME=purchaseQty SIZE=4;"></TD>
120	120	<TD align=center><%= goodsList.getString("pchsUnitprice", in) %> 원</TD>
121		<TD align=center><INPUT TYPE=INPUT NAME=year SIZE=8;"></TD>
	121	<TD align=center><INPUT TYPE=INPUT NAME=inhousePmntYmd SIZE=8;"></TD>
122	122	</TR>
123	123	<%

Home

Timeline

Roadmap

Browse Source

View

Timeline

2008-12-19: Today

- ✔ 10:28 Ticket [#15](#) (출고이력 생성시 For문에 대한 언급이 없음.) closed by user01
 수정 완료: 수정했습니다.
- ✔ 10:27 Ticket [#14](#) (public boolean checkInvQty 의 인수문제) closed by user01
 수정 완료: 수정했습니다.
- ✉ 10:21 Ticket [#15](#) (출고이력 생성시 For문에 대한 언급이 없음.) created by guest
 클래스 설계서 6page에서 1. 전달받은 dgwItemList로부터 ...
- ✉ 10:10 Ticket [#14](#) (public boolean checkInvQty 의 인수문제) created by guest
 public boolean checkInvQty (String goodsSerno, int dgwQty) throws ...
- ✔ 09:43 Ticket [#13](#) (ORD_ITEM_SERNO가 없음.) closed by user01
 수정 완료
- ✉ 09:21 Ticket [#13](#) (ORD_ITEM_SERNO가 없음.) created by guest
 클래스 설계서의 retrieveConfrimDgw 에서 (9Page) ORD_ITEM_SERNO 를 가져오지 않고 ...

2008-12-18: Yesterday

- ✔ 20:33 Ticket [#12](#) (출고 화면설계서 전체) closed by user01
 수정 완료: 처리완료되었습니다.
- ✉ 20:13 Ticket [#12](#) (출고 화면설계서 전체) created by guest
 화면설계서의 모든 필드가 E 아니면 H로 되어 있는데 사실 넘어가서 사용하는 값은 몇개 안됩니다. 이 값을 전부다 hidden ...
- ✔ 20:04 Ticket [#10](#) (상품구매 - 메일 송신 관련..) closed by user02
 수정 완료
- ✔ 19:43 Ticket [#11](#) (출고화면에서 order by의 설명이 부족합니다.) closed by user01
 수정 완료: order by 주문번호 desc 최근순으로 먼저 보는것이 나올 것 같습니다.

8. Trac FAQ

A. Trac 사용의 이점은 무엇입니까?

- i. 이 내용은 Trac 이라기 보다는 이슈 트래커 사용의 이점이 무엇인지에 대해 이야기 하는 것이 맞습니다. 이슈 트래커의 대표적인 장점은, 프로젝트 이슈의 투명성, 공동작업에 효율성 증진, 지식축적, 릴리즈관리, 소스연동을 통한 접근성 강화 등이 있습니다. 이슈(=장애, 버그, 기능개선 요청 등)를 관리하는 전통적인 방식인 엑셀 파일관리나 웹 게시판 등의 방법은 이슈의 라이프사이클 제대로 제공하지 못하며, 결국 경험의 축적이 자산화로 연결되지 못하게 되는 흔한 경우를 만든다고 보시면 되겠습니다 프로젝트의 이슈는 프로젝트에서 정의한 이슈 작업흐름(work flow)에 따라 생성에서 종료까지가 이슈의 상태기반으로 담당자와 작업이

유기적으로 연결되어야 하며, 그 과정이 투명하게 보여야 하고, 이력(history)으로 남겨질 수 있어야 하겠습니다. 이슈 트래커들은 그 과정을 시스템 적으로 지원합니다.

B. Trac 은 어느 정도 규모에서 사용가능 한가요?

- i. Trac 에서 제공하는 Category(=component) 의 한계상 다층 깊이의 hierarchy 는 제공하고 있지 않습니다. 따라서 부서와 파트가 많아지고 모듈이 많아지면 복잡도가 증가할 수 있습니다. 하지만 그 규모의 수준이랑게 기반 프로젝트를 어떻게 하위 프로젝트로 사이징 할 수 있는냐에 따라 달라질 수 있습니다. 국외 사례는 제외 하더라도, 국내에서 대표적으로 알려준 케이스를 꼽자면, DAUM 에서는 Trac 을 전사적 기본 이슈 관리도구로 사용했으며, 최근 구글에서 인수한 TNC(태더&컴패니)의 유명한 블로그 툴인 Tatter Tools (Tistory 기반툴) 개발에 Trac 이 사용되었습니다. (여담으로 이야기 드리자면, 최근 DAUM 은 JIRA 라는 상용 제품으로 이슈 관리도구를 변경한 것으로 알고 있습니다. NAVER 는 JIRA 가 예전부터 표준 툴이었다고 들었습니다.) 우선은 Trac 이나 Mantis 등의 오픈소스 툴을 사용하고, 상용툴로의 이전은 복잡도와 프로젝트 상황에 맞추어 가면 될 것 같습니다.

C. 이슈 관리의 단점은 없습니까?

- i. 이슈 관리의 단점이라기 보다는 이슈 자체에 대한 잘못된 해석이 건강한 이슈관리를 훼손할 수 있습니다. 이를테면, '이슈를 많이 만드는 팀은 문제 있는 팀' 이라던가, '결함이 많이 발견되는 모듈 개발자는 능력이 다소 부족한 개발자' 같은 시각으로 일련의 '평가'와 이슈를 결부시키려고 하는 경우가 있습니다. 이것은 프로젝트의 이슈(=위험, Risk)자체를 투명하게 이끌어내어서 적극적으로 관리하고 경험으로 축적하고자 하는 이슈 트랙 시스템이 가진 선의의 의지와 상반되는 형태로 이슈 자체의 은폐를 조장할 가능성이 있습니다. 따라서, 절대로 이슈는 해결해야 하는 Task 그 이상도 이하로도 여기지 않는 풍토를 만드는 것이 중요합니다.

D. 기존 시각에서 볼 때 새로운 툴이라는 것에 대한 부정적인 시각은 어떠한가요?

- i. 저는 생산성을 높이기 위해서라면 사용할 수 있는 자원은 최대한 사용해야 한다고 봅니다. 한 사람이 고생해서 2 사람 이상이 편해 질 수 있다면(즉, ROI 가 맞는다면), 누군가는 해야 합니다. '툴은 툴일 뿐!'이라는 담언은 툴을 사용하지 말라는 담언이 아닙니다. 최대한 잘 사용하되 의존적이어서는 안된다는 이야기라는걸 모두들 아실 겁니다. 최초에 컴퓨터가 사무업무 기반을 대체할 수 있다고 생각한 사람은 많지 않았습디다. 하지만 그 생산성을 당해낼 수 없었기에 따라갈 수 밖에 없었으며, 중요한 점은 조기에 얼마나 혼란 없이 받아 들일 수 있는가가 관건이라 생각합니다.

E. 기타 이슈 관리 툴에는 어떠한 것들이 있나요?

- i. MANTIS, 정확히는 MANTS BT (버그 트래킹), MANTIS BTS 등으로 불립니다. 버그 수집에 초점이 맞추어져 있었는데, 최근에는 보고서 기능을 필두로 이슈 관리쪽으로도 사용된다고 합니다. 데모는 http://www.mantisbt.org/demo/my_view_page.php 에서 보실 수 있습니다.
 - ii. JIRA, 정확히는 Atlassian JIRA 입니다. 상용 툴로써 현재까지는 가장 활발히 이용되고 있고, 가장 발전된 형태라고 칭해지고 있습니다. <http://www.atlassian.com/software/jira/screenshots/default.jsp> 에서 화면을 보실 수 있습니다. 요즘은 대부분 JIRA 로 가는 추세인 것 같습니다. <http://jira.springframework.org/browse/SPR> 링크는 스프링 프레임워크의 이슈 트랙화면입니다. 마찬가지로 JIRA 로 구성되어 있습니다.
 - iii. Bugzilla, 가장 오래된 툴입니다. <https://landfill.bugzilla.org/bugzilla-tip/request.cgi> 에서 살펴보실 수 있습니다. 인터페이스가 다소 고전적이고, 버그 추적에 집중되어 있으며, 세분화된 항목들로 인해 이슈 등록 시에 다소 혼란스러울 수 있습니다.
 - iv. 비교 관련 글로는 http://www.ibm.com/developerworks/kr/library/s_issue/20071127/ 를 참조하시면 되겠습니다.
 - v. 이 외에도 다양하게 존재하고 있으며, 좀 더 머리가 복잡해 지고 싶으시면 <http://www.michaelflanakin.com/Articles/Comparisons/WebBasedIssueTrackers/tabid/198/Default.aspx> 에 들어서 본격적으로 살펴보시는 것도 나쁘지 않은 선택이라 봅니다.
- F. 이슈를 제기하는 과정(new ticket)에서 OS 라던가, 브라우저 버전이라던가 하는 내용도 포함시키려면 어떻게 해야 하나요?**
- i. Trac 에서는 custom field 라고 해서 원하는 필드를 추가하는 기능을 제공하고 있습니다. <http://trac.edgewall.org/wiki/TracTicketsCustomFields> 을 참조하세요.
- G. 티켓을 발급했는데, 오타가 있더라고요. 수정하고 싶은데 어떻게 해야 하나요?**
- i. 웹 admin 화면이나 trac-admin 커맨드를 사용해서 TICKET_MODIFY(혹은 TICKET_EDIT 시리즈) 권한을 주면 수정이 가능합니다. '자기가 게시한 항목만 수정 가능'식의 권한 부여는 TICKET_EDIT 시리즈를 이용하세요. 그리고, 어떠한 경우에도 명시적으로 누가 마지막으로 수정했고, 이전과의 차이점이 무엇인지는 바로 해당 화면에서 볼 수 있습니다.
- H. 각종 항목들을 한글화 하고 싶은데 어떻게 하면 되나요?**
- i. Edit 페이지 버튼이 있을 경우 눌러서 한글로 메시지를 교체하면 됩니다. 기본적으로 trac 이 위키(wiki)방식으로 되어 있어서 변경 가능합니다. 그리고 차기 버전에는 다국어 지원이 기본으로 탑재될 예정입니다. (현재 한글화 진행 중입니다)
- I. 보고서(Report)가 원하는 형태가 아닌데 바꿀 수 없나요?**

- i. Report 는 custom query 에서 필터나 조건을 사용하여 쿼리를 만들어 낸 다음 save query 버튼을 눌러서 Report 로 등록하면 됩니다. 물론, 현재 있는 report 에 이어서 번호가 붙습니다.

J. Report 번호를 수정하고 싶은데요, 어떻게 해야 하나요?

- i. 번호 수정은 DB 에 직접 접근하는 방식 이외에는 수정이 안됩니다. 다만 기본 report 선택은 trac.ini 파일의 mainav 항목에서 조절 가능합니다.

K. 소스코드의 한글이 깨져 나옵니다.

- i. Trac.ini 파일에서 encoding 부분을 적당한 한글 인코딩 타입으로 수정해 줍니다.

```
[trac]
authz_file =
authz_module_name =
auto_reload = False
base_url =
check_auth_ip = true
database = sqlite:db/trac.db
default_charset = MS949
htdocs_location =
```

L. 등록되는 모든 이슈를 받아보고 싶은데요. 방법이 없나요?

- i. 가의 열의가 뜨겁다 할 수 있겠습니다. 그리고 자신의 메일함을 희생할 각오도 있으시군요. Trac.ini 파일의 [notification] 항목을 살펴보면, 다양한 형태로 메일을 받으실 수 있습니다. 이를테면 항상 숨은 참조자로 메일 받기(smtp_always_bcc) 같은 항목이 그렇습니다. 마찬가지로 관련해서는 help/guide 를 살펴보시기 바랍니다.

[부록] Trac 추가 기능 활용예

1. Trac Wiki

- A. Moinmoin wiki 문법을 따릅니다.
- B. Help/Guide 항목의 TracWiki 페이지를 참조합니다.

2. Source Syntax Highlight

- A. Python 2.5 를 다운 받아서 home 디렉터리를 D:\VisualSVN Server\trac\python 로 선택해서 인스톨 합니다.

- B. <http://silvercity.sourceforge.net/> 에 접속해서 Python 2.5 용 인스톨 파일을 내려받아서 Python home 을 위에서 정한 D:\VisualSVN Server\trac\python 로 선택해서 인스톨 합니다.
- C. Cmd 창을 열어서 CD %APPDATA%\Subversion 으로 이동 후 config 파일을 아래와 같이 수정합니다.

```
[miscellany] 영역의 enable-auto-props = yes 주석을 풀어주고
[auto-props] 영역에
*.java = svn:mime-type=text/x-java
*.jsp = svn:mime-type=text/x-java
추가 한다.
```

- D. VisualSVN 재기동으로 완료

3. Eclipse Plugin 연동

- A. Xml-rpc plugin 설치가 필요합니다.
- B. 이클립스 mylyn 사이트에서 trac 용 connector 를 download 해야 합니다.
- C. Python 을 이용한 컴파일 작업도 필요합니다.
D:\VisualSVN Server\trac\python\xmlrpcplugin\trunk>python setup.py bdist_egg
- D. <http://trac-hacks.org/wiki/XmlRpcPlugin> 페이지의 installation 부분을 참조합니다.
- E. 이클립스 연동 관련해서는 <http://unitstep.net/blog/2008/01/19/using-assemblas-trac-with-eclipse-mylyn-xml-rpc-access/> 페이지를 참고합니다.

문서 개정이력

날짜	변경 사항	작업자
2008.12.23	1.0 작성 완성	채수원
2009.01.19	프로젝트 예제[부록2] 및 기타 프로젝트 정보 제거 후 public open	채수원

문서관련 문의는 doortts@gmail.com 으로 문의 주세요.