



simpli-city

The Road User Information System Of The Future

WP6 – Personal Mobility Assistant

D6.4: Application Design Studio Prototype

Deliverable Lead: ASC

Contributing Partners: ASC, TALK, TIE

Delivery Date: 04/2015

Dissemination Level: Public

Version 1.0

This deliverable describes the work carried out during the development of the final prototype of the Application Design Studio component of the SIMPLI-CITY platform. It specifies the scope of this version and the degree of fulfilment of the requirements to be covered by the component. It specifies how to install and execute the different subcomponents implemented.



Document Status	
Deliverable Lead	Jan Reehuis, ASC
Internal Reviewer 1	Marina Giordanino, CRF
Internal Reviewer 2	Christoph Hochreiner, TUV
Type	Deliverable
Work Package	WP6: Personal Mobility Assistant
ID	D6.4: Application Design Studio Prototype
Due Date	31.03.2015
Delivery Date	30.04.2015
Status	For Approval

Document History	
Contributions	<p>V0.1, Stefan Schulte, Philipp Hoenisch, TUV, 04.12.2012, Added document structure.</p> <p>V0.2, Jan Reehuis, ASC, 27.02.2015, Initial Document Structure.</p> <p>V0.3, Jan Reehuis, ASC, 17.03.2015, Further Elaboration of the Section Content.</p> <p>V0.4, Arturo Brotons, TIE, 08.04.2015, Updated requirements, added screenshot of plugins folder after installation, small corrections.</p> <p>V0.5, Jan Reehuis, ASC, 09.04.2015, Detailing of several Sections</p> <p>V0.6, Arturo Brotons, TIE, 10.04.2015, Elaborating Section 5.1</p> <p>V0.7, Jan Reehuis, ASC, 14.04.2015, Finalizing the Document for Review</p> <p>V0.8, Arturo Brotons, TIE, 14.04.2015, Small corrections before Review</p> <p>V0.9, Jan Reehuis, ASC, 21.04.2015, Integration of the internal Review comments</p> <p>V0.10, Jan Reehuis, ASC, 23.04.2015, Integration of the internal Review comments</p>
Final Version	V1.0, Jan Reehuis, ASC, April 27 st 2015

Note

This deliverable is subject to final acceptance by the European Commission.

Disclaimer

The views represented in this document only reflect the views of the authors and not the views of the European Union. The European Union is not liable for any use that may be made of the information contained in this document.

Furthermore, the information is provided “as is” and no guarantee or warranty is given that the information is fit for any particular purpose. The user of the information uses it at its sole risk and liability.

D6.4_Application_Design_Studio_Prototype_v1.0_F or_Approval.docx	Document Version: 1.0	Date: 2015-04-30	Status: For Approval	Page: 3 / 24
http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201			

Project Partners



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Vienna University of Technology (Coordinator),
Austria



Ascora GmbH, Germany



TIE Nederland B.V., The Netherlands



Technische Universität Darmstadt, Germany



IBM Research – Ireland
Smarter Cities Technology Centre



Forschungsgesellschaft Mobilität, Austria



Talkamatic AB, Sweden



Atos Worldline, Spain



Centro Ricerche FIAT, Italy



SRM – Reti e Mobilità, Italy

Executive Summary

This deliverable describes the work carried out during the development of the final prototype of the SIMPLI-CITY Application Design Studio component.

The document starts by introducing the Application Design Studio and describing the scope of this final prototype.

Afterwards, the degree of fulfilment of each requirement to be covered by the component and specified in the Requirements Analysis Report (D2.3) is described.

In the current version, the functionality of the Application Design Studio is given. It provides installation and execution of the component, which can be used by the developer to easily develop Apps for the SIMPLI-CITY platform.

The latest sections of this document describe how potential users (i.e. developers for SIMPLI-CITY) can prepare, install and execute the Application Design Studio component. For this, a step-by-step process to install and make use of the prototype is provided.

This deliverable D6.4 will conclude the work on T6.4 - Application Design Studio.

D6.4_Application_Design_Studio_Prototype_v1.0_F or_Approval.docx	Document Version: 1.0	Date: 2015-04-30	Status: For Approval	Page: 5 / 24
http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201			

Table of Contents

1	Introduction	7
1.1	SIMPLI-CITY Project Overview	7
1.2	Deliverable Purpose, Scope and Context.....	8
1.3	Document Status and Target Audience.....	8
1.4	Abbreviations and Glossary.....	8
1.5	Document Structure.....	8
2	Prototype Scope and Requirements Coverage.....	9
2.1	Application Design Studio – General Information	9
2.2	Scope of the Final Prototype	10
2.2.1	Development Base Plugin.....	10
2.2.2	App Project Wizard	10
2.2.3	Application Development Plugin	11
2.2.4	Manifest Editor	11
2.2.5	Documentation Plugin.....	11
2.3	Covered Requirements.....	11
3	Preparations	16
3.1	SIMPLI-CITY Application Design Studio (App Developers).....	16
4	Installation (Deployment)	17
4.1	SIMPLI-CITY Application Design Studio IDE Plugin (App Developers)	17
5	Execution and Usage of the Software.....	18
5.1	SIMPLI-CITY Application Design Studio IDE Plugin (App Developers)	18
6	Summary	24

1 Introduction

SIMPLI-CITY – The Road User Information System of the Future – is a project funded by the Seventh Framework Programme of the European Commission under Grant Agreement No. 318201. It provides the technological foundation for bringing the “App Revolution” to road users by facilitating data integration, service development, and end user interaction.

Within this document, the final prototype of the Application Design Studio will be presented. The document accompanies the corresponding software prototype, which is the main content of the deliverable.

1.1 SIMPLI-CITY Project Overview

Analogously to the “App Revolution”, SIMPLI-CITY adds a “software layer” to the hardware-driven “product” mobility. SIMPLI-CITY will take advantage of the great success of mobile apps that are currently being provided for systems such as Android, iOS, or Windows Phone. These apps have created new opportunities and even business models by making it possible for developers to produce new apps on top of the mobile device infrastructure. Many of the most advanced and innovative apps have been developed by players formerly not involved in the mobile software market. Hence, SIMPLI-CITY will support third party developers to efficiently realise and sell their mobility-related service and app ideas by a range of methods and tools, including the Mobility Services and Application Marketplaces.

In order to foster the wide usage of those services, a holistic framework is needed which structures and bundles potential services that could deliver data from various sources to road user information systems. SIMPLI-CITY will provide such a framework by facilitating the following main project results:

- **Mobility Services Framework:** A next-generation European Wide Service Platform (EWSP) allowing the creation of mobility-related services as well as the creation of corresponding apps. This will enable third party providers to produce a wide range of interoperable, value-added services, and apps for drivers and other road users.
- **Mobility-related Data as a Service:** The integration of various, heterogeneous data sources like sensors, cooperative systems, telematics, open data repositories, people-centric sensing, and media data streams, which can be modelled, accessed, and integrated in a unified way.
- **Personal Mobility Assistant:** An end user assistant that allows road users to make use of the information provided by apps and to interact with them in a non-distracting way – based on a speech recognition approach. New apps can be integrated into the Personal Mobility Assistant in order to extend its functionalities for individual needs.

To achieve its goals, SIMPLI-CITY conducts original research and applies technologies from the fields of Ubiquitous Computing, Big Data, Media Streaming, the Semantic Web, the Internet of Things, the Internet of Services, and Human-Computer Interaction. For more information, please refer to the project website at <http://www.simpli-city.eu>.

1.2 Deliverable Purpose, Scope and Context

The purpose of this document is to provide the means to use the final prototype of the Application Design Studio and exploit its functionalities. For this, the scope of the Application Design Studio, prototype requirements and preparations for developers as well as an installation and usage guide are provided.

The final Application Design Studio prototype is the outcome of the discussions and implementation work done in project months 19 to 30. It provides the implementation of the functionalities of the Application Design Studio as provided with SIMPLI-CITY deliverables D3.2.1 (Functional Specification), and D3.2.2 (Technical Specification). This deliverable D6.4 is the final prototype of the Application Design Studio and will conclude T6.4.

1.3 Document Status and Target Audience

This document is listed in the Description of Work (DoW) as “Public”, since the content is functional complete (beta prototype) but subject to changes, depending on the upcoming demands in other work packages and/or issues with the current implementation.

1.4 Abbreviations and Glossary

A definition of common terms and roles related to the realization of SIMPLI-CITY as well as a list of abbreviations is available in the supplementary document “Supplement: Abbreviations and Glossary”, which is provided in addition to this deliverable.

Further information can be found at <http://www.simpli-city.eu>.

1.5 Document Structure

This deliverable is broken down into the following sections:

Section 1 provides an introduction for this deliverable including a general overview of the project and outlines the purpose, scope, context, status and target audience of this deliverable.

Section 2 provides an overview of the scope and relationship of the prototype, showing how the Application Design Studio fits into the overall SIMPLI-CITY software framework and describing the outcome of this final prototype. Furthermore, an assessment of the requirements covered by this prototype is given.

Section 3 presents the requirements and preparations to be done by software developers if they want to use the Application Design Studio prototype.

Section 4 states information about the installation and deployment of the provided software package.

Section 5 describes how software developers can use the provided functionalities.

Finally, Section 6 provides a summary of the document.

D6.4_Application_Design_Studio_Prototype_v1.0_F or_Approval.docx	Document Version: 1.0	Date: 2015-04-30	Status: For Approval	Page: 8 / 24
http://www.simpli-city.eu/		Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201		

2 Prototype Scope and Requirements Coverage

2.1 Application Design Studio – General Information

The Application Design Studio is a toolset for developers to create and publish SIMPLI-CITY Apps to the Application Marketplace. It gives support over the whole chain of developing an app for SIMPLI-CITY.

In detail, the App Project Wizard guides the developer through the creation process for a new SIMPLI-CITY App. With the gathered information, the wizard will generate the necessary project structure and sets up some templates. Based on these basic preparations the developer can directly focus on the code for the App.

During the implementation, the rich documentation is in place to show how to use the SIMPLI-CITY Application Runtime Features in the app by giving examples, tutorials and a technical documentation of all functions. For editing important metadata files like the App Manifest, the Application Design Studio provides a visual editor, so the developer does not have to take care of the file structure.

A bundling mechanism will package the app for the developer, so it can be tested in the Personal Mobility Assistant and later published in the Application Marketplace. This will be supported by a wizard, which will ask for the necessary information.

Figure 1 shows the relation of the Application Design Studio regarding the SIMPLI-CITY Global Architecture. For the full Global Architecture, refer to deliverable D3.1.

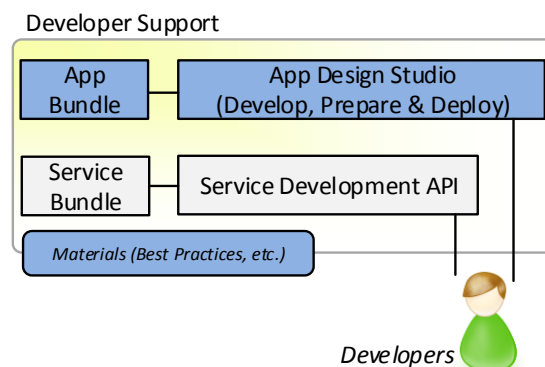


Figure 1: Relation of the Application Design Studio Regarding the SIMPLI-CITY Global Architecture

2.2 Scope of the Final Prototype

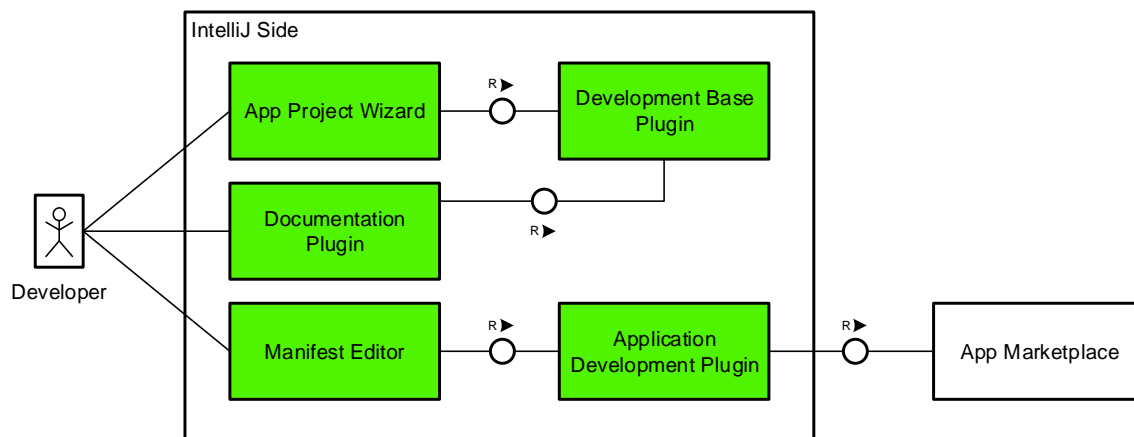


Figure 2: Scope of the Final Prototype of the Application Design Studio

Figure 2 depicts the status of development of the final prototype of the Application Design Studio, showing the subcomponents that are covered within this prototype. It has to be noted, that the Application Marketplace is not in the scope of D6.4 but is included in Figure 2 for completeness.

The status of the implementation is shown using the following colour codes:

- Green: Fully implemented.
- Orange: Partially implemented.
- White: No implementation so far or not part of this deliverable.

In the following subsections, the scope and status of the single subcomponents (as depicted in Figure 2) will be discussed in more detail. For the Functional Specification and Technical Specification of these subcomponents, refer to SIMPLI-CITY deliverables D3.2.1 and D3.2.2, respectively.

2.2.1 Development Base Plugin

The Development Base Plugin provides a unified access to build-in functionalities of the IntelliJ IDE for the specific development plugins. For instance, the hook into the Project Wizard and the creation of a menu item in IntelliJ for SIMPLI-CITY provides the access to the Application Design Studio functionality for the app developer. This Base Plugin is also used by the Service Development API to create a unified access to functionalities like the documentation and the management of developer credentials.

2.2.2 App Project Wizard

To ease the creation of a new app development project in IntelliJ, the Application Design Studio is integrated into the built-in project wizard. This wizard will guide the app developer through the process of setting up a new app project by requesting the needed information and creating the basic project structure. In this way, the app developer can focus on the app implementation, without caring about the internal structure.

2.2.3 Application Development Plugin

The Application Market Place provides a RESTful interface to be used by the Application Design Studio for bundle persistence and to publish the app in the Market. Within this prototype, the basis of this subcomponent has been developed, including a mockup version of the RESTful interface for the communication with the server side subcomponents. Specifically, the RESTful interface provides methods for app publishing, which can be used by the plugin to test the RESTful interface and the format of the messages.

2.2.4 Manifest Editor

Every SIMPLI-CITY app needs some meta-information which are processed by the Application Marketplace to gain all needed information during the app publishing procedure. The Manifest Editor provides a handy GUI on top of the internal manifest file structure that enables an easy creation and modification.

2.2.5 Documentation Plugin

To support the developer while writing Apps and Services for SIMPLI-CITY, the Documentation Plugin is a one-stop-shop for tutorials, explanations and examples. For the Application Design Studio, the documentation plugin provides the whole workflow from creating the app project, writing the first app methods, accessing functionality of the Application Runtime Environment, bundling and publishing an app.

2.3 Covered Requirements

This section describes the degree of fulfilment of the requirements to be covered by the Application Design Studio and specified in the Requirements Analysis Deliverable (D2.3) and the Functional Specification (D3.2.1).

Table 1: Requirements Related to Application Design Studio and their Degree of Fulfilment

Requirement	Degree of Fulfilment	Comment
Must Have Requirements		
U105. Access to the dialog system U106. Access to cloud services U159. Support for dialogue U160. Support for apps U161. Support for data services U162. Support for backend services	100%	For the requirements U106, U160, U161, and U162 the Application Design Studio makes use of the functionality developed in the Application Runtime Environment component. It provides easy methods for the <i>App</i> class to access the different components of the SIMPLI-CITY platform. The development and use of the dialogue system, mentioned in U105 and U159, is used to provide an easy to use editor in the Application Design Studio.
U155: Provision of Java API U157: Standard programming interface U158: Easy access to API through the developer studio	100%	All provided functionalities of the Application Runtime Environment component are accessible via a shared Java library for developers in the Application Design Studio. During the project creation process, the API is already loaded and included into the new app project.
U166: Identification of the developer/signature	100%	Every SIMPLI-CITY app has a manifest file. In this file, all metadata about the app are stored and during the publishing procedure the Application Marketplace processes this data. This enables the user to browse this information already before installing the app in the PMA.

Requirement	Degree of Fulfilment	Comment
U168: Provision of source code examples U169: Provision of UI templates U170: Provision of best practices U171: Provision of tutorials U172: Provision of guidelines U173: Provision of examples	100%	With the Developer Guide, a rich documentation is provided to the developer. It includes code examples, references, tutorials, guidelines and best practices on how to develop a good app for the SIMPLI-CITY platform. When creating a new project for an app, the template already provides a functional example which demonstrates the use of some Application Runtime Environment functionality.
U178: Definition of minimum hardware requirements of apps	100%	The minimum hardware requirements of an app are ensured by PMA. As the PMA is an Android app around the SIMPLI-CITY apps, the permissions are defined during the install procedure. This ensures that all requirements are already fulfilled.
U182: Provision of statistics, e.g., usage, traffic	100%	During development, the Application Design Studio uses the built-in debugging mechanism of the Android plugin in IntelliJ. This enables a seamless integration into the IDE. During runtime, the Application Runtime Environment catches all occurring errors and forwards them to the Application Marketplace. This provides a convenient overview of all errors for the developer.

Requirement	Degree of Fulfilment	Comment
Should Have Requirements		
U23: Unified look & feel within the project U24: Unified look & feel for 3rd party developers U25: Usage of UI guidelines within the project U26: Intuitive usability	100%	As the Application Runtime Environment forces the developer to use the SIMPLI-CITY dialogue system, the UI of every SIMPLI-CITY app will be structured in the same manner. This also ensures that the whole app UI will be navigable in a multimodal way. Further, this also improves the usability, since the SIMPLI-CITY dialog system is based on the principles of simplicity, safety, usability and functionality.
U102: Backwards compatibility of API	100%	The interface of the Application Runtime Environment is designed to only introduce new methods but never changing the already existing ones. This ensures that older apps will be still executable without modification on new API versions.
U174: Permit to develop an app in less than a day U175: Hide complexity from the developer	100%	Due to the template for a new app project with a fully runnable SIMPLI-CITY app template, developers just have to add their own code. The Application Runtime Environment API provides methods to invoke or query services of the SIMPLI-CITY platform, without caring about the communication between the device and the Service Runtime Environment. It avoids boilerplate code for the developer and hides the complexity of the SIMPLI-CITY platform.

Could Have Requirements		
U180: Easy debugging of services U181: App crash reports U183: Provision of bug reports U184: Provision of crash reports	100%	The report generation relies on the PMA side and is carried out by the Error Handler, a subcomponent of the Application Runtime Environment. This requires that U181, U183 and U184 are fulfilled. Requirement U180 is provided by the implementation of U182.

3 Preparations

This section provides information about the required preparations for potential users (app developers) in order to use the functionalities of the delivered prototype.

3.1 SIMPLI-CITY Application Design Studio (App Developers)

In order to deploy the Application Design Studio, it is necessary to have IntelliJ IDEA Community Edition 13 or 14 installed. It can be downloaded from:

- <http://www.jetbrains.com/idea/download>

Detailed instructions on downloading and installing IntelliJ IDEA can be found here:

- <http://confluence.jetbrains.com/display/IntelliJIDEA/Basics+and+Installation>

The SIMPLI-CITY Application Design Studio IDE Plugin was developed and tested against versions 13 and 14 of the IDE on Windows 7 and Ubuntu 14.04 as operating system.

The plugin was developed to be used with the IntelliJ IDEA IDE only. It will not work with other IDEs like Eclipse¹ or NetBeans² (the choice of the target IDE is described in the corresponding section of D3.2.2 containing the Technical Specification).

The plugin is written in the Java language and so it is cross-platform. It is integrated in the IntelliJ IDEA and is subject to the same platform requirements as the host IDE. Please refer to the following page for a detailed list of system requirements:

- <http://www.jetbrains.com/idea/download>

The installation and use of the plugin was tested on Windows 7 and Ubuntu 14.04 to ensure the cross-platform use. Further, in this document it is presumed that IntelliJ IDEA Community edition 13 or 14 is successfully installed and running on the developer's computer.

¹ <https://eclipse.org/>

² <https://netbeans.org/>

4 Installation (Deployment)

This section provides guidelines on how to install and deploy the final prototype of the Application Design Studio.

4.1 SIMPLI-CITY Application Design Studio IDE Plugin (App Developers)

The SIMPLI-CITY Application Design Studio IDE Plugin is supplied as a zip archive: *app-design-studio-bundle.zip*. To deploy the plugin, the content of the zip file needs to be unzipped directly into the *plugins* folder of the IntelliJ IDEA installation folder. On Windows this folder can be usually found under

```
<SYSTEM DRIVE>\Users\<USER ACCOUNT NAME>\.IdeaIC<VERSION>
```

For further information visit <https://intellij-support.jetbrains.com/entries/23358108>.

After unzipping, the resulting folder structure should look like in Figure 3.

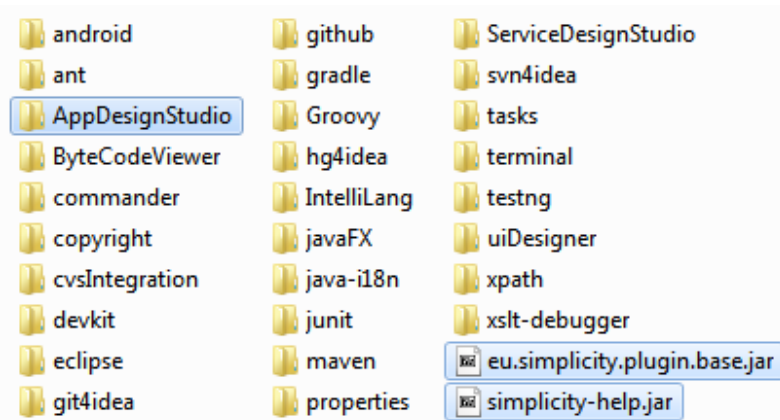


Figure 3: File Structure after Installing the Application Design Studio Bundle

In the *AppDesignStudio* folder, there should be exactly one folder called *lib*. It contains the plugin jar file and auxiliary libraries needed.

After the plugin is unzipped to the right location, the developer can start using it straight away (see Section 5.1 on executing and usage).

5 Execution and Usage of the Software

This section describes how to use the different subcomponents of the prototype.

5.1 SIMPLI-CITY Application Design Studio IDE Plugin (App Developers)

For the usage of the plugin, it is assumed that the user is already familiar with the family of IDEs based on IntelliJ IDEA, e.g., Android Studio. To access the plugin, IntelliJ IDEA has to be started up and running. Once all the elements are loaded, the plugin menu entry “SIMPLI-CITY” can be found in the main menu bar (see Figure 10). Here, the main functions of the Application Design Studio are accessible, e.g., the documentation on how to develop an app for SIMPLI-CITY can be opened (see Figure 10) or, after the development, the app can be bundled and tested here.

The “New SIMPLI-CITY App Project Wizard” can be found through the IntelliJ built in Project Creation Wizard (see Figure 4 and Figure 5). It will guide the developer through the “New SIMPLI-CITY App” project process and creates the necessary project structure.

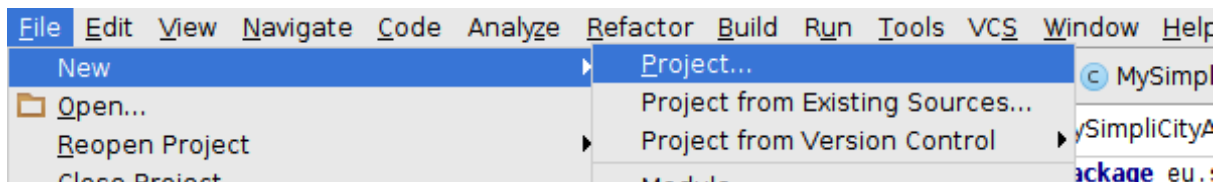


Figure 4: Execute the New Project Wizard

After opening the wizard, the user needs to select the “New SIMPLI-CITY App” (see Figure 5), under the “SIMPLI-CITY” types, in order to create a new app project.

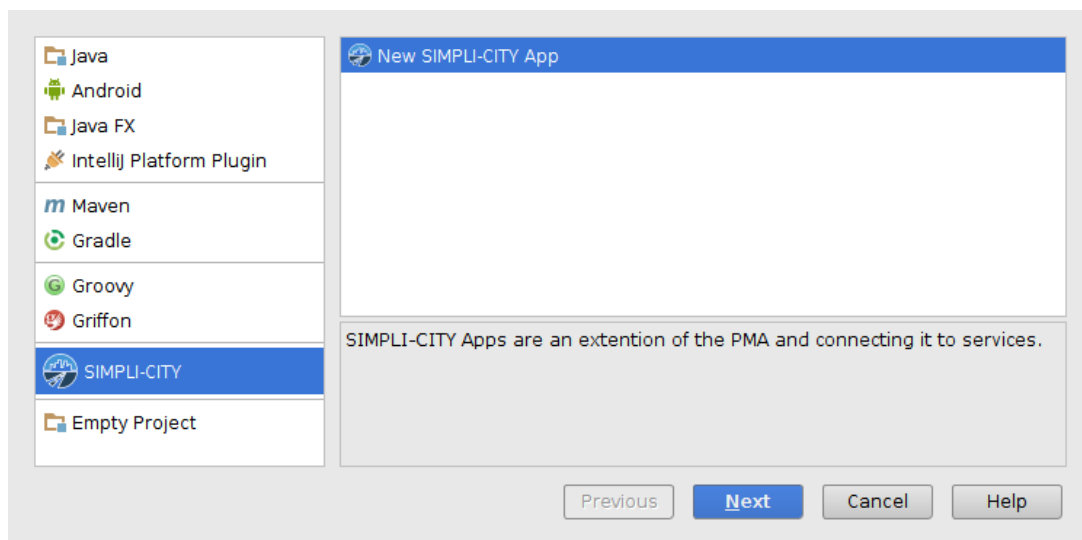


Figure 5: Choosing the New App Project Wizard

Then the user needs to choose a Group ID, an Artefact ID and the initial version of the project since SIMPLI-CITY Apps projects are based on Maven as shown in Figure 6. After this step, further information about the new project can be entered, i.e., the location in the local file system, package name, main class, author and company (see Figure 7).

GroupId: eu.simpli_city.apps Inherit

ArtifactId: mySimpliCityApp

Version: 1.0-SNAPSHOT Inherit

Buttons: Previous, Next, Cancel, Help

Figure 6: New App Wizard – App Bundling Information

Project name: MySimpliCityApp

Project location: ~/IdeaProjects/MySimpliCityApp

Package Name: eu.simpli_city.apps

Main Class: MySimpliCityApp

Your App ID: eu.simpli_city.apps.MySimpliCityApp

Author: John Doe

Company: SIMPLI-CITY Consortium

More Settings

Buttons: Previous, Finish, Cancel, Help

Figure 7: New App Wizard – App Configuration

Finally, when the user clicks *Finish*, a file structure similar to the one shown in Figure 8 is created based on the data entered in the wizard.

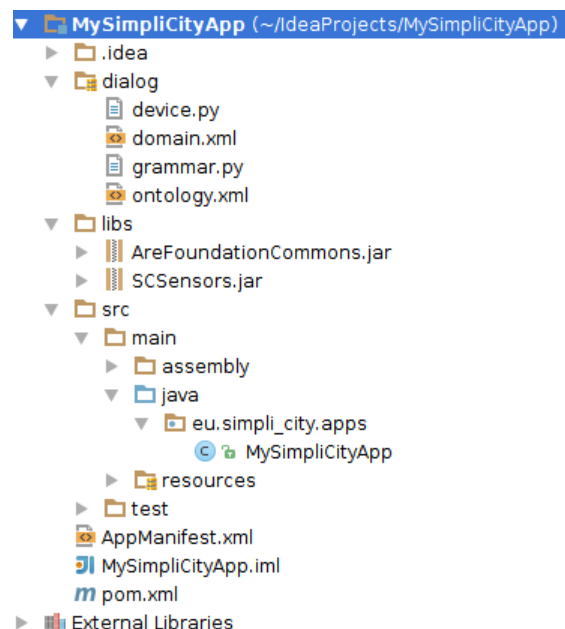


Figure 8: Project Structure after the Project Generation

The App Manifest Editor will be automatically invoked when opening the AppManifest.xml file. It supports the developer in setting up or editing the manifest, e.g., changing the name

of the app, the version, the description, without requiring any knowledge about the underlying XML structure (see Figure 9).

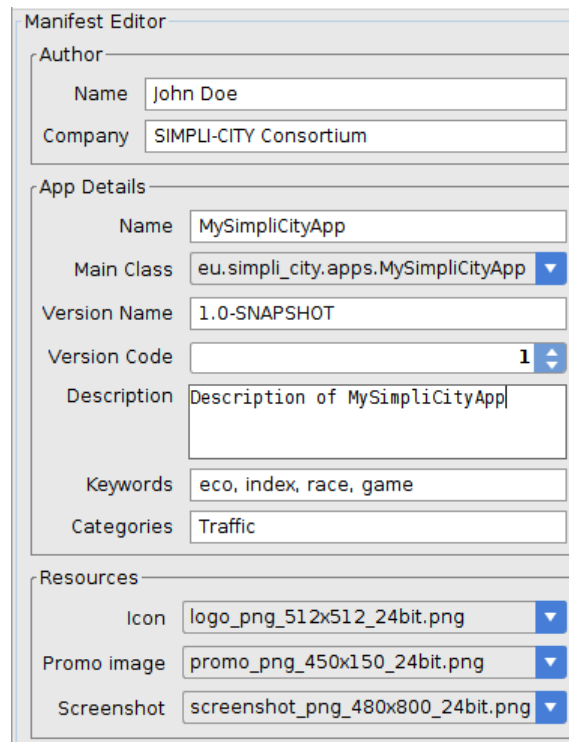


Figure 9: The App Manifest Editor

To support the developer during the whole development phase, the Application Design Studio includes a full-featured help, called “Developer’s Guide”. As it can be seen in Figure 10, the “Developer’s Guide” is also accessible by the SIMPLI-CITY Menu in the IDE.

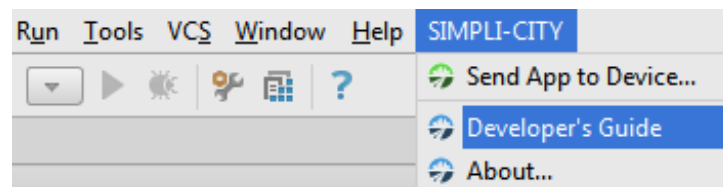


Figure 10: Accessing the SIMPLI-CITY Developer’s Guide

The Developer’s Guide documents both developer environments for SIMPLI-CITY: the Service Development API and the Application Design Studio. It is divided into different sections to give examples, guidelines and best practices to the developer (see Figure 11).



Figure 11: Application Design Studio Developer Guide

For testing the app under development, the Application Design Studio offers the direct push of the app bundle to a development device, where the PMA is installed. So the developer is able to interact with the new app like it was installed by the SIMPLI-CITY Application Marketplace. This option can be found under “SIMPLI-CITY > Send App to Device...” (see Figure 12).

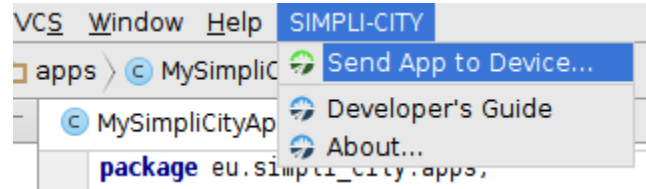


Figure 12: Access the Send to Device Functionality

The opened dialog, as shown in Figure 13, allows building the three needed bundles to test and publish the App:

- MMDI bundle: contains the dialog files and the App Manifest XML, needed to install on the MMDI backend.
- PMA bundle: contains a dex file³ for the Java classes, a set of resources that are used by the app and the App Manifest that is required to install on a PMA.
- Full bundle: contains all the aforementioned files in the previous bundles, which can be published on the Application Marketplace.

³ DEX files contain Java bytecode (the equivalent of binaries but for the Java Virtual Machine) to be executed on the Dalvik virtual machine used by Android operative systems.

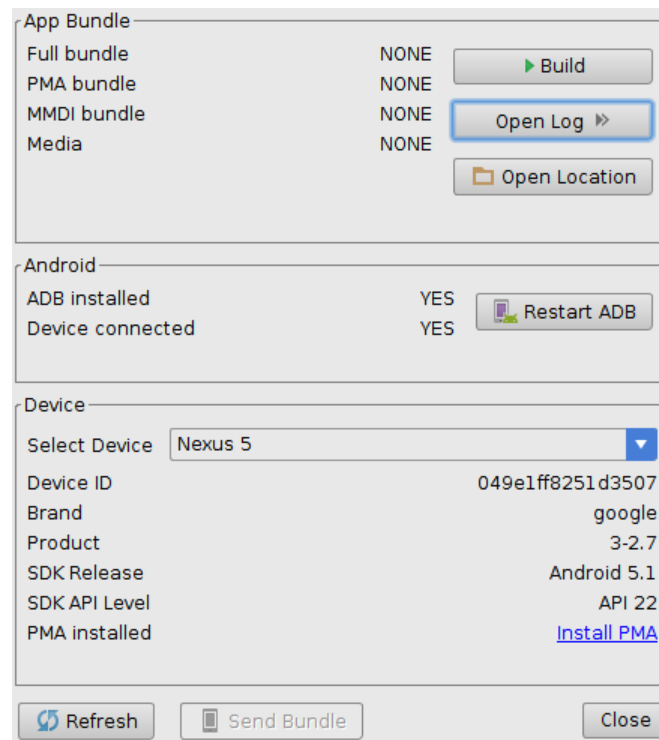


Figure 13: The Send to Device Dialog

A console output for the building process is provided so the user can get some insights on how the bundles are being build and also whether any error occurred during this process (see Figure 14). If the build is successful, the generated bundles will be found on the file system when clicking the “Open Location” button.

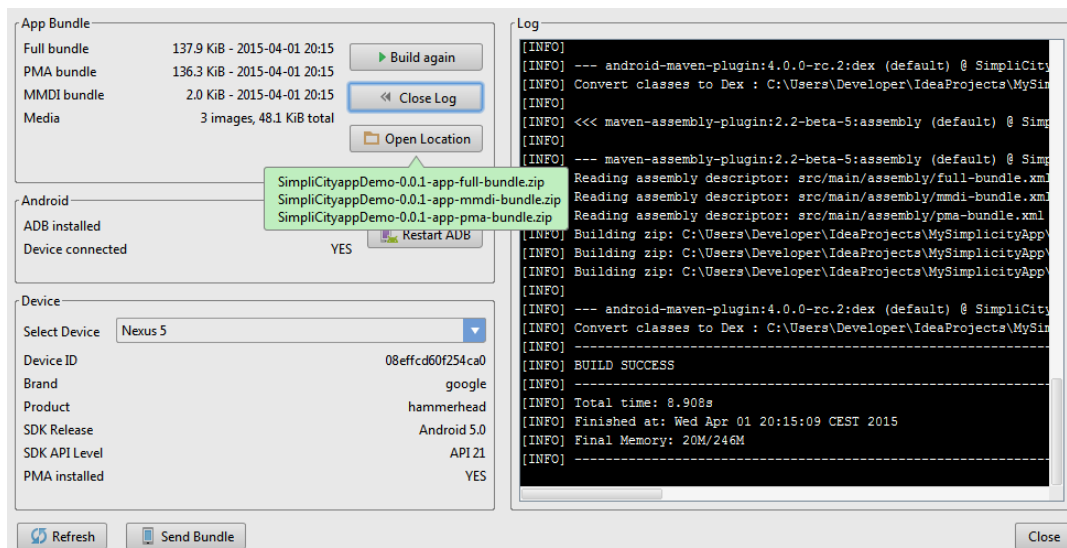


Figure 14: Successfully Generated Bundles

Also, if a compatible device is connected through USB cable and available by the Android Debug Bridge tool, the developer is able to push the generated bundle to the device. If any of the requirements to perform the operation is not met, an informative warning explaining the cause will be displayed, as shown in Figure 15.

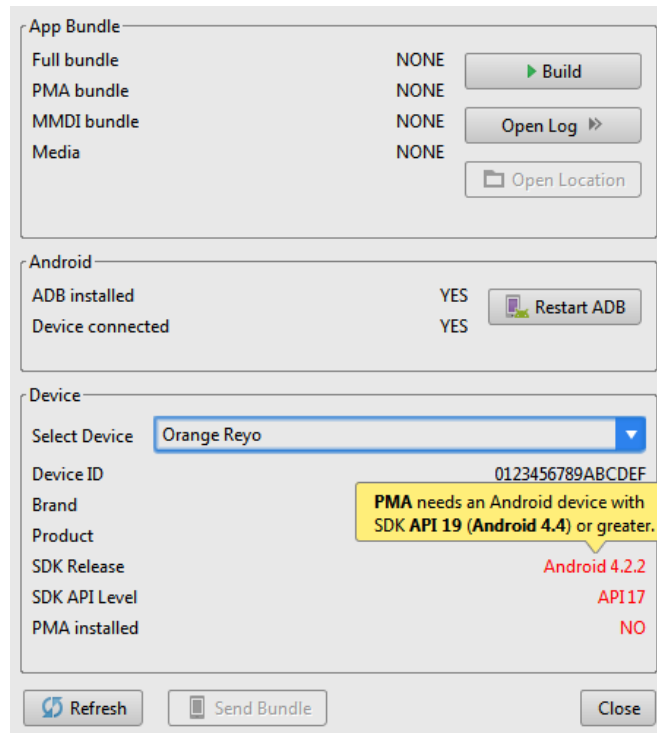


Figure 15: Incompatible Requirements of the Connected Device

6 Summary

This deliverable has presented and described the scope of the prototype of the Application Design Studio component for the SIMPLI-CITY Personal Mobility Assistant. Within this prototype, the development of the subcomponents has been finished. It is the basis for an easy and unified development for the SIMPLI-CITY platform.

With this prototype, developers are able to create a SIMPLI-CITY app project, bundle and publish the app to the Marketplace without knowing and caring about the internal app structure and the technical implementation of the Application Runtime Environment features. This approach eases the development for the SIMPLI-CITY platform as the developer can focus on the actual app code and does not care about SIMPLI-CITY specific mechanisms.

An analysis of the degree of fulfilment of the requirements to be covered by the component as specified in the Requirements Analysis Report (D2.3) has been presented.

Moreover, all the required steps to install, deploy, and execute the Application Design Studio Prototype have been discussed in detail.