



# Configuration Management



## Transit Droid Team

Austin Takam  
Daniel Magni  
Paul Smelser  
Razvan-Lada Moldovan  
Yasser Al-Hasan

---

# REVISION HISTORY

Date	Rev.	Description	Author(s)
	1.0	Initial version	Paul Smelser

---

# CONTENTS

1	Introduction .....	3
1.1	Purpose.....	3
1.2	Scope.....	3
1.3	Definitions, Acronyms and Abbreviations.....	3
2	Organization of the CM Framework.....	4
2.1	Roles and Responsibilities .....	4
2.1	Applicable rules .....	4
2.2	CM Tools .....	4
3	Configuration Management Process .....	5
3.1	Identification .....	5
4	Configuration Management and Control .....	5
4.1	Change requests .....	5
4.2	Change Evaluation .....	5
4.3	Change Implementation .....	5
5	Configuration Audits and Review .....	5
6	Project Schedules and CM Milestones .....	6
7	Glossary .....	7

---

# CONFIGURATION MANAGEMENT

## 1 INTRODUCTION

### 1.1 PURPOSE

The purpose of this Configuration Management (CM) plan is to document and inform project stakeholders about CM with the Transit Droid system. This plan allows us to integrate changes to the project easily while reducing possible side effects caused by the modifications.

### 1.2 SCOPE

This plan will outline the roles and responsibilities of the team along with the tools that will be used to track and implement changes. This applies to all documentation and code related to the three main components of Transit Droid: the web services, the mobile application and the terminal application.

### 1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Please refer to the Project Glossary for definitions, acronyms and abbreviations needed to properly interpret this document.

## 2 ORGANIZATION OF THE CM FRAMEWORK

All the members of the Transit Droid project share the same responsibilities.

### 2.1 ROLES AND RESPONSIBILITIES

- Software Developers
  - All five members of the Transit Droid project must contribute equally to coding, testing and documenting their work. In addition to this, all members must adhere to the CM rules defined here.

### 2.1 APPLICABLE RULES

- All code must be added and stored in the SVN repository.
- Code that is checked-in must be able to compile and build.
- In the case of revision conflicts, they must be resolved between the developers as soon as possible.
- Documentation, such as PPT presentations, PDF documents, WORD documents and text files must be maintained on the Google Drive repository.

### 2.2 CM TOOLS

- Subversion
  - This is a free open-source version control system that manages, tracks and stores files over time.
- Atlassian Fisheye
  - This is a paid web-based source code management software which has support for Subversion Repositories.
- Atlassian JIRA
  - This is a paid web-based issue & project tracking software.
- Eclipse IDE
  - This free tool will be used for writing and building our software. In addition to this, it also has a plugin for Subversion that we can use to track and manage our code.

## 3 CONFIGURATION MANAGEMENT PROCESS

### 3.1 IDENTIFICATION

There are three types of Configuration Items (CI): Code, Metadata and Documentation.

1. Code is kept in the SVN repository where it can be accessed and updated by all members. Code must have proper comments and meaningful variable names. This will make it easier for refactoring. Finally, JAVA coding standards must be followed.
2. Metadata such as the SQL scripts used to create database tables will also be kept in the SVN repository.
3. Documentation is kept in the Google Drive repository.

## 4 CONFIGURATION MANAGEMENT AND CONTROL

### 4.1 CHANGE REQUESTS

For major change requests, each developer should work only on CIs that are assigned to them. If however there is a need to change another CI then the developer must notify its owner and get permission first. For minor changes such as spelling errors and formatting, permission is not required but the CI owner must be notified.

### 4.2 CHANGE EVALUATION

Major changes will be evaluated by the whole team during each weekly meeting.

### 4.3 CHANGE IMPLEMENTATION

If the change request is evaluated and approved then changes will be committed in the SVN repository and a new revision will be created as a result.

## 5 CONFIGURATION AUDITS AND REVIEW

The Transit Droid can be followed at any time by looking at the remote SVN repository and the JIRA board.

## 6 PROJECT SCHEDULES AND CM MILESTONES

Here are the milestones for the Transit Droid project:

- Prototype (November 1<sup>st</sup> 2012)
- First release (December 20<sup>th</sup> 2012)
- Second release (January 31<sup>st</sup> 2012)
- Third release (March 31<sup>st</sup> 2012)
- Final release (April 4<sup>th</sup> 2012)

## 7 GLOSSARY

Java	A high-level imperative computer programming language.
IDE	Integrated development environment.
PPT	A slide presentation format used by the Microsoft PowerPoint program.
WORD	A text file format used by the Microsoft Word program.
PDF	A file format that shows an image of media such as text or images.
SVN	An open source revision control system (Subversion).
Google Drive	Online file storage service by Google.
JIRA	A paid web-based issue & project tracking software.