



OpenGXT Developer Guide



OPENGXT

OpenGXT Developer Guide v1.0

2019 년 10 월 1 일 1 판 발행

지은이: 이민파, 김지현

감수: 양계용

표지 디자인: 박재현

편집: 김지현

펴낸이: (주)망고시스템

펴낸곳: 가이아쓰리디㈜

주소: 대전 유성구 관평동 1359 한신에스메카 230 호

전화: 042-330-0400

팩스: 042-330-0410

출판등록: 제 2012-000016 호

ISBN: 978-89-969532-9-6(95000)

이 책은 비매품입니다.

Copyright Notice

Copyright © 2010-2019 MangoSystem Inc. All Rights Reserved.

Address: 2307-ho, Pyengchon O'biztower, 126, Beolmal-ro, Dongan-gu, Anyang-si, Gyeonggi-do, 431-060 South Korea

Tel: 82-31-450-3411 Fax: 82-31-450-3414

E-mail: master.mangosystem@gmail.com

Homepage: <http://www.mangosystem.com>

Online Documentation: <http://gxt.mangosystem.com>

Restricted Rights Legend

이 소프트웨어(OpenGXT for Server®, OpenGXT for Desktop®) 개발자가이드의 내용은 저작권법, 컴퓨터 프로그램 보호법 및 국제 조약에 의해서 보호받고 있습니다. 개발자가이드의 내용은 MangoSystem Inc. 와 사용권 계약 하에서만 사용이 가능하며, 사용권 계약을 준수하는 경우에만 사용 또는 복제할 수 있습니다.

이 개발자가이드의 전부 또는 일부분을 MangoSystem의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물작성 등의 행위를 하여서는 안됩니다.

Trademarks

OpenGXT for Server®, OpenGXT for Desktop®은 MangoSystem Inc. 의 등록 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

Open Source Software Notice

본 제품은 "GeoTools", "GeoServer", "uDig"에 의해 개발 또는 라이선스 된 오픈 소스 소프트웨어를 포함합니다. 관련 상세 정보는 아래에 기재된 사항을 참고해 주십시오.

- GeoTools: <http://www.geotools.org>
- GeoServer: <http://geoserver.org>
- uDig: <http://udig.refractory.net>

안내서 정보

안내서 제목: Developer Guide for OpenGDS-Analysis

발행일: 2019-10-01

소프트웨어 버전: OpenGXT for Server 1.0

소프트웨어 버전: OpenGXT for Desktop 1.0

안내서 버전: 1.0



목 차

1. Introduction	6
1.1. 개요	6
1.2. 문서의 구성	7
2. How to Build	8
2.1. 개발환경 설치	8
2.1.1. JDK (Java SE Development Kit) 7 설치	8
2.1.2. Apache Maven 설치	9
2.1.3. GitHub Desktop 설치	11
2.2. 기반 소프트웨어 빌드	12
2.2.1. GeoTools (선택사항)	12
2.2.2. GeoServer (선택사항)	13
2.2.3. uDig	14
2.3. OpenGDS/Analysis 빌드	16
2.3.1. GeoTools Analysis Process	16
2.3.2. GeoServer WPS Extension	17
2.4. uDig 배포버전 생성 (선택사항)	18
2.4.1. NSIS 설치 (선택)	18
2.4.2. uDig Product Build	19
2.4.3. Deploy	19
3. How to Customize	21

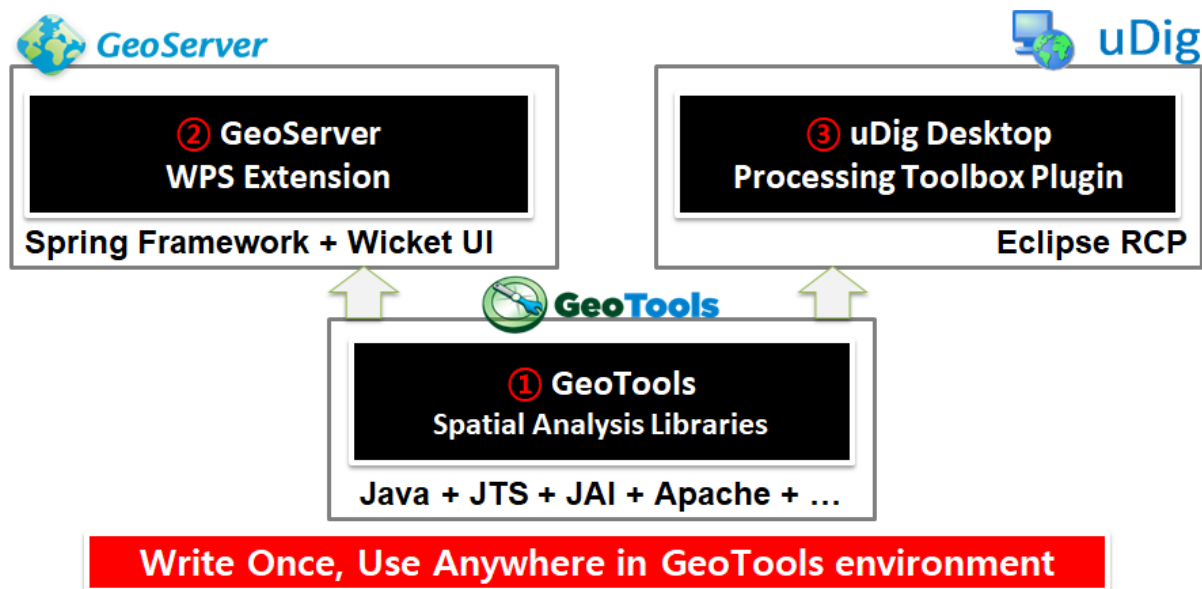
3.1.	Eclipse 에서 개발환경 설정.....	21
3.1.1.	Eclipse IDE 설치	21
3.1.2.	플러그인 설치	22
3.1.3.	Workspace 개발환경 설정	24
3.2.	OpenGDS/Analysis: GeoTools, GeoServer 커스터마이징	31
3.2.1.	프로젝트 불러오기	31
3.2.2.	프로세스 추가	32
3.2.3.	GeoServer WPS – Custom PPIO 추가	39
3.3.	OpenGDS/Analysis: uDig 커스터마이징	42
3.3.1.	uDig Target Platform 설정.....	42
3.3.2.	uDig 실행	44
3.3.3.	프로젝트 불러오기	45
3.3.4.	프로세스 추가	46
3.3.5.	플러그인 빌드 및 배포	47
4.	How to Use	48
4.1.	GeoServer WPS 설치 및 활용	48
4.2.	uDig 플러그인 설치 및 활용	48
5.	References	49
5.1.	How to Contribute.....	49
5.2.	Reference	49

1. Introduction

1.1. 개요

2014 년부터 현재까지 (주)망고시스템은 국토공간정보연구사업의 하나인 [공간정보 SW 활용을 위한 오픈소스 가공기술 개발¹ - 오픈소스 기반 공간자료 분석기술 개발] 과제를 수행 중에 있습니다.

이 문서의 목적은 Windows 환경에서 공간자료 분석기술 개발과제의 성과물 (OpenGDS/Analysis)에 대한 개발환경의 구성, 빌드, 배포, 활용, 커스터마이징에 대한 전 과정을 설명합니다. OpenGDS/Analysis 는 GeoTools², GeoServer³, uDig⁴ 기반에서 운영되므로 3 개의 프로젝트가 모두 운영될 수 있는 개발환경을 구성합니다.



¹ <http://opensdi.org>

² <http://geotools.org>

³ <http://geoserver.org>

⁴ <http://udig.refractory.net>

1.2. 문서의 구성

이 문서는 다음과 같이 3 개의 장으로 구성되어 있습니다.

■ [How to Build](#)

- 개발환경 설치
- 기반 소프트웨어 빌드
- OpenGDS/Analysis 빌드
- uDig 설치파일 빌드

■ [How to Customize](#)

- GeoTools Process 생성
- uDig 플러그인 Process 추가

■ [How to Use](#)

- GeoServer WPS 설치 및 활용
- uDig 플러그인 설치 및 활용

2. How to Build

이 장은 개발환경 설치, 기반 소프트웨어 빌드, OpenGDS/Analysis 빌드, uDig 설치파일 빌드로 구성되어 있습니다.
















2.1. 개발환경 설치

필수 개발 환경인 Java Development Kit 7(JDK 1.7), Apache Maven, Eclipse IDE, GitHub Desktop 을 설치하는 과정입니다.

2.1.1. JDK (Java SE Development Kit) 7 설치

다음의 [Java Archive Downloads - Java SE 7 – Oracle] 페이지로 이동하여 JDK 7 최신버전을 다운로드 합니다.

- <http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html>

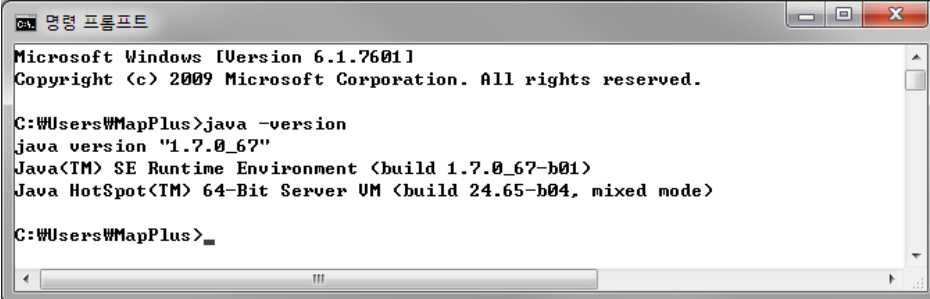
Java SE Development Kit 7u80		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux x86	130.44 MB	 jdk-7u80-linux-i586.rpm
Linux x86	147.68 MB	 jdk-7u80-linux-i586.tar.gz
Linux x64	131.69 MB	 jdk-7u80-linux-x64.rpm
Linux x64	146.42 MB	 jdk-7u80-linux-x64.tar.gz
Mac OS X x64	196.94 MB	 jdk-7u80-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.77 MB	 jdk-7u80-solaris-i586.tar.Z
Solaris x86	96.41 MB	 jdk-7u80-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.72 MB	 jdk-7u80-solaris-x64.tar.Z
Solaris x64	16.38 MB	 jdk-7u80-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	140.03 MB	 jdk-7u80-solaris-sparc.tar.Z
Solaris SPARC	99.47 MB	 jdk-7u80-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24.05 MB	 jdk-7u80-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.41 MB	 jdk-7u80-solaris-sparcv9.tar.gz
Windows x86	138.35 MB	 jdk-7u80-windows-i586.exe
Windows x64	140.09 MB	 jdk-7u80-windows-x64.exe
Back to top		

다운로드 한 파일(예: jdk-7u80-windows-x64.exe)을 더블 클릭하여 java 를 설치합니다.

[제어판] → [시스템] → [고급 시스템 설정]으로 이동하여 [시스템 속성] 창에서 다음과 같이 [고급] 탭의 [환경변수(N)...] 버튼을 눌러 시스템 변수에 다음 변수를 확인 또는 등록합니다.

변수 이름	변수 값
JAVA_HOME	C:\Program Files\Java\jdk1.7.0_80

이제 [명령 프롬프트]를 실행 후 `java -version` 명령어를 입력하여 설치를 확인합니다.



```

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\WMapPlus>java -version
java version "1.7.0_67"
Java(TM) SE Runtime Environment (build 1.7.0_67-b01)
Java HotSpot(TM) 64-Bit Server VM (build 24.65-b04, mixed mode)

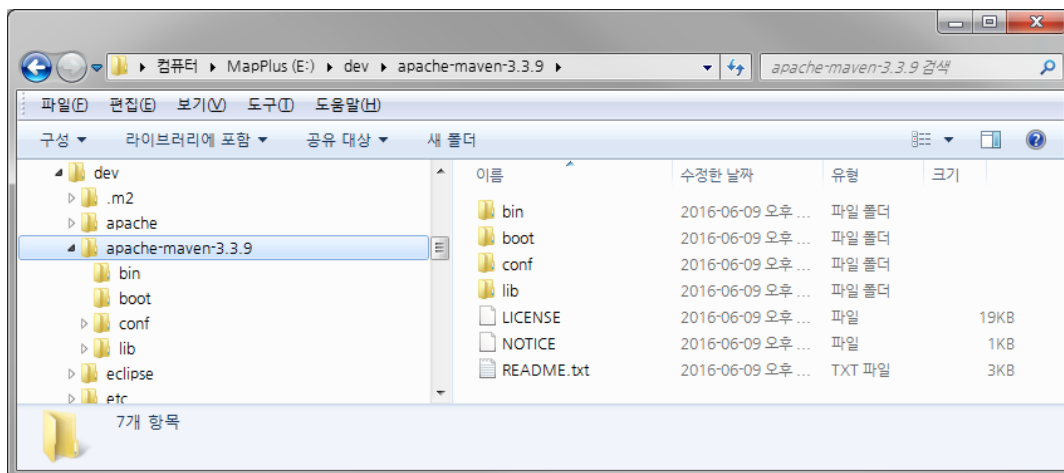
C:\Users\WMapPlus>
  
```

2.1.2. Apache Maven 설치

소프트웨어 프로젝트 관리 도구인 Apache Maven 설치 과정입니다. 다음의 Apache Maven Project 사이트로 이동하여 최신버전(2016 년 6 월 현재 3.3.9 버전)을 다운로드 합니다.

- <https://maven.apache.org>

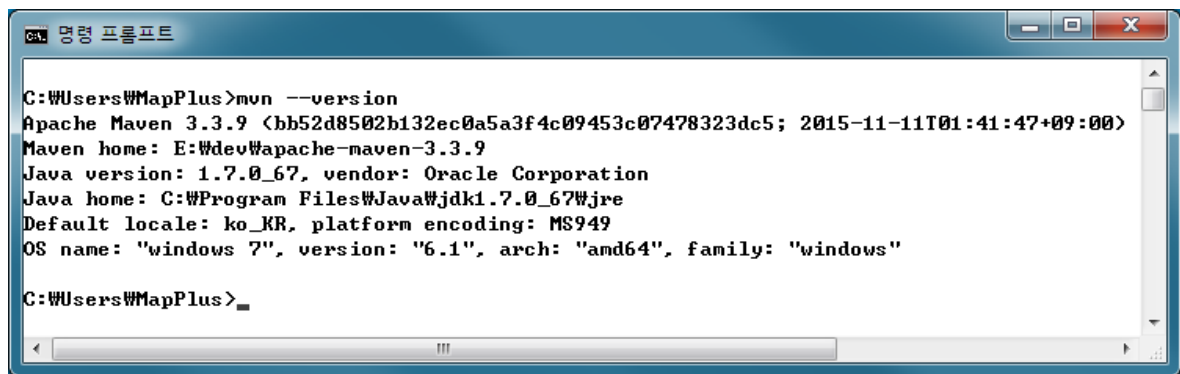
다운로드 한 `apache-maven-3.3.9-bin.zip` 파일을 설치할 위치(다음 예에서는 `e:\dev` 폴더)로 이동하여 압축 해제합니다.



[제어판] → [시스템] → [고급 시스템 설정]으로 이동하여 [시스템 속성] 창의 [고급] 탭에서 [환경변수(N)...] 버튼을 눌러 시스템 변수에 다음 변수를 차례로 등록합니다.

변수 이름	변수 값
M2_HOME	E:\dev\apache-maven-3.3.9
MAVEN_OPTS	-Xms256m -Xmx512m
Path	%M2_HOME%\bin 추가

이제 [명령 프롬프트]를 실행 후 `mvn --version` 또는 `mvn -v` 명령어를 입력하여 설치를 확인합니다.



```

C:\Users\WMapPlus>mvn --version
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T01:41:47+09:00)
Maven home: E:\dev\apache-maven-3.3.9
Java version: 1.7.0_67, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.7.0_67\jre
Default locale: ko_KR, platform encoding: MS949
OS name: 'windows 7', version: '6.1', arch: 'amd64', family: 'windows'

C:\Users\WMapPlus>

```

Maven 의 기본 저장소는 [사용자폴더/.m2/repository] 폴더이며, 이를 변경하고자 할 경우 **M2_HOME/conf/settings.xml** 파일을 열어 **localRepository** 태그를 찾아 다음과 같이 저장할 위치를 변경해주면 됩니다. 이 경로는 반드시 기억해 두시기 바랍니다.

```
<localRepository>E:\dev\W.m2\repository</localRepository>
```

또한 GeoTools 와 GeoServer Mirror 사이트를 다음과 같이 추가해 줍니다.

```

<mirrors>
  <mirror>
    <id>boundlessgeo</id>
    <name>Boundless Cloud Repository</name>
    <url>https://repo.boundlessgeo.com/main</url>
    <mirrorOf>boundless</mirrorOf>
  </mirror>
</mirrors>

```

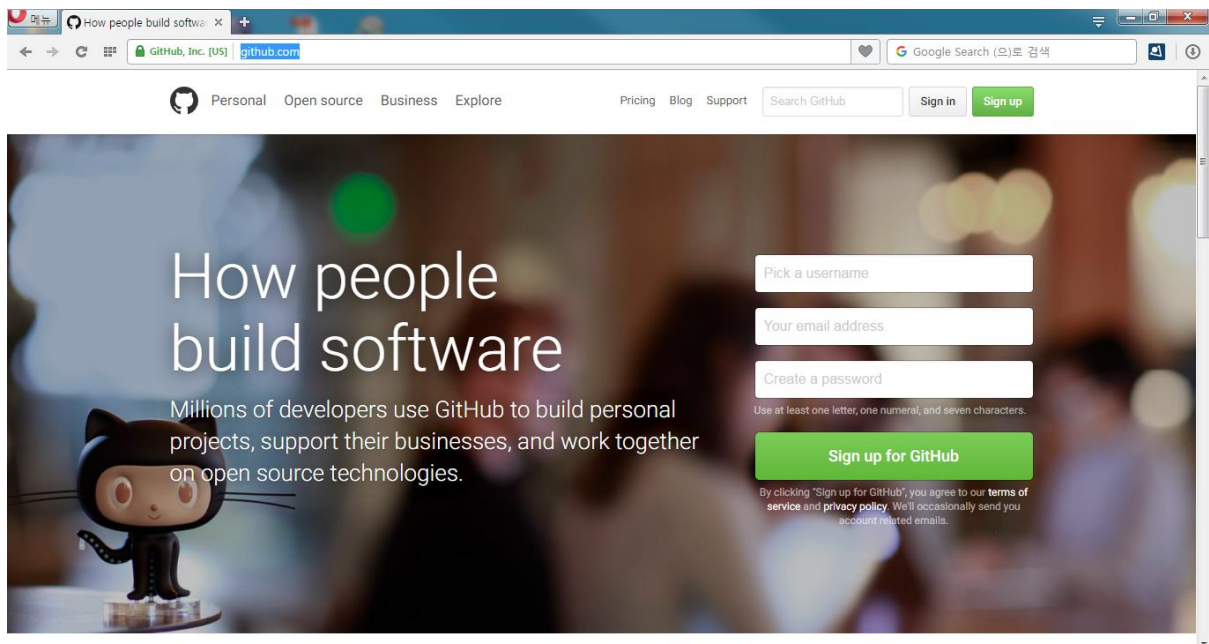
2.1.3. GitHub Desktop 설치

본 프로젝트와 관련된 모든 프로그램의 소스는 GitHub 에 등록되어 관리됩니다. 따라서 GitHub 가입 및 Git 사용이 필수입니다.

2.1.3.1. GitHub 가입

우선 다음의 GitHub 홈페이지로 이동하여 가입을 먼저 합니다.

- <https://github.com>



2.1.3.2. GitHub Desktop 설치

Windows 환경에서 Git 을 편리하게 사용할 수 있는 도구입니다. GUI 환경 또는 Git Shell 환경 모두 사용이 가능합니다.

우선 다음의 GitHub Desktop 사이트로 이동한 후 최신버전의 프로그램을 다운로드 후 설치합니다.

- <https://desktop.github.com>

GitHub Desktop 또는 Git 의 사용법은 이 문서에서는 별도로 다루지 않습니다.

2.2. 기반 소프트웨어 빌드

OpenGDS/Analysis 는 GeoServer 2.8.x 와 uDig 2.0 Latest 버전을 모두 지원하기 위해 다음과 같이 GeoTools 14.x, GeoServer 2.8.x, uDig Master 버전을 함께 사용합니다.

GeoTools 14.x 버전에서 컴파일 된 OpenGDS/Analysis 는 GeoServer 2.8.x 에서도 운영이 가능하며 테스트 되었습니다.

GeoTools	GeoServer	uDig	Java	비고
8.x ~ 10.x	2.4.x	1.5	6	
11.x	2.5.x	2.0 Alpha	7	2014 년
12.x	2.6.x	-	7	
13.x	2.7.x	-	7	2015 년
14.x	2.8.x	2.0 Latest	7	2016 년
15.x	2.9.x	-	8	2017 년 이후
16.x	2.10.x	-	8	2017 년 이후
17.x	2.11.x	-	8	2017 년 이후
18.x	2.12.x	-	8	2017 년 이후

타 버전을 사용하려면 위 버전 테이블을 참조하십시오.

이하 GitHub 사용자는 **user (GitHub 계정)**이며 원 프로젝트의 Fork 를 가정합니다.

2.2.1. GeoTools (선택사항)

다음의 GeoTools GitHub 페이지로 이동하여 Fork 후 소스코드를 로컬 저장소에 내려받습니다.

- <https://github.com/geotools/geotools>

다음은 Git 클론 후 Maven 빌드 및 Eclipse 프로젝트를 생성하는 과정입니다. Git Shell(Bash Shell) 실행 후 다음의 명령어를 단계별로 입력합니다.

```
git clone git@github.com:user/geotools.git
또는
git clone https://github.com/user/geotools

cd geotools
git checkout -b 14.x origin/14.x

mvn install -Dall
또는
mvn -DskipTests install -Dall // skipping tests

mvn eclipse:eclipse -Dall
```

만약 Build 과정에서 오류가 발생하거나 더 많은 옵션을 보려면 다음 사이트를 참조하십시오.

- <http://docs.geotools.org/latest/userguide/build/maven/build.html>

2.2.2. GeoServer (선택사항)

다음의 GeoServer GitHub 페이지로 이동하여 Fork 후 소스코드를 로컬 저장소에 내려받습니다.

- <https://github.com/geoserver/geoserver>

다음은 Git 클론 후 Maven 빌드 및 Eclipse 프로젝트를 생성하는 과정입니다. Git Shell(Bash Shell) 실행 후 다음의 명령어를 단계별로 입력합니다.

```
git clone git@github.com:user/geoserver.git
또는
git clone https://github.com/user/geoserver

cd geoserver
```

```
git checkout -b 2.8.x origin/2.8.x

cd src
mvn install -P allExtensions
또는
mvn -DskipTests install -P allExtensions // skipping tests

mvn eclipse:eclipse -Dall
```

만약 Build 과정에서 오류가 발생하거나 더 많은 옵션을 보려면 다음 사이트를 참조하십시오.

- <http://docs.geoserver.org/stable/en/developer/maven-guide/index.html>

2.2.3. uDig

다음의 uDig GitHub 페이지로 이동하여 Fork 후 소스코드를 로컬 저장소에 내려받습니다.

- <https://github.com/locationtech/udig-platform>

다음은 Git 클론 후 Maven 빌드 및 Eclipse 프로젝트를 생성하는 과정입니다. uDig 프로젝트는 소스코드에 Eclipse 프로젝트가 이미 설정되어 있습니다. Git Shell(Bash Shell) 실행 후 다음의 명령어를 단계별로 입력합니다.

```
git clone git@github.com:user/udig-platform.git
또는
git clone https://github.com/user/udig-platform

cd udig-platform
git checkout master

mvn install -f pom-libs.xml
또는
mvn -DskipTests install -f pom-libs.xml // skipping tests
```

```
mvn clean install -Pproduct -Psdk // Tycho Build
```

만약 Build 과정에서 오류가 발생하거나 더 많은 옵션을 보려면 다음 사이트를 참조하십시오.

- <https://github.com/locationtech/udig-platform>

2.3. OpenGDS/Analysis 빌드

기반 소프트웨어 빌드를 성공했으면 이제 OpenGDS/Analysis 프로젝트를 빌드 합니다.

다음의 OpenGDS/Analysis GitHub 페이지로 이동하여 Fork 후 소스코드를 로컬 저장소에 내려 받습니다.

- https://github.com/mapplus/spatial_statistics_for_geotools_udig

다음은 Git 클론 후 Maven 빌드 및 Eclipse 프로젝트를 생성하는 과정입니다. Git Shell(Bash Shell) 실행 후 다음의 명령어를 단계별로 입력합니다.

```
git clone git@github.com:user/spatial_statistics_for_geotools_udig.git
또는
git clone https://github.com/user/spatial_statistics_for_geotools_udig

cd spatial_statistics_for_geotools_udig
git checkout master
```

OpenGDS/Analysis 프로젝트는 다음의 하위 프로젝트로 구성되어 있습니다.

구분	프로젝트명	설명
GeoTools	gt-process-spatialstatistics	분석 프로세스
GeoServer	gs-wps-spatialstatistics	프로세스 파라미터 입출력
uDig	org.locationtech.udig.processingtoolbox	Processing Toolbox 플러그인
	org.locationtech.udig.processingtoolbox-feature	Processing Toolbox 피쳐
	org.locationtech.udig.processingtoolbox-site	Processing Toolbox 배포

2.3.1. GeoTools Analysis Process

아래의 빌드를 수행하면 target 폴더에 gt-process-spatialstatistics-14.5.jar 파일이 생성됩니다.


```
cd GeoTools\process-spatialstatistics  
mvn clean install
```

2.3.2. GeoServer WPS Extension

아래의 빌드를 수행하면 target 폴더에 gs-wps-spatialstatistics-2.8.5.jar 파일이 생성됩니다.

```
cd GeoServer\gs-wps-spatialstatistics  
mvn clean install
```

2.4. uDig 배포버전 생성 (선택사항)

uDig 과 Tycho 기반의 설치파일을 빌드 하는 과정입니다.

본 프로젝트에서 제공하는 uDig 버전 외 커스터마이징 된 uDig 설치 본을 제공하고자 하는 경우에는 이 장을 참고하십시오.

더 자세한 내용은 다음 사이트를 참고하십시오.

- <https://github.com/locationtech/udig-platform/tree/master/deploy>

2.4.1. NSIS 설치 (선택)

uDig 의 Windows Installer 는 스크립트 기반으로 동작하는 윈도우용 설치 시스템인 NSIS (Nullsoft Scriptable Install System)를 사용합니다. 다음 NSIS 페이지로 이동하여 최신버전(2016 년 6 월 현재 2.51 버전)을 다운로드 합니다.

- <http://nsis.sourceforge.net>

다운로드 한 파일(nsis-2.51-setup.exe)을 설치합니다.



2.4.2. uDig Product Build

Git Shell (Bash Shell) 실행 후 udig-platform 프로젝트 폴더로 이동하여 Tycho Build 를 수행합니다.

```
mvn clean install -Pproduct -Psdk // 코드가 수정된 경우
또는
mvn install -Pproduct -Psdk
```

2.4.3. Deploy

udig-platform 의 deploy 폴더로 이동합니다. Windows 64bit 환경에서 설치파일을 생성하는 경우에는 다음의 파일을 사용합니다.

- win64.sh
- support_functions.sh
- versions.sh

win64.sh 파일을 텍스트편집기에서 불러온 후 아래 내용을 확인합니다.

```
- assemble "win64" "win32.win32.x86_64" "${JRE_WIN64}"
```

versions.sh 파일을 텍스트편집기에서 불러온 후 아래 내용을 수정 또는 확인합니다.

```
- export SERIES=2.0
- export VERSION=2.0.0-SNAPSHOT

- export JRE="${BASE}/jre"
- export JRE_WIN64=jre1.7.0.win64
```

위 두 설정에 의해 최종 생성되는 Windows Installer 는 다음과 같은 이름을 가집니다.

- udig-**2.0.0-SNAPSHOT.win32.win32.x86_64.exe**

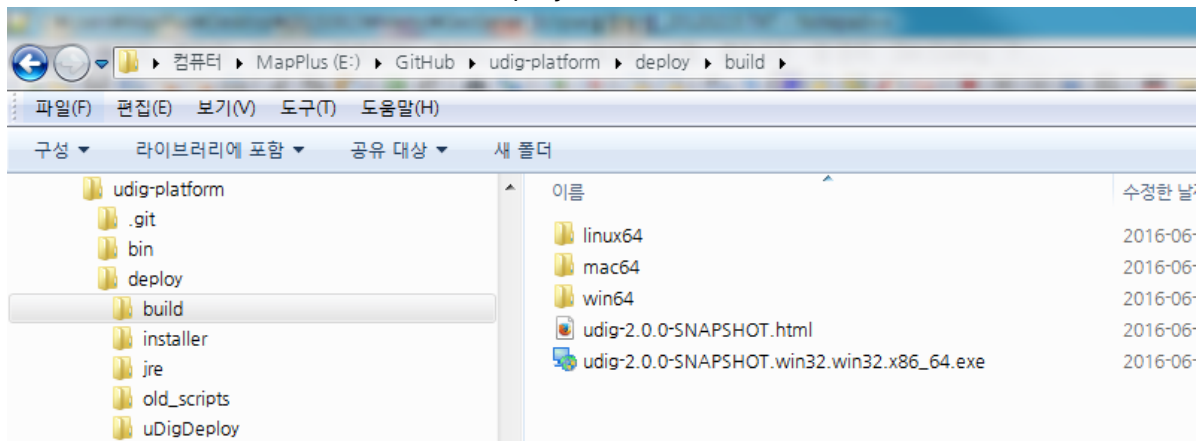
jre 폴더에 다음과 같은 구조의 jre 압축파일(jre1.7.0.win64.zip)을 준비하여 복사해 둡니다.

- jre1.7.0.win64.zip 파일 내에 jre 폴더가 포함되도록 합니다.

마지막으로 Git Shell(Bash Shell)에서 다음을 실행합니다.

```
./clean.sh  
./win64.sh
```

다음과 같이 각 OS 별 설치파일이 deploy 폴더의 build 폴더 내에 생성됩니다.



3. How to Customize

이 장은 OpenGDS/Analysis 프로젝트에 새로운 분석기능을 추가하는 과정을 설명합니다.

3.1. Eclipse 에서 개발환경 설정

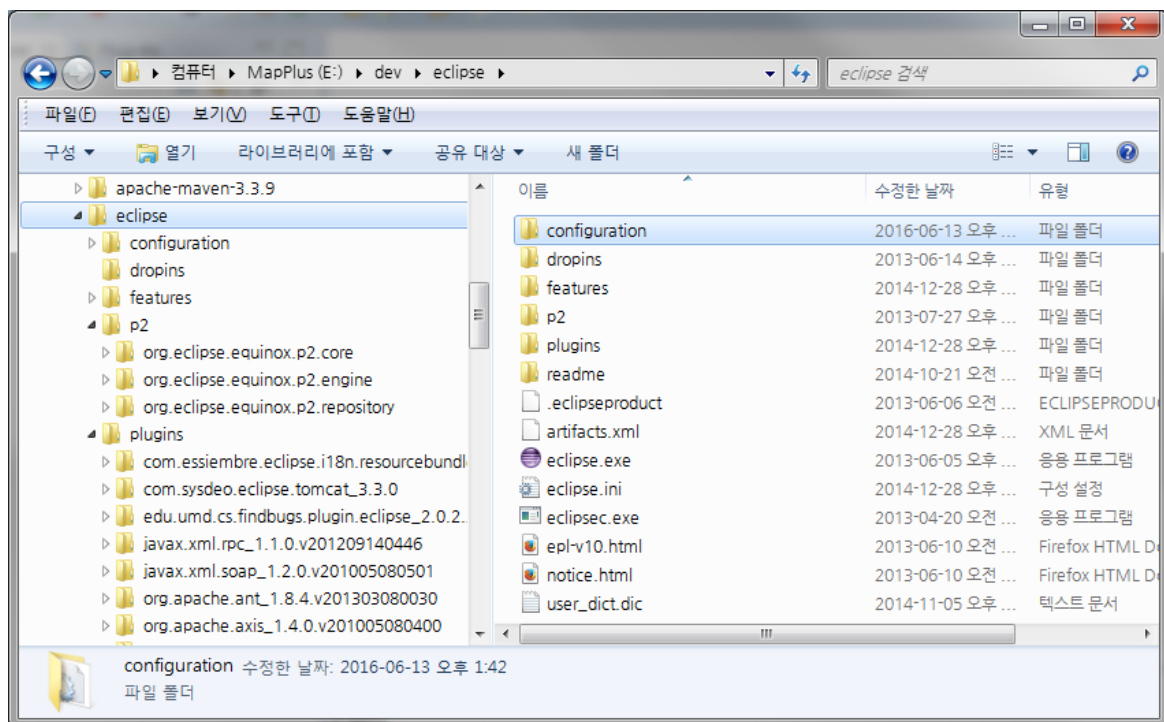
3.1.1. Eclipse IDE 설치

통합 개발 환경인 Eclipse IDE 를 설치하는 과정입니다.

다음의 Eclipse 사이트로 이동하여 4.3 Kepler 또는 4.4 Luna 버전의 [Eclipse Modeling Tools]을 다운로드 합니다. **2016 년 현재 4.4 Luna 버전은 uDig 최신버전과 호환성 문제가 발생할 수 있으므로 4.3 Kepler 사용을 권장합니다.**

- <http://www.eclipse.org/downloads/packages/eclipse-modeling-tools/keplersr2>
- <http://www.eclipse.org/downloads/packages/eclipse-modeling-tools/lunarsr2>

다운로드 한 Eclipse 4.3 Kepler (eclipse-modeling-kepler-SR2-win32-x86_64.) 또는 4.4 Luna (eclipse-modeling-luna-SR2-win32-x86_64.zip) 를 설치할 위치 (다음 예에서는 e:\dev 폴더) 로 이동하여 압축 해제합니다.



Eclipse 설치폴더의 eclipse.ini 파일을 열어서 다음을 추가 또는 수정합니다.

```
-vmargs
-Dosgi.requiredJavaVersion=1.7
-Xms256m
-Xmx1024m
```

3.1.2. 플러그인 설치

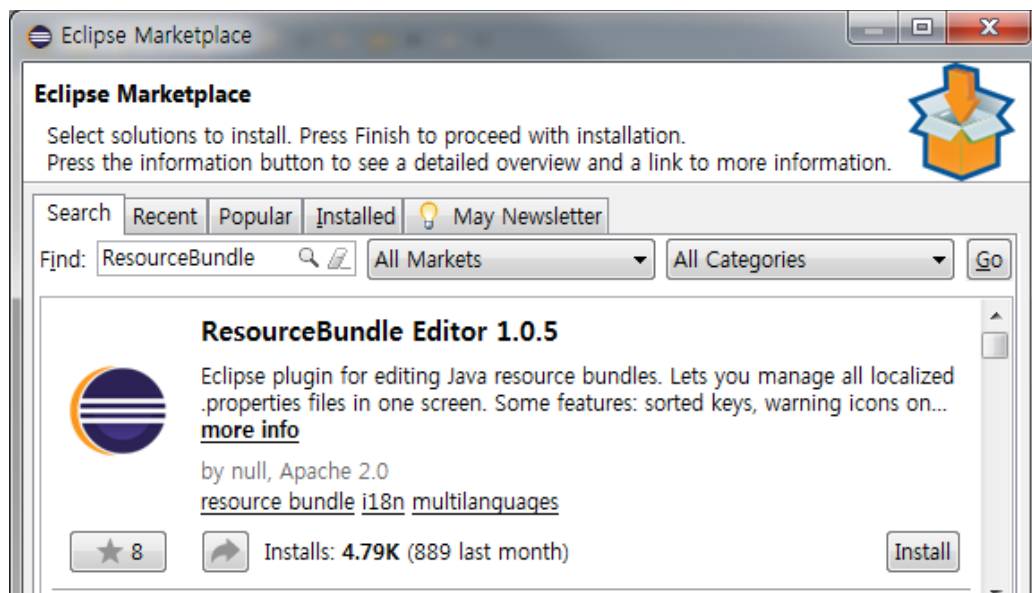
본 프로젝트에서 필수적으로 활용되는 Eclipse 필수 플러그인을 설치하는 과정입니다.

■ ResourceBundle Editor

지역화 (Localization) 리소스 편집기인 ResourceBundle Editor 플러그인을 설치하는 과정입니다.

Eclipse 를 실행한 후 다음과 같이 [Help] → [Eclipse Marketplace...] 메뉴를 실행합니다.

[Search] 탭의 [Find] 입력항목에 ResourceBundle 입력 후 [Go] 버튼을 눌러 검색을 시작합니다. 다음과 같이 ResourceBundle Editor 1.0.5 버전을 확인 후 [Install] 버튼을 눌러 설치를 시작합니다. 설치 완료 후 Eclipse 를 재 시작하면 플러그인 설치가 완료됩니다.



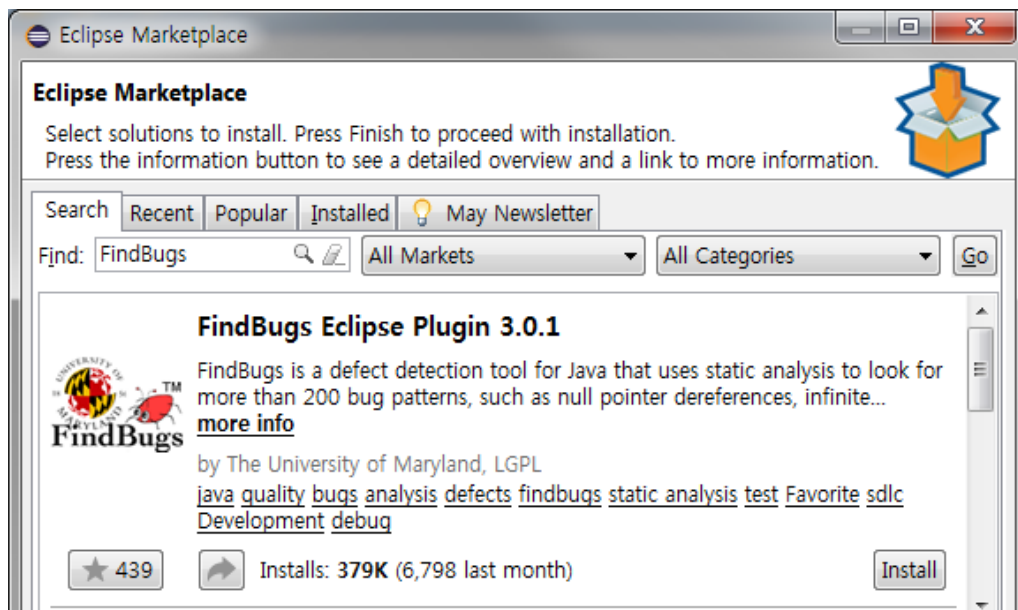
■ FindBugs 플러그인 설치

코드 리뷰 과정에서 많이 활용하는 FindBugs 플러그인을 설치하는 과정입니다.

Eclipse 를 실행한 후 다음과 같이 [Help] → [Eclipse Marketplace...] 메뉴를 실행합니다.

[Search] 탭의 [Find] 입력항목에 FindBugs 입력 후 [Go] 버튼을 눌러 검색을 시작합니다.

다음과 같이 FindBugs Eclipse Plugin 3.0.1 버전을 확인 후 [Install] 버튼을 눌러 설치를 시작합니다. 설치 완료 후 Eclipse 를 재 시작하면 플러그인 설치가 완료됩니다.



■ ObjectAid 설치 (선택사항)

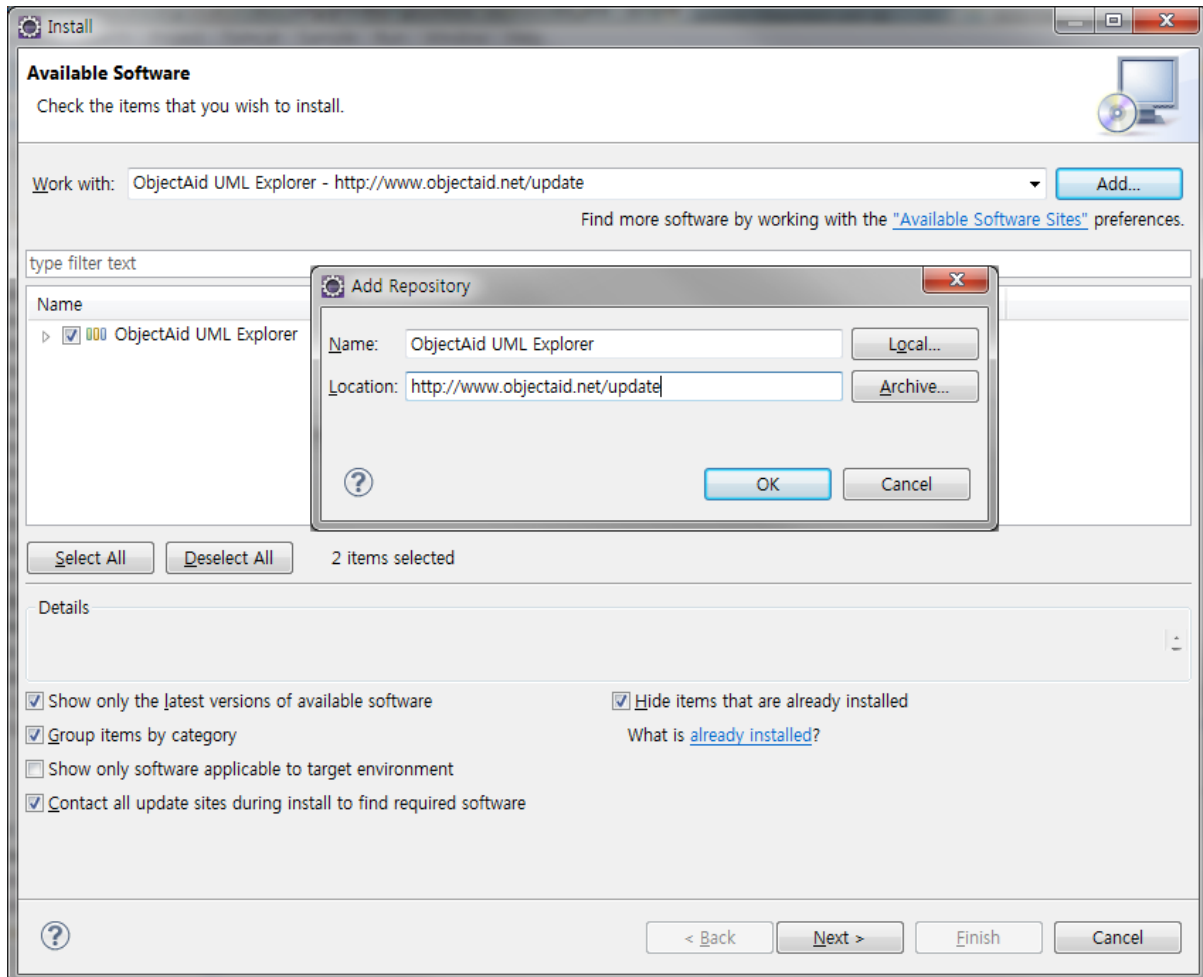
작성한 클래스들의 다이어그램을 문서화하고 시각화하는 ObjectAid UML Explorer⁵ 플러그인 설치하는 과정입니다.

Eclipse 를 실행한 후 다음과 같이 [Help] → [Install New Software...] 메뉴를 실행합니다.

[Add...] 버튼을 눌러 다음과 같이 저장소를 추가 후 ObjectAid UML Explorer 중 ObjectAid Class Diagram 을 설치합니다.

⁵ <http://www.objectaid.com>

Name	Location
ObjectAid UML Explorer	http://www.objectaid.net/update

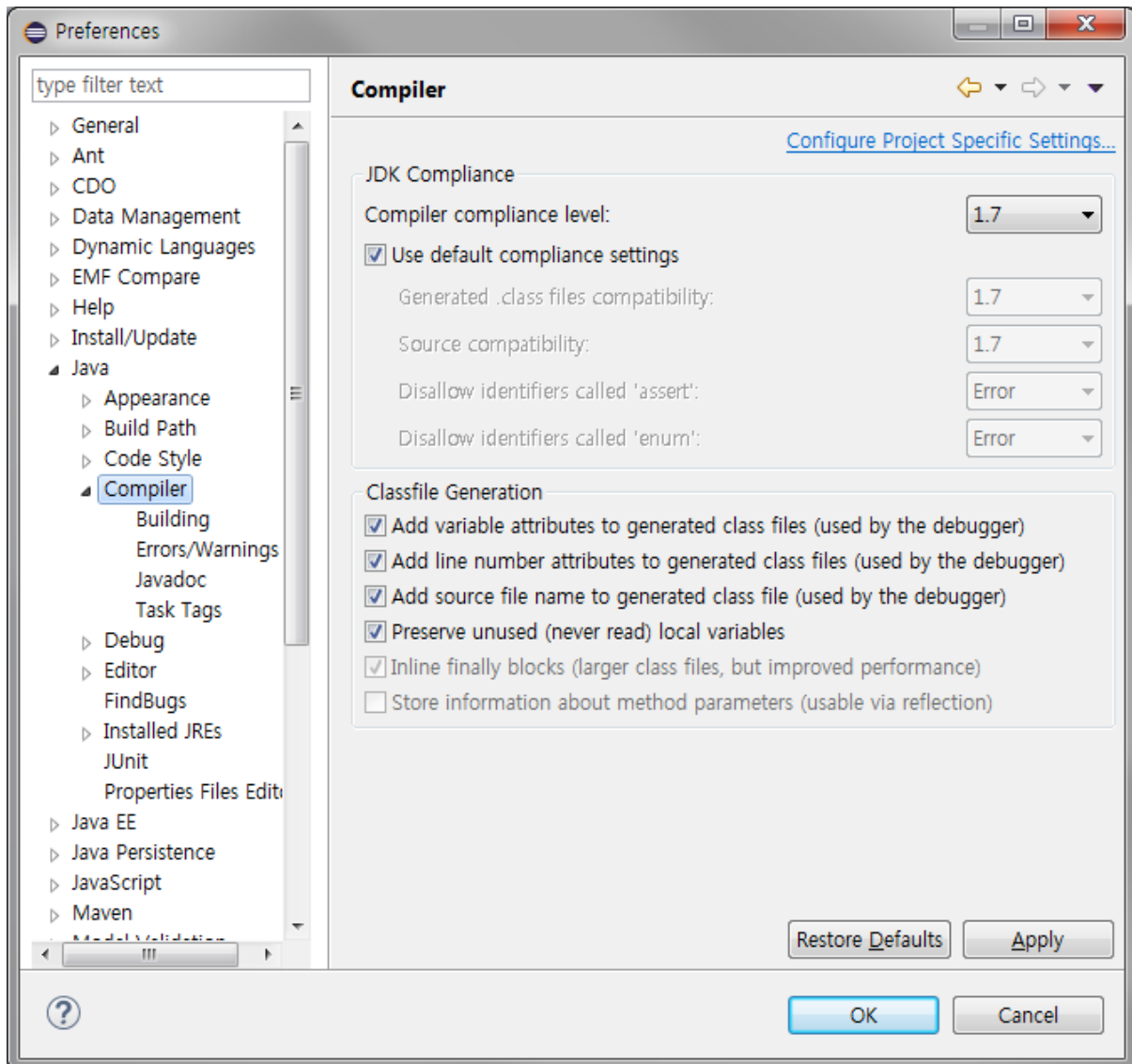


3.1.3. Workspace 개발환경 설정

Eclipse 를 시작하여 개발을 진행할 Workspace 폴더를 설정합니다. 설정하는 Workspace 폴더에 이후 설정하는 모든 환경이 적용됩니다.

■ Java 컴파일러 확인

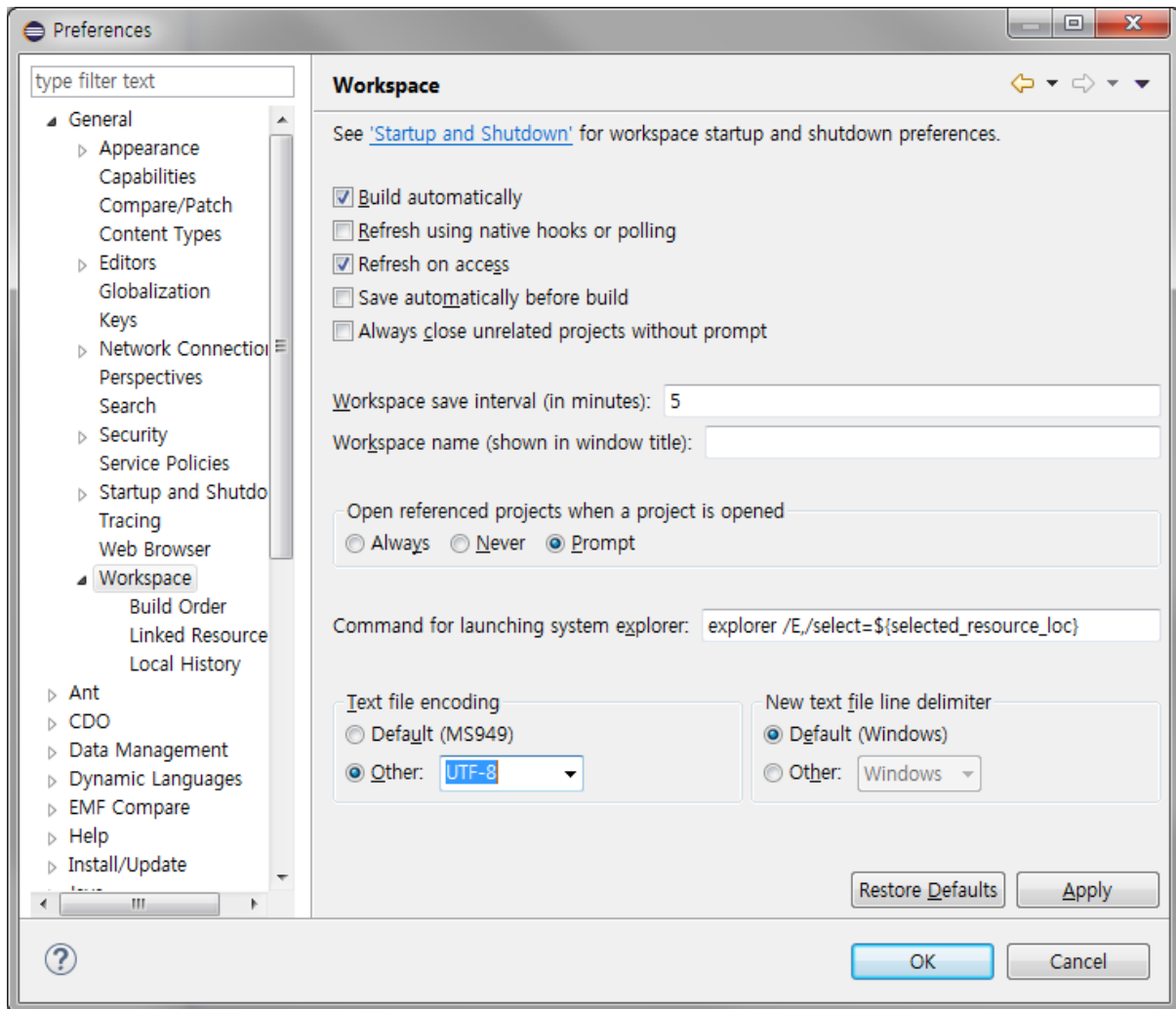
[Window] → [Preferences] 메뉴 실행 후 [Java] → [Compiler] 탭으로 이동합니다. 다음과 같이 Compiler compliance level 이 1.7 로 설정합니다.



■ Workspace 텍스트 파일 인코딩 설정

이 프로젝트에서 사용하는 GeoTools, GeoServer, uDig 등의 소프트웨어 코드는 UTF-8 코드베이스를 사용하므로 다음과 같은 변경을 권장합니다.

우선 [Window] → [Preferences] 메뉴 실행 후 [General] → [Workspace] 탭으로 이동합니다. 다음과 같이 [Text file encoding] 패널에서 [Other] 버튼 체크 후 [UTF-8]을 선택한 후 [Apply] 버튼을 눌러 설정을 저장합니다.



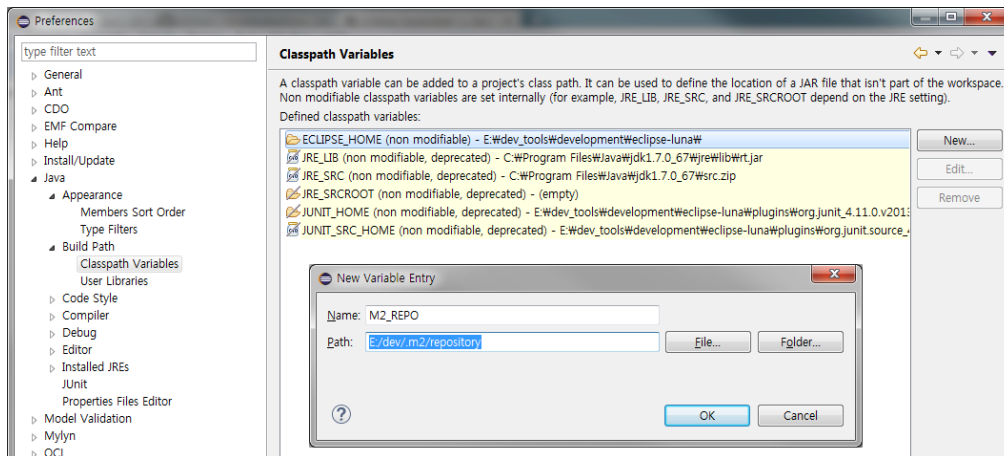
■ Maven M2_REPO 설정

Maven 개발환경을 설정하기 위해 Maven Repository 를 설정하는 과정입니다.

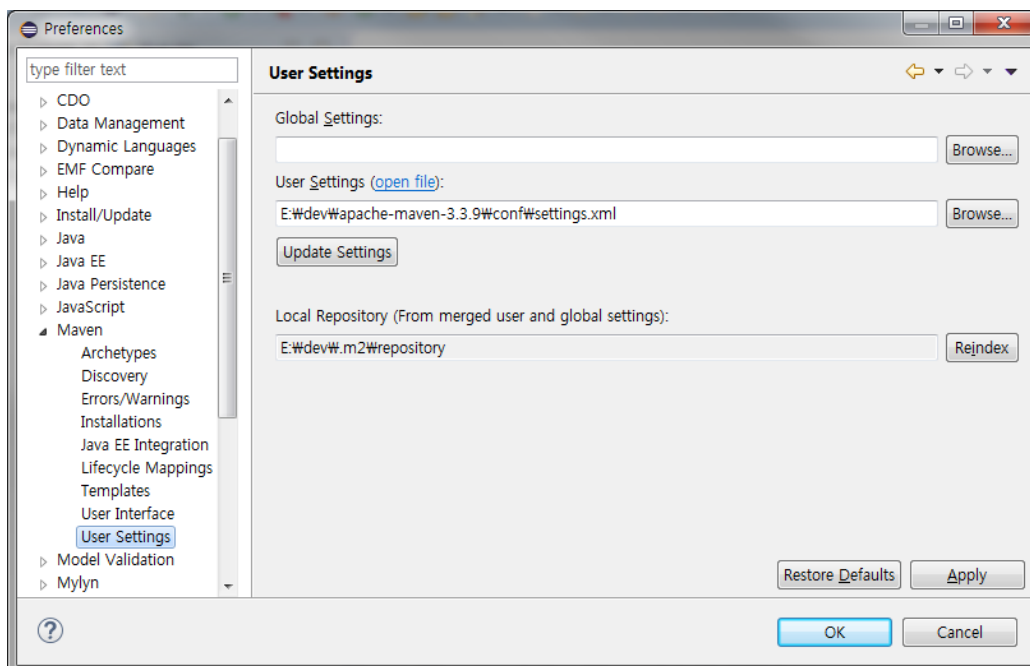
우선 [Window] → [Preferences] 메뉴 실행 후 [Java] → [Build Path] → [Classpath Variables] 탭으로 이동합니다. [New...] 버튼을 눌러 다음과 같이 새로운 변수를 등록합니다.

Name	Path
ME_REPO	C:\Users\사용자명\m2\repository

여기에서 Path 는 [Apache Maven 설치] 과정에서 설정한 Repository⁶ 경로를 말합니다.



만약 이미 M2_REPO 가 설정되어 있고 변경된 새로운 경로를 적용하고자 하는 경우에는 [Maven] → [User Settings] 탭으로 이동합니다. [Update Setting] 항목의 [Browse...] 버튼을 눌러 Maven 설치폴더의 conf 폴더 내 settings.xml 파일 선택 후 [Update Settings] 버튼을 눌러 경로를 변경해 주면 됩니다.

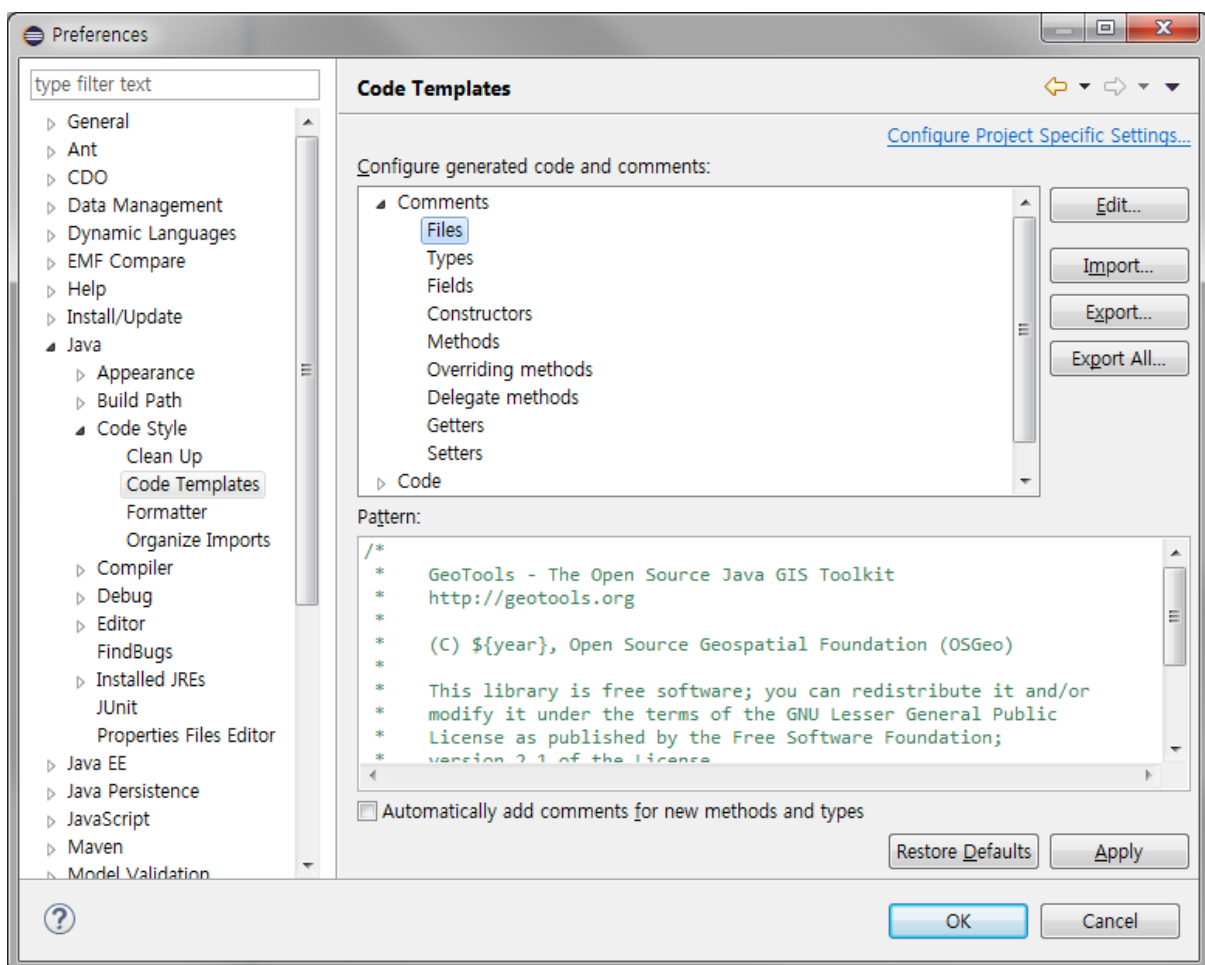


⁶ 최초 설치 시 기본 경로는 C:\Users\사용자명\m2\repository

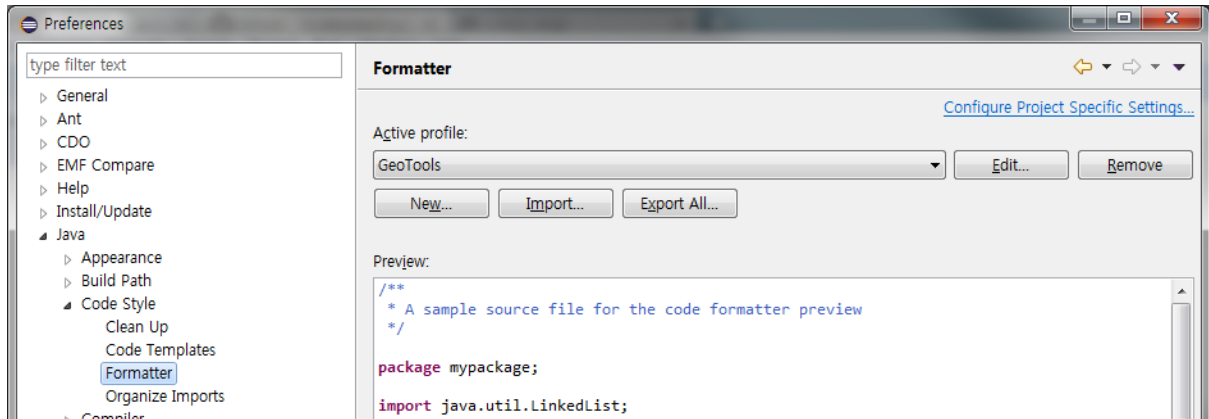
■ 코드 스타일 설정

우선 [Window] → [Preferences] 메뉴를 실행합니다. 이하는 GeoTools 기반 개발에 사용하는 코드 템플릿과 스타일입니다. 만약 uDig의 템플릿과 스타일을 사용하려면 GitHub 로컬 저장소의 [udig-platform\extras\org.locationtech.udig.dev] 폴더 내에 있는 codetemplates.xml(코드 템플릿), codeformatter.xml(코드 포맷) 파일을 사용하면 됩니다.

[Java] → [Code Style] → [Code Templates] 탭으로 이동합니다. [Import...] 버튼을 눌러 GitHub 로컬 저장소의 [geotools\build\weclipse] 폴더 아래에 있는 codetemplates.xml 파일을 선택 후 우 하단의 Apply 버튼을 설정합니다.



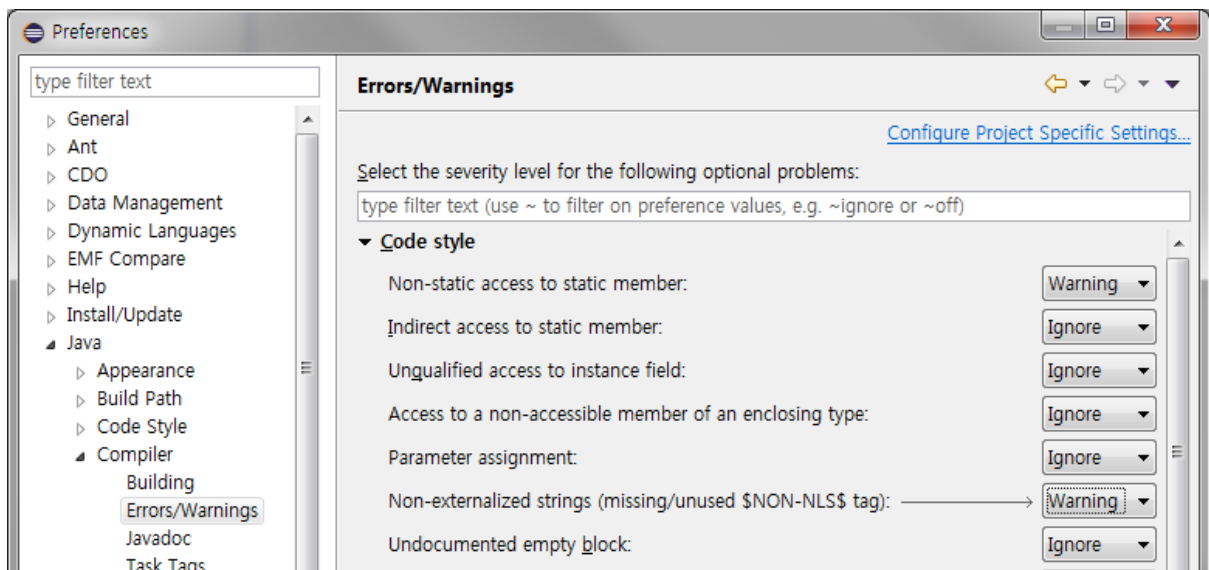
[Java] → [Code Style] → [Formatter] 탭으로 이동합니다. [Import...] 버튼을 눌러 GitHub 로컬 저장소의 [geotools\build\weclipse] 폴더 아래에 있는 formatter.xml 파일을 선택 후 우 하단의 Apply 버튼을 설정합니다.



■ uDig 플러그인 개발 환경설정

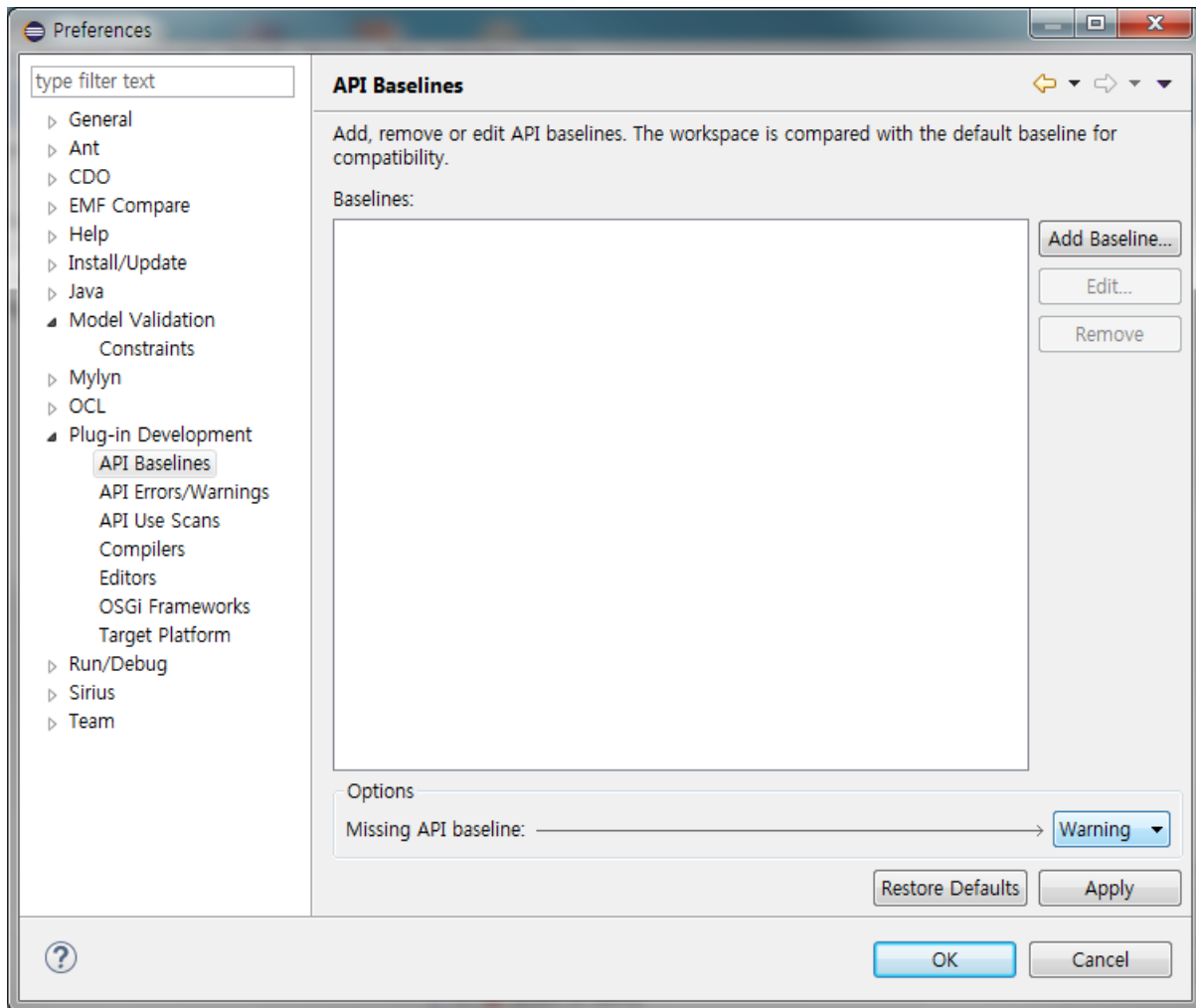
uDig 플러그인 개발에 필요한 환경 설정입니다. 우선 [Window] → [Preferences] 메뉴를 실행합니다.

[Java] → [Compiler] → [Errors/Warnings] 탭으로 이동합니다. [Code Style] 항목을 펼친 후 [Non-externalized strings] 항목을 Warning 으로 변경 후 적용합니다.



[Plug-in Development] → [API Baselines] 탭으로 이동합니다. [Code Style] 항목을 펼친 후 [Non-externalized strings] 항목을 Warning 으로 변경 후 적용합니다.

하단 옵션 패널의 [Missing API baseline] 을 Warning 또는 Ignore 로 변경합니다.



이제 개발할 Workspace 에 대한 환경설정이 완료되었습니다.

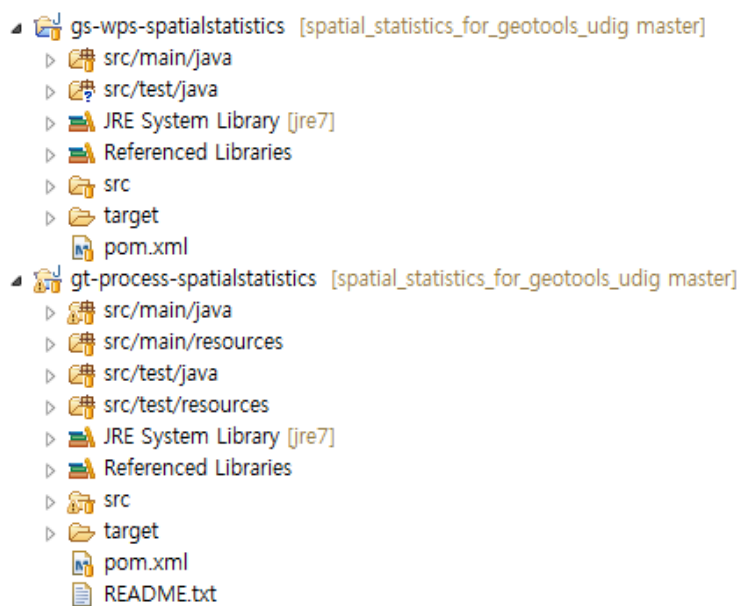
GitHub 의 uDig, GeoTools, GeoServer, OpenGDS/Analysis 프로젝트를 Import 하여 사용하면 됩니다.

3.2. OpenGDS/Analysis: GeoTools, GeoServer 커스터마이징

3.2.1. 프로젝트 불러오기

[File] → [Import...] 메뉴 실행 후 다음과 같이 [Existing Projects into Workspace] 선택 후 [Next] 버튼을 눌러 다음으로 넘어갑니다.

다음과 같이 [Browse...] 버튼을 누른 후 GitHub 로컬 저장소의 [spatial_statistics_for_geotools_udig] 폴더를 선택 후 [gt-process-spatialstatistics], [gs-wps-spatialstatistics] 프로젝트를 선택합니다.

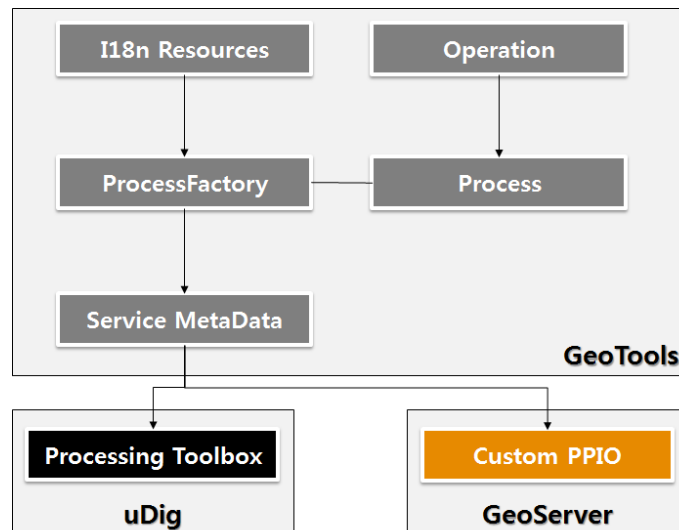


프로젝트의 구성은 다음과 같습니다

구분	프로젝트명	설명
GeoTools	gt-process-spatialstatistics	분석 프로세스
GeoServer	gs-wps-spatialstatistics	프로세스 파라미터 입출력
uDig	org.locationtech.udig.processingtoolbox	Processing Toolbox 플러그인
	org.locationtech.udig.processingtoolbox-feature	Processing Toolbox 피쳐
	org.locationtech.udig.processingtoolbox-site	Processing Toolbox 배포

3.2.2. 프로세스 추가

분석 프로세스는 다음의 과정에 의해 새롭게 추가됩니다.



우선 시작하기 전에 GeoTools 사용자 가이드의 Process 부분을 참조 바랍니다.

- <http://docs.geotools.org/latest/userguide/unsupported/process/index.html>

■ Process 함수 정의

다음과 같이 공간 및 속성조건을 이용하여 벡터 레이어의 피처 수를 계산하는 함수를 예제로 설명합니다. 이미 이 프로세스는 현재 프로젝트에 추가되어 있습니다.

```
CountFeatures(SimpleFeatureCollection inputFeatures, Filter filter): Integer
```

■ ProcessFactory 와 Process 추가

우선 위 함수를 바탕으로 `org.geotools.process.spatialstatistics` 패키지 명으로 `CountFeaturesProcessFactory` 클래스를 생성합니다.

```

/*
 * GeoTools - The Open Source Java GIS Toolkit
 * http://geotools.org
 *
 * (C) 2014, Open Source Geospatial Foundation (OSGeo)
 */

```



```

*   This library is free software; you can redistribute it and/or
*   modify it under the terms of the GNU Lesser General Public
*   License as published by the Free Software Foundation;
*   version 2.1 of the License.
*
*   This library is distributed in the hope that it will be useful,
*   but WITHOUT ANY WARRANTY; without even the implied warranty of
*   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
*   Lesser General Public License for more details.
*/
package org.geotools.process.spatialstatistics;

import java.util.Collections;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.Map;
import java.util.TreeMap;
import java.util.logging.Logger;

import org.geotools.data.Parameter;
import org.geotools.data.simple.SimpleFeatureCollection;
import org.geotools.feature.NameImpl;
import org.geotools.process.Process;
import org.geotools.util.logging.Logging;
import org.opengis.filter.Filter;
import org.opengis.util.InternationalString;

/**
 * CountFeaturesProcessFactory
 *
 * @author Minpa Lee, MangoSystem
 *
 * @source $URL$
 */
public class CountFeaturesProcessFactory extends SpatialStatisticsProcessFactory {
    protected static final Logger LOGGER = Logging.getLogger(CountFeaturesProcessFactory.class);

    private static final String PROCESS_NAME = "CountFeatures";

    /**
     * CountFeatures(SimpleFeatureCollection inputFeatures, Filter filter): Integer
     */

```

```

public CountFeaturesProcessFactory() {
    super(new NameImpl(NAMESPACE, PROCESS_NAME));
}

@Override
public Process create() {
    return new CountFeaturesProcess(this);
}

@Override
public InternationalString getTitle() {
    return getResource("CountFeatures.title");
}

@Override
public InternationalString getDescription() {
    return getResource("CountFeatures.description");
}

/** inputFeatures */
public static final Parameter<SimpleFeatureCollection> inputFeatures = new
Parameter<SimpleFeatureCollection>(
    "inputFeatures", SimpleFeatureCollection.class,
    getResource("CountFeatures.inputFeatures.title"),
    getResource("CountFeatures.inputFeatures.description"), true, 1, 1, null, null);

/** filter */
public static final Parameter<Filter> filter = new Parameter<Filter>("filter", Filter.class,
    getResource("CountFeatures.filter.title"),
    getResource("CountFeatures.filter.description"), false, 0, 1, null, null);

@Override
protected Map<String, Parameter<?>> getParameterInfo() {
    HashMap<String, Parameter<?>> parameterInfo = new LinkedHashMap<String,
Parameter<?>>();
    parameterInfo.put(inputFeatures.key, inputFeatures);
    parameterInfo.put(filter.key, filter);
    return parameterInfo;
}

/** result */
public static final Parameter<Integer> RESULT = new Parameter<Integer>("result",
    Integer.class, getResource("CountFeatures.result.title"),

```

```

        getResource("CountFeatures.result.description"));

    static final Map<String, Parameter<?>> resultInfo = new TreeMap<String, Parameter<?>>();
    static {
        resultInfo.put(RESULT.key, RESULT);
    }

    @Override
    protected Map<String, Parameter<?>> getResultInfo(Map<String, Object> parameters)
        throws IllegalArgumentException {
        return Collections.unmodifiableMap(resultInfo);
    }
}

```

위 소스코드에서 getResource 함수를 사용하는 부분은 i18n 이 적용되는 부분입니다.

이제 CountFeaturesProcessFactory 와 짝을 이루는 CountFeaturesProcess 클래스를 다음과 같이 생성합니다.

```

/*
 * GeoTools - The Open Source Java GIS Toolkit
 * http://geotools.org
 *
 * (C) 2014, Open Source Geospatial Foundation (OSGeo)
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation;
 * version 2.1 of the License.
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 */
package org.geotools.process.spatialstatistics;

import java.util.HashMap;
import java.util.Map;
import java.util.logging.Level;

```

```

import java.util.logging.Logger;

import org.geotools.data.simple.SimpleFeatureCollection;
import org.geotools.process.Process;
import org.geotools.process.ProcessException;
import org.geotools.process.ProcessFactory;
import org.geotools.process.spatialstatistics.core.Params;
import org.geotools.util.logging.Logging;
import org.opengis.filter.Filter;
import org.opengis.util.ProgressListener;

/**
 * Counts the features in the featurecollection
 *
 * @author Minpa Lee, MangoSystem
 *
 * @source $URL$
 */
public class CountFeaturesProcess extends AbstractStatisticsProcess {

    protected static final Logger LOGGER = Logging.getLogger(CountFeaturesProcess.class);

    public CountFeaturesProcess(ProcessFactory factory) {
        super(factory);
    }

    public ProcessFactory getFactory() {
        return factory;
    }

    public static Integer process(SimpleFeatureCollection inputFeatures, Filter filter,
        ProgressListener monitor) {
        Map<String, Object> map = new HashMap<String, Object>();
        map.put(CountFeaturesProcessFactory.inputFeatures.key, inputFeatures);
        map.put(CountFeaturesProcessFactory.filter.key, filter);

        Process process = new CountFeaturesProcess(null);
        Map<String, Object> resultMap;
        try {
            resultMap = process.execute(map, monitor);
            return (Integer) resultMap.get(CountFeaturesProcessFactory.RESULT.key);
        } catch (ProcessException e) {
            LOGGER.log(Level.FINER, e.getMessage(), e);
        }
    }
}

```

```

    }

    return -1;
}

@Override
public Map<String, Object> execute(Map<String, Object> input, ProgressListener monitor)
    throws ProcessException {
    SimpleFeatureCollection inputFeatures = (SimpleFeatureCollection) Params.getValue(
        input, CountFeaturesProcessFactory.inputFeatures, null);
    Filter filter = (Filter) Params.getValue(input, CountFeaturesProcessFactory.filter, null);
    if (inputFeatures == null) {
        throw new NullPointerException("inputFeatures parameters required");
    }

    // start process
    int count = 0;
    if (filter == null) {
        count = inputFeatures.size();
    } else {
        count = inputFeatures.subCollection(filter).size();
    }
    // end process

    Map<String, Object> resultMap = new HashMap<String, Object>();
    resultMap.put(CountFeaturesProcessFactory.RESULT.key, new Integer(count));
    return resultMap;
}
}

```

GeoTools 에서 Process 를 추가하는 자세한 방법은 현재 프로젝트의 소스를 참고하고나 다음의 GeoTools 개발자 매뉴얼을 참고하십시오.

- <http://docs.geotools.org/latest/userguide/unsupported/process/implement.html>

■ 리소스 추가

/src/main/resources 아래의 resource.properties 파일을 불러옵니다.

다음과 같이 CountFeaturesProcessFactory 클래스를 참조하여 Process 함수의 파라미터를 **알파벳 순서로 정의**합니다. 프로세스와 파라미터는 모두 title 과

description 을 가집니다. 이 작업은 ResourceBundle Editor 를 사용하면 한국어 등 다국어 편집이 동시에 가능합니다.

CountFeatures.description	= Computes the number of features in a features.
CountFeatures.filter.description	= The filter to apply.
CountFeatures.filter.title	= Filter
CountFeatures.inputFeatures.description	= Input features to be calculated.
CountFeatures.inputFeatures.title	= Input Features
CountFeatures.result.description	= The number of features.
CountFeatures.result.title	= Number of features
CountFeatures.title	= Count Features

■ 서비스 메타데이터 등록

/src/main/resources/META-INF/services 아래의 org.geotools.process.ProcessFactory 파일을 불러옵니다. 적당한 위치에 다음의 CountFeaturesProcessFactory 를 추가합니다.

```
org.geotools.process.spatialstatistics.CountFeaturesProcessFactory
```

■ 빌드

[OpenGDS/Analysis 빌드](#) - [GeoTools Analysis Process](#) 항목을 참고하십시오.

3.2.3. GeoServer WPS – Custom PPIO 추가

GeoTools 에서 추가한 프로세스의 반환값이 Geometry, FeatureCollection, GridCoverage, Filter, Expressoin, Java Primitive Type 이 아닌 Custom Class 인 경우 GeoServer 에서 서비스되기 위해서는 XML 등 적절한 포맷으로 인코딩하는 과정이 필요합니다.

■ Custom PPIO 추가

다음은 Quadrat Analysis 의 분석결과인 QuadratResult 클래스를 XML 로 인코딩 또는 디코딩하는 PPIO 구현 예입니다.

기타 포맷에 대한 PPIO 확장은 GeoServer 개발자 매뉴얼을 확인하십시오.

```

/* (c) 2014 Open Source Geospatial Foundation - all rights reserved
 * (c) 2016 MangoSystem
 * This code is licensed under the GPL 2.0 license, available at the root
 * application directory.
 */
package org.geoserver.wps.spatialstatistics.ppio;

import java.io.InputStream;

import javax.xml.namespace.QName;

import org.geoserver.config.util.SecureXStream;
import org.geoserver.wps.ppio.XStreamPPIO;
import org.geotools.process.spatialstatistics.pattern.QuadratOperation.QuadratResult;
import org.xml.sax.ContentHandler;

import com.thoughtworks.xstream.io.xml.DomDriver;
import com.thoughtworks.xstream.io.xml.SaxWriter;
import com.thoughtworks.xstream.io.xml.XmlFriendlyNameCoder;
import com.thoughtworks.xstream.mapper.MapperWrapper;

/**
 * A PPIO to generate good looking xml for the Quadrat analysis process results
 *
 * @author Minpa Lee, MangoSystem
 *
 * @source $URL$
 */
public class QuadratResultPPIO extends XStreamPPIO {

```

```

final XmlFriendlyNameCoder nameCoder = new XmlFriendlyNameCoder("__", "_");

static final QName PPIO_NAME = new QName("http://www.opengis.net/statistics", "QuadratAnalysis");

protected QuadratResultPPIO() {
    super(QuadratResult.class, PPIO_NAME);
}

@Override
protected SecureXStream buildXStream() {
    SecureXStream xstream = new SecureXStream(new DomDriver("UTF-8", nameCoder)) {
        @Override
        protected boolean useXStream11XmlFriendlyMapper() {
            return true;
        }

        @Override
        protected MapperWrapper wrapMapper(MapperWrapper next) {
            return new UppercaseTagMapper(next);
        }
    };

    xstream.processAnnotations(QuadratResult.class);
    xstream.alias("QuadratAnalysis", QuadratResult.class);

    return xstream;
}

@Override
public void encode(Object object, ContentHandler handler) throws Exception {
    // bind with the content handler
    SaxWriter writer = new SaxWriter(nameCoder);
    writer.setContentHandler(handler);

    // write out xml
    buildXStream().marshal(object, writer);
}

@Override
public Object decode(InputStream input) throws Exception {
    return buildXStream().fromXML(input);
}
}

```


/src/main/java/ 아래의 applicationContext.xml 파일을 열어 다음과 같이 추가한 PPIO 를 등록합니다. id 는 유일해야 하며 중복을 피하기 위해 클래스 이름을 사용하십시오.

```
<bean id="QuadratResultPPIO"  
class="org.geoserver.wps.spatialstatistics.ppio.QuadratResultPPIO" />
```

■ 빌드

[OpenGDS/Analysis 빌드](#) - [GeoServer WPS Extension](#) 항목을 참고하십시오

3.3. OpenGDS/Analysis: uDig 커스터마이징

2016 년 6 월 현재 Eclipse 에서 uDig 을 실행하려면 4.4 Luna 버전과는 호환되지 않을 수 있습니다. 이에 대한 자세한 내용은 다음 사이트를 참조하십시오.

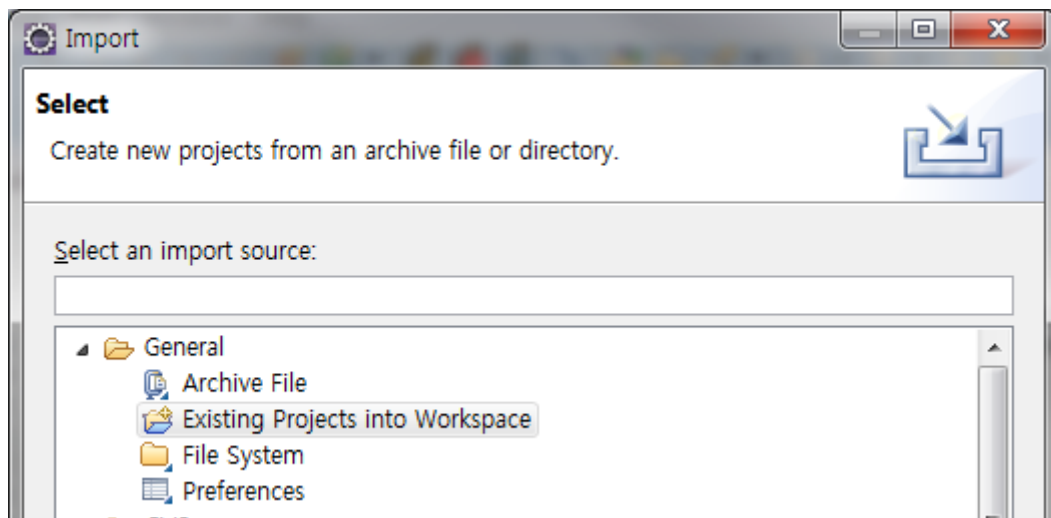
- <https://github.com/locationtech/udig-platform>

3.3.1. uDig Target Platform 설정

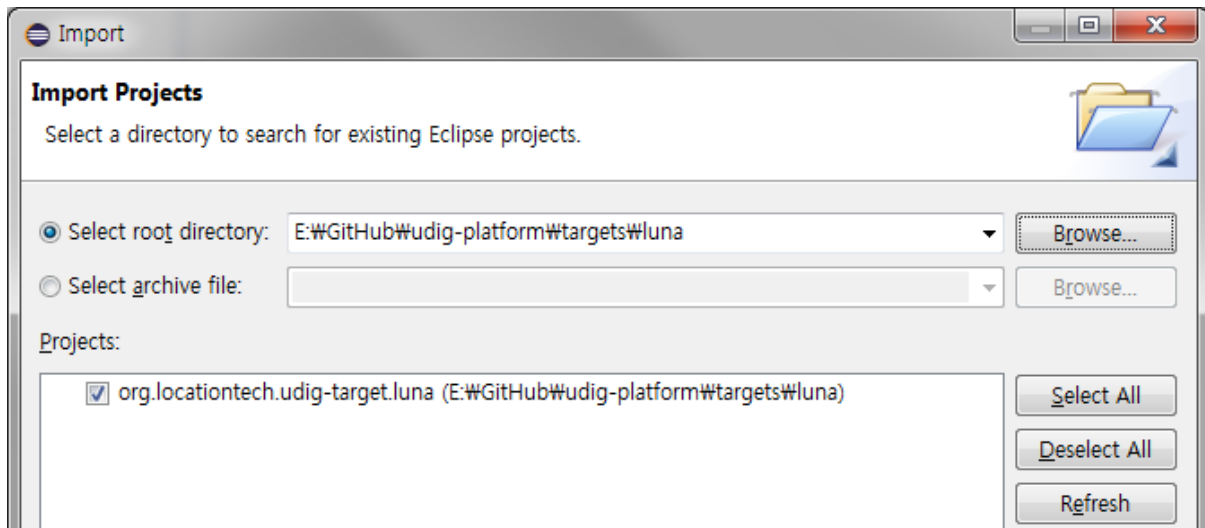
uDig 은 다음 번들을 참조하고 다운로드하기 위해 Target Platform 을 사용합니다:

- Eclipse Rich Client Platform (Indigo 또는 Luna)
- Babel Project (번역 제공)
- Orbit (Eclipse 법률 팀에 의해 확인된 오픈 소스 컴포넌트)

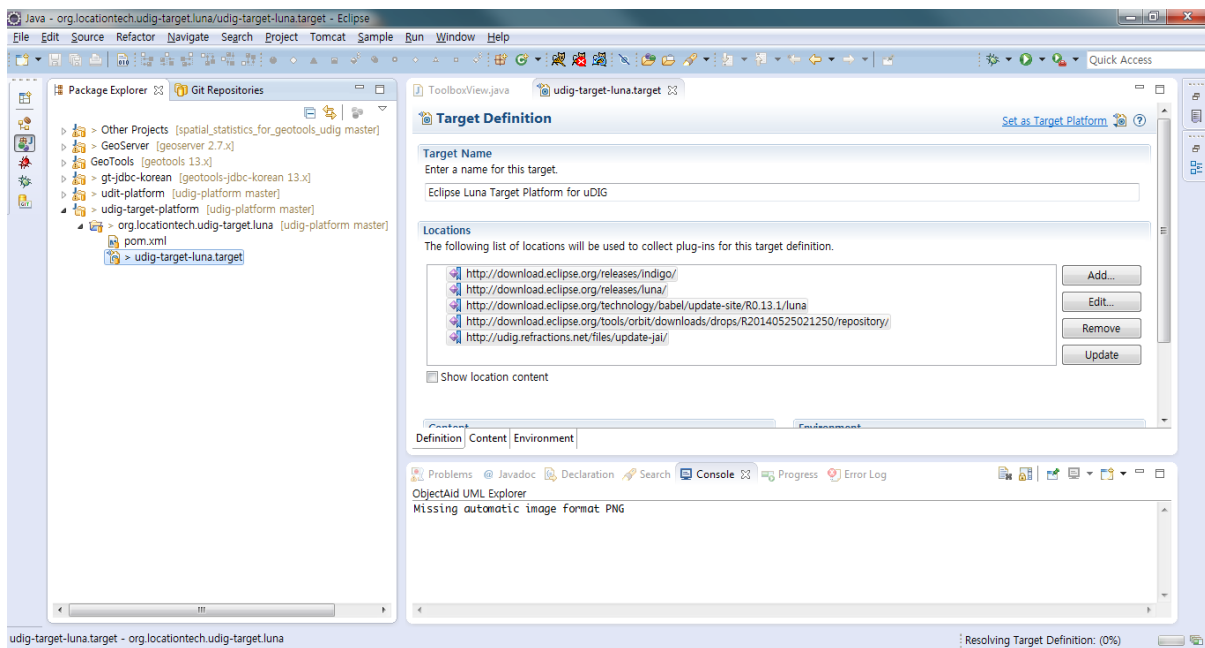
Eclipse 실행 후 [File] → [Import...] 메뉴 실행 후 다음과 같이 [Existing Projects into Workspace] 선택 후 [Next] 버튼을 눌러 다음으로 넘어갑니다.



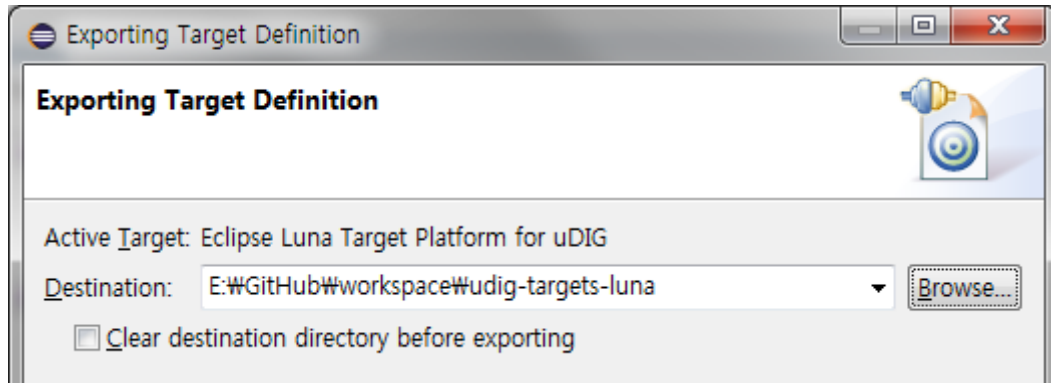
다음과 같이 [Browse...] 버튼을 누른 후 GitHub 로컬 저장소의 [udig-platform\targets\luna] 폴더를 선택합니다. 그림과 같이 [org.locationtech.udig-target.luna] 프로젝트 선택 후 [Finish] 버튼을 눌러 현재 Workspace 로 불러옵니다.



불러온 프로젝트에서 [udig-target-luna.target]을 열고 우 상단의 [Set as Target Platform]을 클릭합니다. 경우에 따라 수분 내지 수십분이 소요될 수 있습니다.



타겟 정의는 온라인 자원을 사용하기 때문에 오프라인 개발 목적을 위해서 로컬에 저장하는 것이 좋습니다. Target Platform 설정이 완료되면 [Set as Target Platform] 링크 오른쪽의 Export (📁) 버튼을 눌러 Target Platform 을 로컬에 저장합니다.




3.3.2. uDig 실행

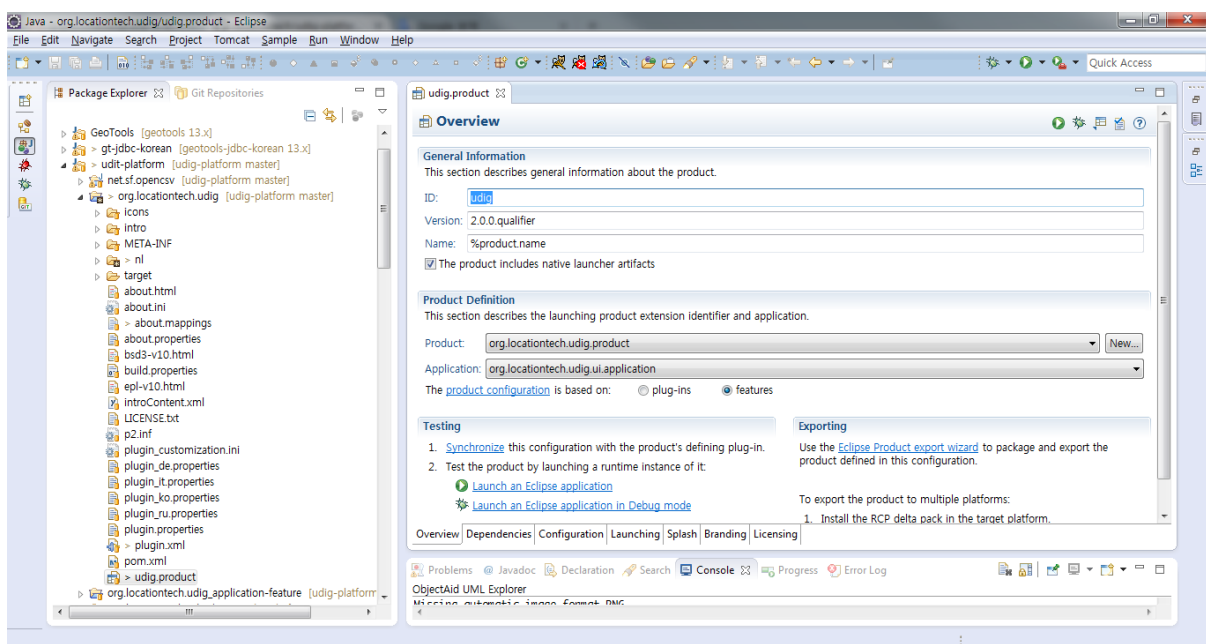
Target Platform 정의가 완료되면 uDig의 나머지 프로젝트들을 불러옵니다.

[File] → [Import...] 메뉴 실행 후 다음과 같이 [Existing Projects into Workspace] 선택 후 [Next] 버튼을 눌러 다음으로 넘어갑니다.

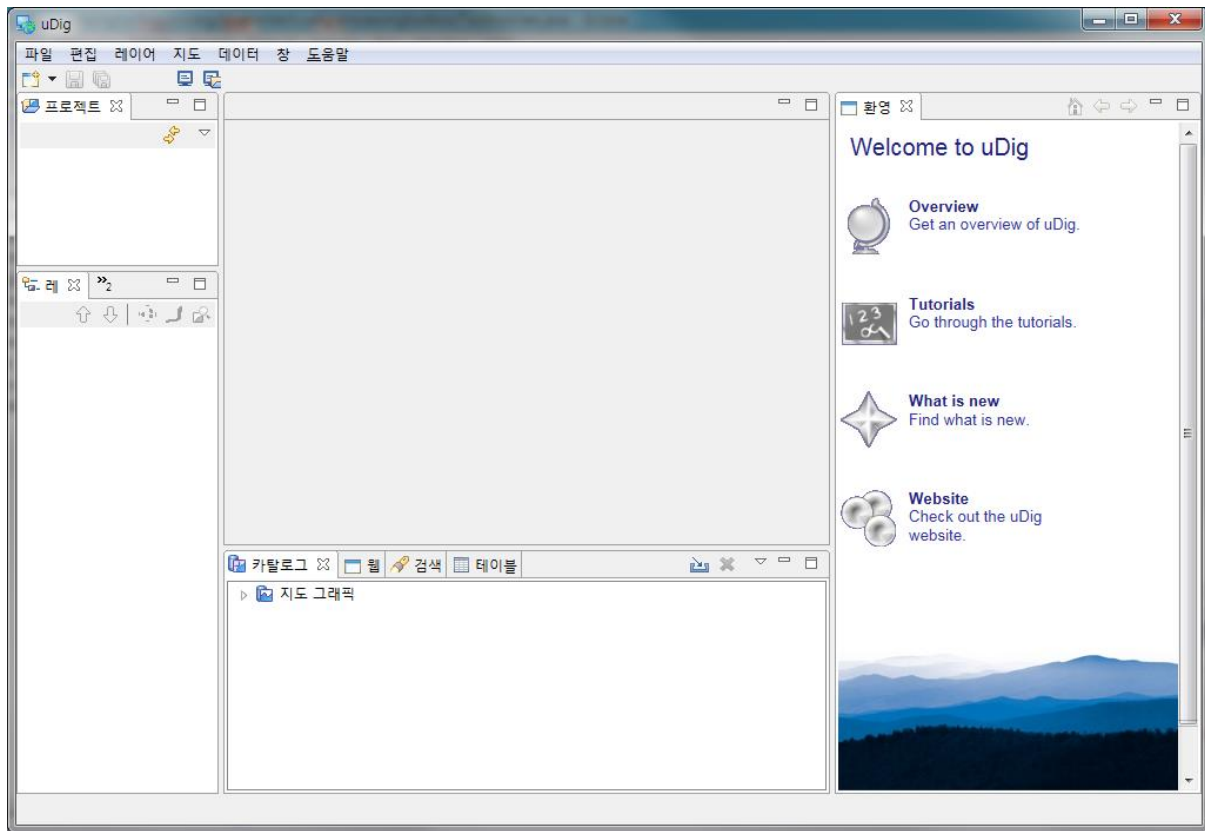
다음과 같이 [Browse...] 버튼을 누른 후 GitHub 로컬 저장소의 [udig-platform] 폴더를 선택 후 [test]와 [tutorials] 문자가 포함되지 않은 모든 프로젝트를 선택합니다.

불러온 프로젝트에서 [org.locationtech.udig] 프로젝트를 선택하고 [udig.product]를 더블 클릭하여 다음과 같이 불러옵니다.

 [Launch an Eclipse application](#) 을 눌러 uDig 을 실행합니다.



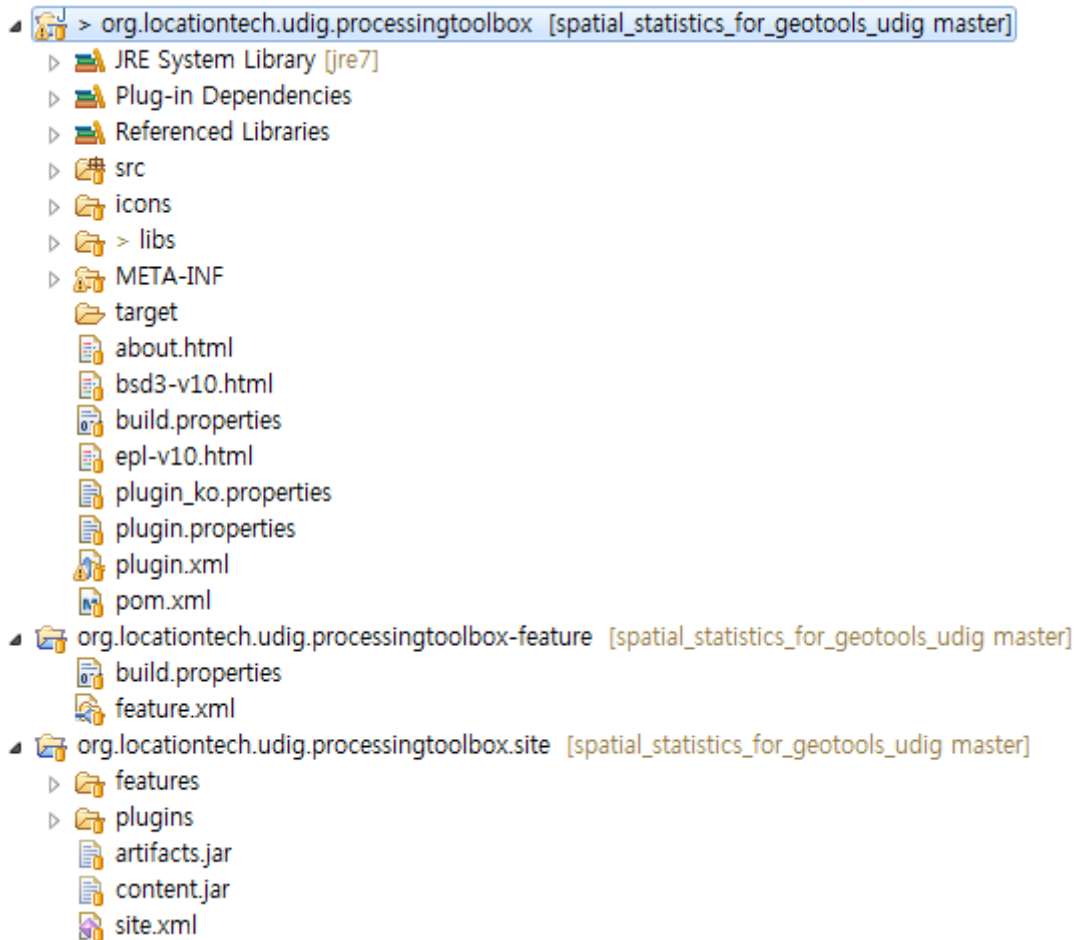
uDig 의 실행화면은 다음과 같습니다.



3.3.3. 프로젝트 불러오기

[File] → [Import...] 메뉴 실행 후 다음과 같이 [Existing Projects into Workspace] 선택 후 [Next] 버튼을 눌러 다음으로 넘어갑니다.

다음과 같이 [Browse...] 버튼을 누른 후 GitHub 로컬 저장소의 [spatial_statistics_for_geotools_udig] 폴더를 선택 후 [org.locationtech.udig.processingtoolbox], [org.locationtech.udig.processingtoolbox-feature], [org.locationtech.udig.processingtoolbox-site] 프로젝트를 선택합니다.



3.3.4. 프로세스 추가

툴박스에 GeoTools 프로젝트에서 추가한 새로운 분석 프로세스를 추가하는 방법입니다.

■ GeoTools Process 파일 갱신

[org.locationtech.udig.processingtoolbox] 프로젝트의 `libs` 폴더에 최신버전의 [gt-process-spatialstatistics-14.5.jar] 파일을 업데이트 합니다.

■ 프로세스 등록

[org.locationtech.udig.processingtoolbox] 프로젝트의 `ToolboxView` 클래스를 불러옵니다.

다음 그림과 같이 새로 추가한 `ProcessFactory` 를 적절한 카테고리에 추가합니다.

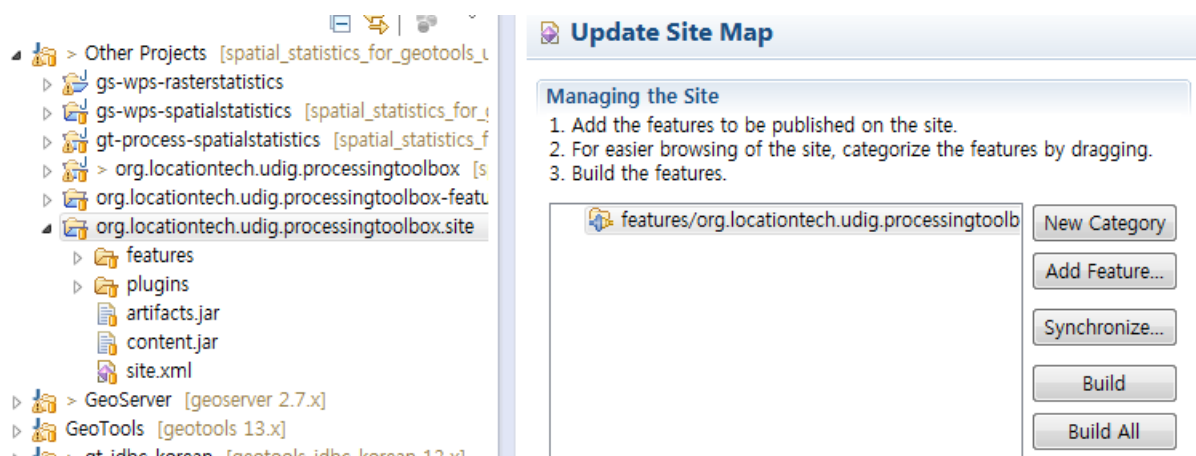
```

329
330 // Spatial Distribution
331 TreeParent distributionTools = new TreeParent(Messages.ToolboxView_Distribution, null, null);
332 ssTools.addChild(distributionTools);
333 buildTool(distributionTools, "org.geotools.process.spatialstatistics.MeanCenterProcessFactory");
334 buildTool(distributionTools, "org.geotools.process.spatialstatistics.MedianCenterProcessFactory");
335 buildTool(distributionTools, "org.geotools.process.spatialstatistics.CentralFeatureFactory");
336 buildTool(distributionTools, "org.geotools.process.spatialstatistics.SDProcessFactory");
337 buildTool(distributionTools, "org.geotools.process.spatialstatistics.SDEProcessFactory");
338 buildTool(distributionTools, "org.geotools.process.spatialstatistics.DirectionaMeanProcessFactory");
339 }

```

3.3.5. 플러그인 빌드 및 배포

[org.locationtech.udig.processingtoolbox.site] 프로젝트의 [site.xml] 파일을 불러옵니다.
다음 그림과 같이 [Build] 버튼을 눌러 배포 버전을 생성합니다.



최신 배포버전은 망고시스템이 관리하며 공식 배포 사이트를 통해 플러그인 업그레이드가 가능합니다.

4. How to Use

GeoServer 와 uDig 배포버전에서 OpenGDS/Analysis 분석기능을 활용하는 방법입니다.

4.1. GeoServer WPS 설치 및 활용

2016 년 현재 GeoServer 2.5.x ~ 2.8.x 를 지원합니다. 다음의 사이트로 이동하여 설치하고자 하는 버전의 GeoServer 및 WPS Extension 을 다운로드 하여 설치합니다.

- <https://sourceforge.net/projects/geoserver>

GeoServer 및 OpenGXT for Server 의 설치와 활용은 다음 문서를 참조하십시오.

- <https://sourceforge.net/projects/mango-spatialstatistics/files/GeoServer/docs>
- OpenGXT for Server v1.0 User Manual.pdf

4.2. uDig 플러그인 설치 및 활용

OS 별 uDig 2.0.0-SHAPSHOT 설치버전은 다음에서 다운로드 할 수 있습니다.

http://www.mangosystem.com:8080/udig/udig-2.0.0-SNAPSHOT.win32.win32.x86_64.exe

http://www.mangosystem.com:8080/udig/udig-2.0.0-SNAPSHOT.win32.win32.x86_64.zip

http://www.mangosystem.com:8080/udig/udig-2.0.0-SNAPSHOT.linux.gtk.x86_64.zip

http://www.mangosystem.com:8080/udig/udig-2.0.0-SNAPSHOT.macosx.cocoa.x86_64.zip

uDig 및 Spatial Toolbox 플러그인의 설치와 활용은 다음 문서를 참조하십시오.

- <https://sourceforge.net/projects/mango-spatialstatistics/files/uDig/docs>
- OpenGXT for Desktop v1.0 User Manual.pdf

5. References

5.1. How to Contribute

OpenGXT for Server 개발에 참여하는 방법은

- GitHub 를 이용하여 직접 개발에 참여
- Transifex 를 이용하여 지역화(한국어 등)에 참여하는 방법이 있습니다.

다음 URL 을 참고하십시오.

■ *GitHub*

- https://github.com/mapplus/spatial_statistics_for_geotools_udig

■ *Transifex*

- <https://www.transifex.com/mangosystem/ss-rd>

5.2. Reference

- <http://www.opengeospatial.org/standards/wps>
- <http://docs.geoserver.org/stable/en/user/extensions/wps/index.html>
- https://github.com/mapplus/spatial_statistics_for_geotools_udig

[문서의 끝]



OpenGXT Developer Guide