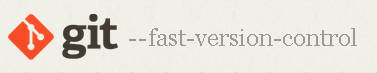
**<TIZEN DEVEOPMENT GUIDE>**

**목차**

1. TIZEN Develpoment working Mechnism
2. TIZEN IDE / Development Environment Setting
3. TIZEN SPECIFIC SOURCE DOWNLOAD (Windows)
4. TIZEN SPECIFIC SOURCE DOWNLOAD (LINUX)
5. TIZEN ALL SOURCE DOWNLOAD (Windows)
6. TIZEN ALL SOURCE DOWNLOAD (LINUX)
7. Building Packages Locally with GBS (LINUX)
8. TIZEN Reference phone Setting
9. TIZEN IMAGE PORTING
10. Creating Tizen Images with MIC
11. TIZEN PROJECT CONTRIBUTE
12. **TIZEN Develpoment working Mechnism**
13. SCM ( Source Code Management)

오픈소스 기반의 타이젠의 SCM는 두 가지 파트를 가지고 있다.

1. Git



URL : <http://www.git-scm.com>

Git은 강력하고 유연하고 오버헤드가 적은 version control system 이다.

협동하며 일하는 개발자들에게 효율적이고 탄탄하다는 장점이 있다.

1. Gerrit

Gerrit은 웹 기반의 code review system 이다.

Git을 활용해서 만들어진 project들을 review 할 수 있다.

1. OBS (Open Build Service)

OBS 시스템은 Linux 기반의 개발자들이 오픈소스 소프트웨어를 손쉽게 만들고 릴리즈 하는 것을 지원하는 시스템이다. Linux 기반이면 다양한 하드웨어 아키텍쳐에서도 다 쓸 수 있도록 지원한다.

1. TIZEN Open Source Working Mechanism

Git repository는 Project에게 2가지의 branch를 가질 수 있게 해준다.

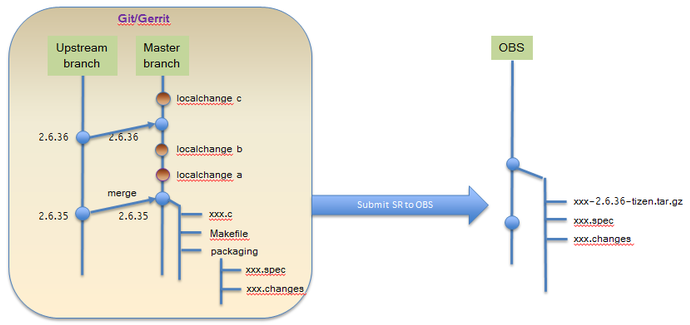
1. Master Branch (필수)

프로젝트의 가장 상위 branch 이다.

1. Upstream Branch (선택)

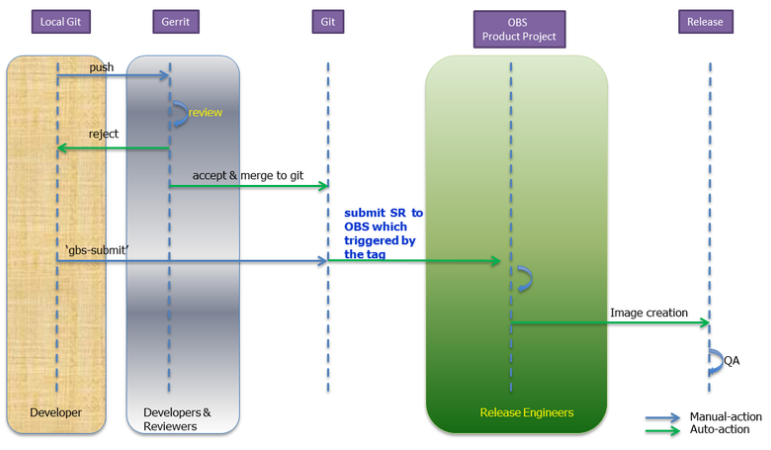
새로 만들어지는 프로젝트들이 Upstream Branch에 올라가면 나중에 Master Branch 로 최종적으로 올라가는 구조이다.

1. Package Generation at the Backend



위의 그림은 소스코드 저장소(repository)에서 OBS으로 submit하는 구조이다.

1. 타이젠 package development workflow



1. 개발자들이 개발환경을 구축하고 개발툴을 설치한다.
2. 개발자들이 소스코드를 clone하고 local build를 통해 local 검증을 한다.
3. 개발자들이 패치들을 Gerrit에 올린다.
4. 타이젠 back-end 서비스와 reviewer들이 개발자가 올린 패치를 검증하게 된다. 검증은 automated testing, manual testing, respectively testing을 통해서 하게 된다. 그리고 패치의 quality에 따라 -1, 0, +1 로 투표를 한다.
5. automated testing :
6. Manual testing :
7. Maintainer들은 개발자가 submit한 패치들을 검증한 다음 승인하게 된다. 그 후에 코드들을 merge 함으로서 it 저장소를 바꾸게 된다.
8. Maintainer/Developer 들은 gbs submit(Linux 터미널에서) 명령을 통해서 OBS에 패키지를 submit 한다.
9. 타이젠 back-end service는 pre-release와 normal release 프로세스를 동시에 실행한다. pre-release 프로세스 에서는 특정 패키지들을 포함하는 타이젠 image들을 만들고 다른 여러 개발자들이 review 할 수 있도록 submit 한다.
10. 7번의 작업이 accpet이 되면 소스코드들은 OBS 저장소에 최종 통합되게 된다. Tizen image로 publish 되어 release되는 과정을 거칠 수 있게 되다.
11. Tizen Roles and Responsibilities
12. Developer들의 책임
13. 코드를 작성하고 Git의 개발 branch에 submit 한다.
14. 또는 이미 제출된 코드를 테스트하고 검증하고 -1 또는 +1점을 부여하여 review 한다.
15. Maintainer들의 책임
16. upstream, development branch와 같은 부가적인 branch 만들기
17. upstream branch를 master branch로 재조정하기
18. 코드를 review 하고 패치들에게 -2,+2점을 부여하여 승인하기
19. Reviewer들의 책임
20. 제출된 코드들을 review하고 +2, -2점을 부여한다.
21. Release Engineer들의 책임
22. OBS에 제출됨을 승인한다.
23. 결과로 나온 image를 smoke-test (본격적인 테스트의 수행에 앞서, 제품의 빌드가 구축된 테스트 환경에서 테스트가 가능한지 여부를 판단하기 위해 주요 모듈이나 시스템을 간단하게 테스트 하는 것) 과 QA 엔지니어가 테스트를 실행할 수 있도록 release area를 변화시킨다.
24. QA Engineer들의 책임
25. 퇴행과 버그를 제거하기 위해 image들을 통합하고 검증하는 작업을 실행한다.
26. **TIZEN IDE / Development Environment (Linux only)**
27. Ubuntu 에서 개발 툴 설치하기 (Ubuntu terminal 사용하면 된다)
28. text editor 를 이용해서 소스 리스트 열기 (vim 사용)

**$ sudo vim /etc/apt/sources.list**

1. source list에 타이젠 툴 저장소 넣기

**$ deb http://download.tizen.org/tools/latest-release/Ubuntu\_12.10 /**

마지막 / 와 URL 사이의 빈칸을 조심해야 한다.

1. 패키지 파일들과 source list 재동기화

**$ sudo apt –get update**

1. 개발 툴, GBS, MIC 설치하기

**$ sudo apt –get install <Package\_Name>**

**$ sudo apt –get install gbs mic**

1. 개발 툴 업그레이드

**$ sudo apt –get install <Package\_Name>**

아니면,

**$ sudo apt –get update**

**$ sudo apt –get upgrade**

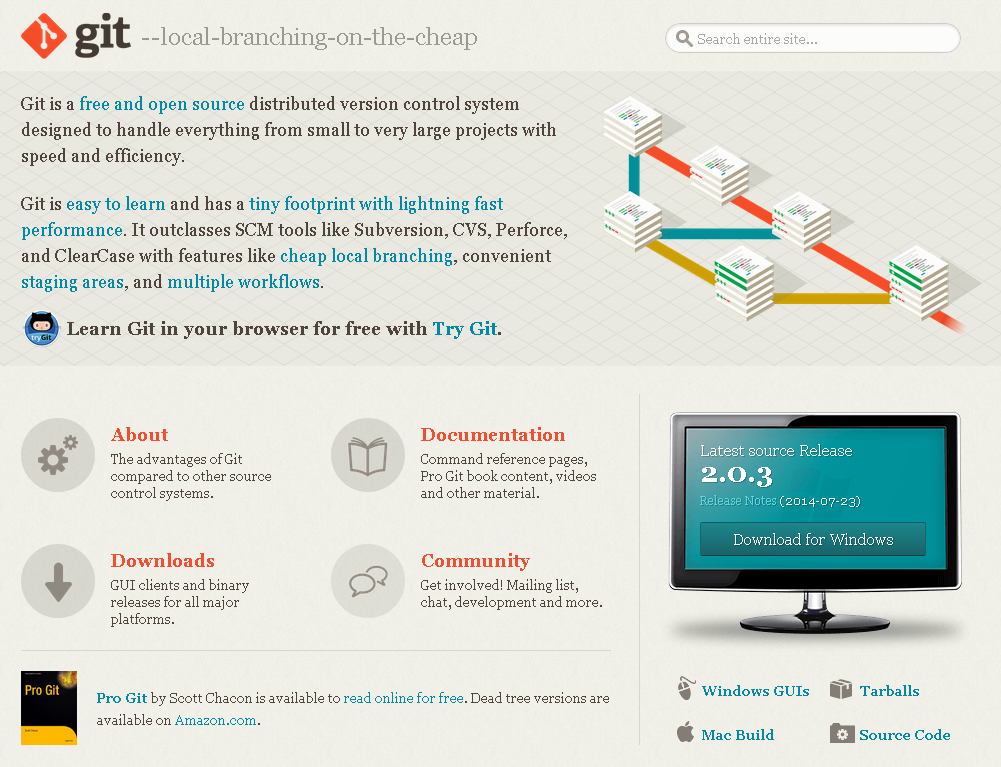
1. **TIZEN OPEN SOURCE DOWNLOAD (Windows 환경)**
2. TIZEN 사용자 계정 만들기

TIZEN SOURCE를 다운로드 받기에 앞서 TIZEN 개발자 홈페이지에 사용자 계정을 생성해야 한다.



1. GIT 설치

URL : <http://git-scm.com>



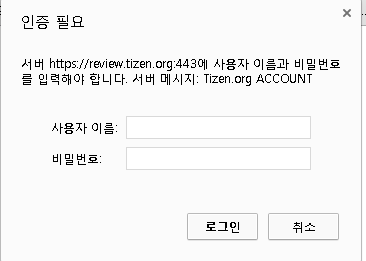
위의 표시부분에서 Download for Windows 를 설치한다.

GIT BASH, GIT HUB 가 설치되는 것을 확인 할 수 있다.

앞으로 GIT BASH에서 TIZEN 코드를 다운로드 받거나 Gerrit을 사용할 때 편리하게 작업할 수 있다.

1. TIZEN Gerrit 사용자 인증

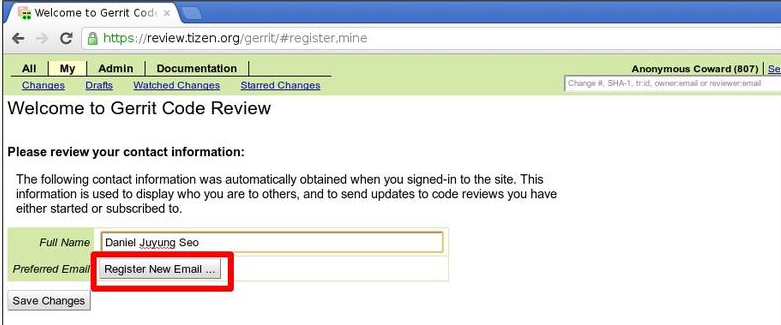
URL : <http://review.tizen.org/gerrit>



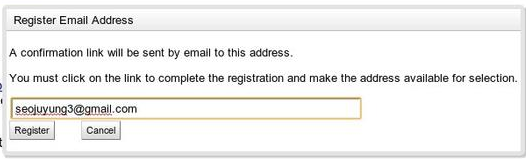
위의 인증 시에는 TIZEN 홈페이지에서 만든 사용자 계정으로 로그인하면 된다.

1. Gerrit 이름과 이메일 등록

TIZEN Gerrit 사이트에 인증 후, 사용자 이름과 이메일을 등록한다.



Full Name 부분을 제대로 기입하고 나서, Register New Email… 버튼을 눌러 이메일 주소를 등록한다.



Register 버튼을 눌러 사용자 정보를 입력 완료한다.

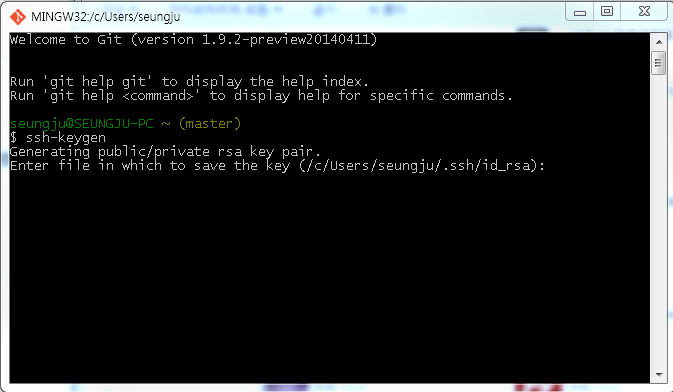


올바르게 사용자 정보를 등록하고 나면, 사이트 화면 우측상단에 자신의 이름과 이메일 주소가 확인된다.

※Settings 의 위치는 저 곳에 있다.

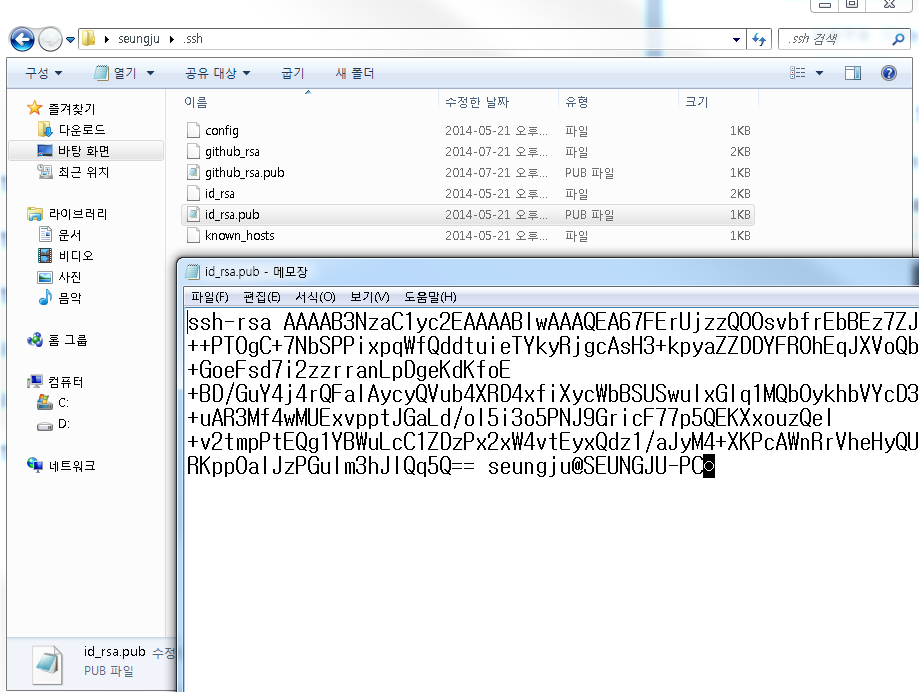
1. Public Key 등록

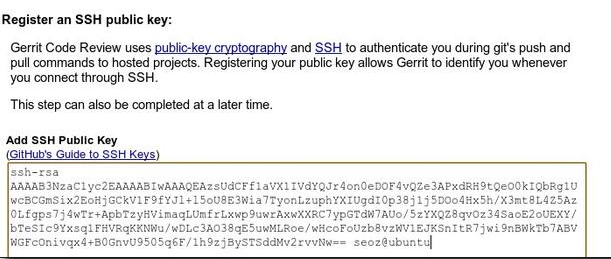
GIT BASH 에서 $ ssh-keygen 을 입력하면 자동으로 public key가 등록된다.



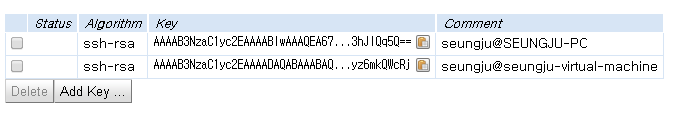
public key가 생성되고 나면 파일을 열어 안의 내용을 복사한 다음,

아래 그림처럼 TIZEN Gerrit 사이트 화면 Add SSH Public Key 칸에 붙여 넣기 하면 된다.

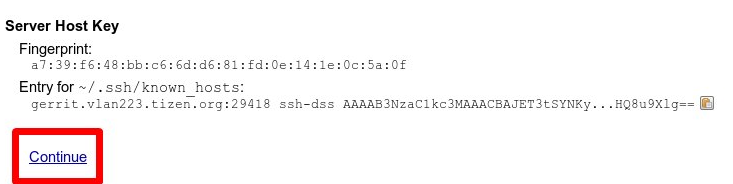




등록이 되면 아래와 같이 확인 가능하다.

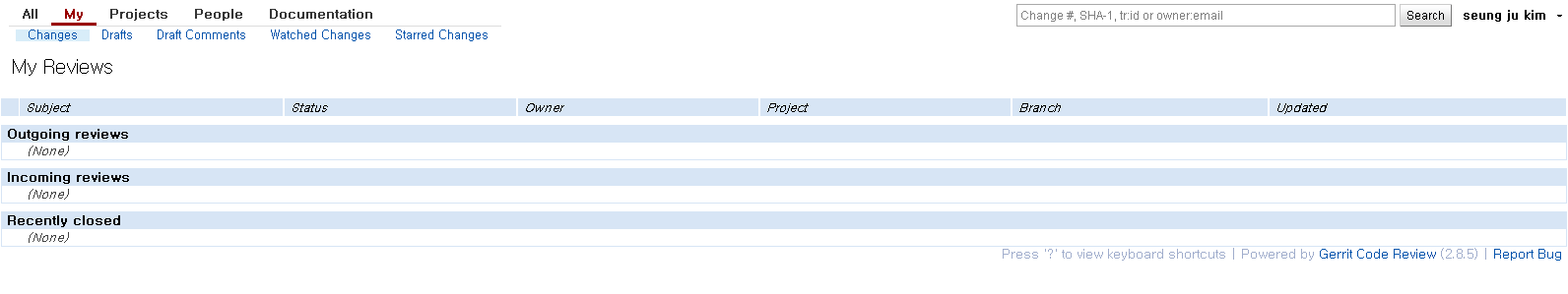


SSH Public Key 등록이 완료 되면 사이트 화면 하단에 위치한 CONTINUE 글자를 눌러서 다음 페이지로 넘어간다.



1. Gerrit 사이트 사용자 등록 완료

사용자 이름, 이메일 주소, Public Key 생성과 입력을 완료하면 Gerrit 사이트를 사용하여 타이젠 소스코드를 다운로드 받을 수 있다.

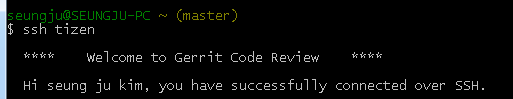


1. SSH connection 확인하기

Git Bash 에서

**$ ssh tizen**

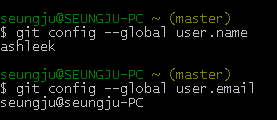
아래와 같은 화면이 떠야 한다.



1. Gerrit Access를 위해 Git 에 이름과 이메일 등록하기

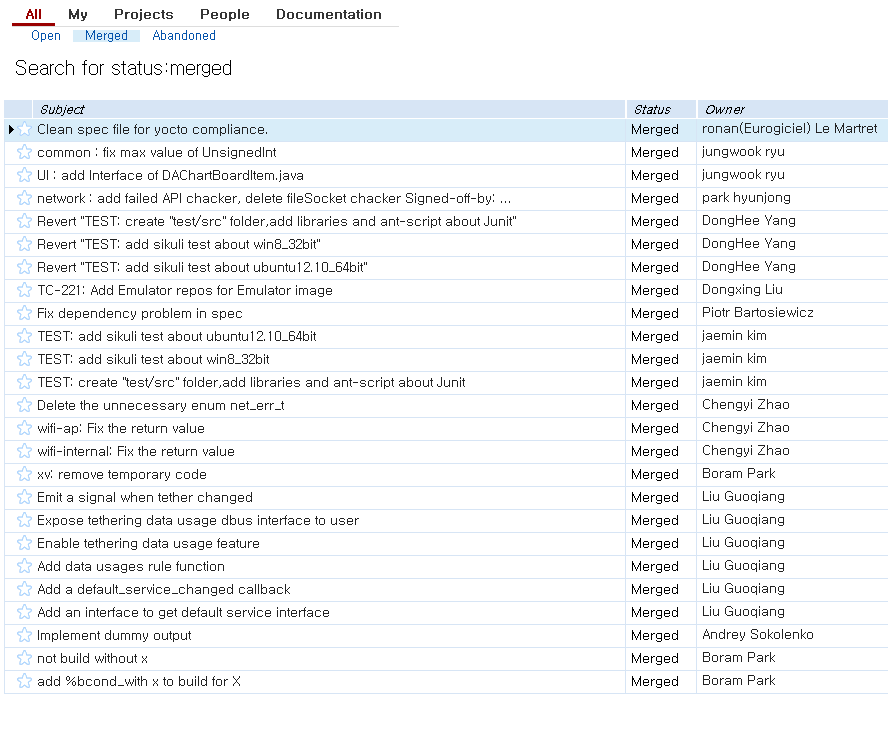
**$ git config –global user.name**

**$ git config –global user.email**



1. 타이젠 소스코드 다운로드

Tizen Gerrit 에서 Tizen Open Source의 전체 git 목록을 확인할 수 있다.



이 소스코드들을 다운로드 받고 싶을 때는 git clone 을 해야 한다.



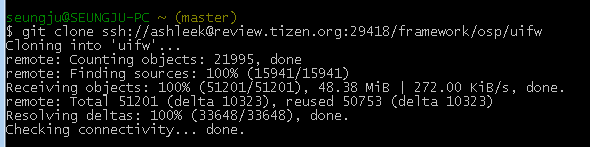
git clone 은 SSH, HTTP 방법으로 clone 받을 수 있다.

(경험 상, HTTP는 보안 상 걸리는 부분이 생길 수 있으므로 SSH를 사용하는 것이 편하다. )

Git Bash 을 실행 시킨 후

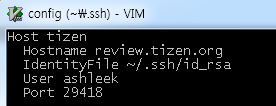
**$ git clone ssh://{계정}@review.tizen.org:29418/{git주소}**

형식으로 clone 하면 된다.



위와 같이 소스코드가 다운로드 되는 것을 확인 할 수 있다.

.ssh 디렉토리의 config 파일에 아래와 같이 사용자이름과 port를 미리 등록해두면 git clone 을 할 때, 사용자 계정 및 포트 번호를 입력하지 않아도 된다.



**$ git clone review.tizen.org:{git주소}**

1. **TIZEN SPECIFIC SOURCE DOWNLOAD(LINUX)**

위의 3. TIZEN specific source download(Windows) 에서 했던 Tizen Git에 사용자 등록하는 것까지는 똑 같은 단계를 거치면 된다. 하지만 Windows 환경에서와는 달리 Git Bash가 아니라 Linux terminal을 사용해서 작업하면 된다.

**$ ssh review.tizen.org gerrit ls –project**

를 통해 tizen gerrit project의 목록을 확인할 수 있다.

**$ git clone ssh://<Username>@review.tizen.org:29418/<Gerrit\_Project>**

명령어를 통해 clone 받을 수 있다.

1. **TIZEN ALL SOURCE(PROJECT) DOWNLOAD (Windows)**

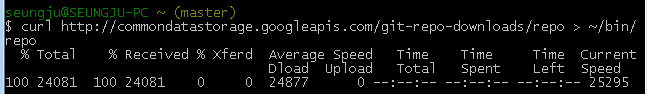
SSH를 통해 타이젠 오픈소스의 모든 프로젝트들과 snapshot source 를 다운로드 받을 수 있다.

1. ~/bin/ 디렉토리 만들기



1. repo script 다운로드 받기

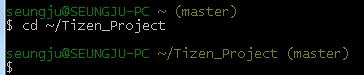
$ curl http:*//commondatastorage.googleapis.com/git-repo-downloads/repo > ~/bin/repo*



1. 타이젠 소스 담을 디렉토리 따로 만들기



경로를 만든 디렉토리로 이동시킨다.



1. 타이젠 전체 프로젝트 cloning 하기

/////////////////////////////////////////// 성공 시 완성//////////////////////////////////////

1. **TIZEN ALL SOURCE(PROJECT) DOWNLOAD (LINUX)**
2. bin 폴더 만들기

**$ mkdir ~/bin/**

**$ PATH=~/bin:$PATH**

1. repo script 다운로드 받기

**$ curl https://commondatastorage.googleapis.com/git-repo-downloads/repo > ~/bin/repo**

1. repo 속성 바꿔서 실행 가능하도록 만들기

**$ sudo chmod a+x ~/bin/repo**

1. 타이젠 프로젝트를 위한 새로운 디렉토리 만들기 그리고 위치 바꾸기

**$ mkdir ~/<Tizen\_Project>**

**$ cd ~/<Tizen\_Project>**

1. 최신 타이젠 전체 프로젝트 다운로드 받기

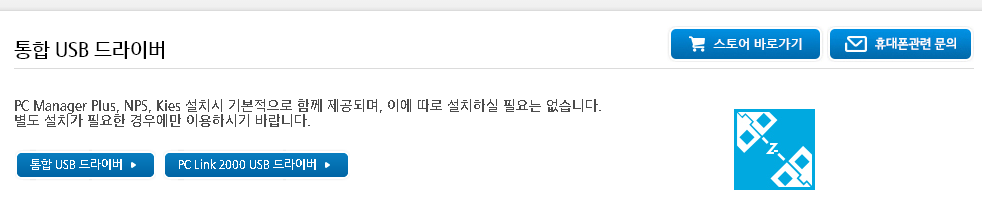


1. 저장소와 동기화 하기

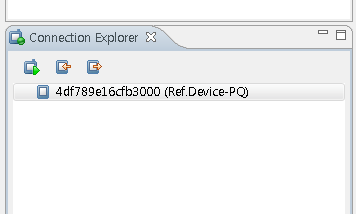
**$ repo sync**

1. **Building Packages Locally with GBS (Linux)**
2. **TIZEN Reference phone에 어플리케이션 올리기**
3. 삼성 USB 통합 드라이버 설치하기

URL : <http://www.samsung.com/sec/support/pcApplication/USB/>



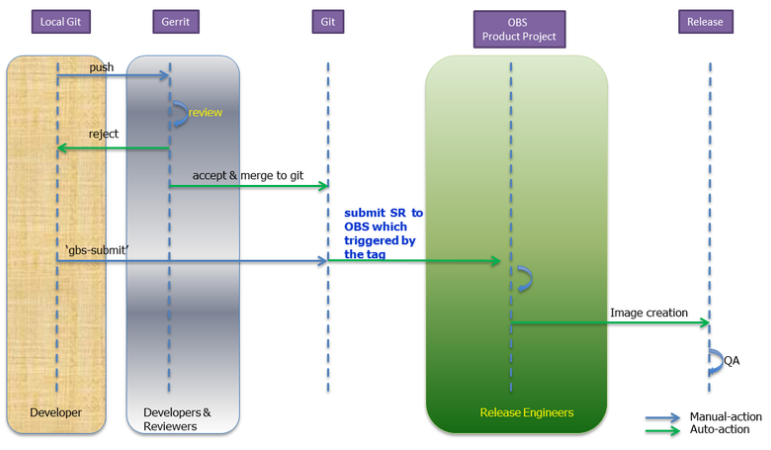
1. Reference phone 환경 설정
   1. 개발자 옵션 -> USB 디버깅 ON
   2. 날짜 및 시간 -> 자동 업데이트 OFF -> 수동으로 현재 날짜 및 시간으로 설정한다. (중요)
2. TIZEN IDE에서 connection 확인하기



위와 같이 connection Explorer 에서 레퍼런스 폰의 정보가 뜨는지 확인한다.

이렇게 폰이 연결되고 나서 프로젝트 RUN AS.. -> TIZEN NATIVE APPLICATION을 하면 자동으로 폰에서 어플리케이션이 실행되는 것을 확인 할 수 있다.

1. **TIZEN IMAGE PORTING**
2. **Creating Tizen Images with MIC**
3. **TIZEN PROJECT CONTRIBUTE**



1. Gerrit 에 패치 Submit 하기
2. 프로젝트 디렉토리 위치를 바꾸고 로컬에서 개발을 진행한다.
3. 변경된 프로젝트를 아래의 명령으로 올린다.

**$ git add <Revised\_File>…**

1. 올린 프로젝트를 아래의 명령으로 commit 시킨다.

**$ git commit**

1. Gerrit에 패치를 Push 시킨다.

**$ git push origin HEAD:refs/for/<remote\_branch\_name>**

Tizen 2.1 -> tizen\_2.1 (브렌치 명시법)

Tizen 2.2 -> tizen\_2.2

Tizen 3.0 -> tizen

1. Gerrit 에 패치 Review 하기

이 단계 에서는 패치에 대한 vote(점수 매기기)가 진행된다.

이 점수에 따라서 이 패치가 최종적으로 merge 될지 아니면 discarded 될지 결정된다. Code Review category에서는 +2, -2 점으로 평가되고 Verified category에서는 +1, -1 점으로 평가되는 방식이다.

1. OBS에 패치 Submit 하기
2. OBS 에 패키지 submit 하기

**$ gbs submit –c <commit\_ID> -m “<Comments>”**

제출 도중에 GBS는 자동으로 아래의 형태의 annotated tag 를 만든다.

**submit /$Tizen\_version/$(%Y%m%d.%H%M%S)**

Gerrit에 바뀐 코드들이 다 통합되고 나면, 통합 요청이 release engineer들에게 가게 되고 release engineer들이 review를 하게 될 것이다.

1. OBS에 패키지 그룹 submit 하기

상호의존도가 높은 다중 변경 패키지는 한번에 그룹으로 submit되어야 한다. 그것을 group submission이라고 부른다.

이런 group submission은 태그를 만들고 submit을 하면 된다.

**$ cd platform/core/appfw/aul-1/**

**$ gbs submit –m “<Comments>”**

위의 명령으로 tag name을 부여받고 나서, 같은 그룹의 다른 패키지에도 –tag 를 인자처럼 쓰면 된다.

**$ cd platform/core/appfw/ail/**

**$ gbs submit –tag <same\_tag> -m “<Comments>”**

1. OBS에 패키지 Review, Aceept 하기 (release engineer 만)

QA engineer들은 독립적인 object들인 테스트하고 또한 Tizen image들을 검증하고 내부의 세부적인 패키지 안의 종합적인 정보들을 release engineer들에게 제공하고 이 패키지들을 승인할지 말지에 대한 경정을 내릴 수 있도록 한다.

아래의 단계를 거친다.

1. 이 패키지가 패키지 빌드에 영향을 미치는지
2. 이 패키지가 새로운 버그를 일으키는지
3. 이 패키지가 퇴보 상황을 일으킬지
4. 이 패키지가 Tizen image의 성능에 영향력이 있는지

이러한 모든 과정들을 거쳐서 release engineer들에게 승인 받으면, Tizen image에 통합되고 normal release 과정을 거쳐서 <https://download.tizen.org/release/daily> 에 포함되게 된다.