

Annex to the Deliverable D200.7

Flspace SDK User and Developer Guide

WP 200

Project Acronym & Number:	Flspace – 604 123
Project Title:	Flspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics
Funding Scheme:	Collaborative Project - Large-scale Integrated Project (IP)
Latest version of Annex 1:	2013-10-03
Start date of the project:	01.04.2013
Duration:	24
Status:	Draft
Editor:	Said Rahma (ATOS)
Contributors (to the “R” part of the deliverable ¹ ; ordered by project partner)	ATOS: Said Rahma Rodriguez, Miryam Villegas
Document Identifier:	Flspace-D200.7-Annex_SDK_User_Developer_Guide-v0.4.docx
Date:	27.02.2015
Revision:	004
Project website address:	http://www.Flspace.eu

¹ Contributors to Flspace code (“P”) include ATB, UDE, IBM, ATOS, KOC, TOG, AST, NKUA, UPM and LimeTri; contributing persons are listed at <https://bitbucket.org/flspace/profile/members>

The Flspace Project

Leveraging on outcomes of two complementary Phase 1 use case projects (Flnest & SmartAgriFood), aim of Flspace is to pioneer towards fundamental changes on how collaborative business networks will work in future. Flspace will develop a multi-domain Business Collaboration Space (short: Flspace) that employs FI technologies for enabling seamless collaboration in open, cross-organizational business networks, establish eight working Experimentation Sites in Europe where Pilot Applications are tested in Early Trials for Agri-Food, Transport & Logistics and prepare for industrial uptake by engaging with players & associations from relevant industry sectors and IT industry.

Project Summary

As a use case project in Phase 2 of the FI PPP, Flspace aims at developing and validating novel Future-Internet-enabled solutions to address the pressing challenges arising in collaborative business networks, focussing on use cases from the Agri-Food, Transport and Logistics industries. Flspace will focus on exploiting, incorporating and validating the Generic Enablers provided by the FI PPP Core Platform with the aim of realising an extensible collaboration service for business networks together with a set of innovative test applications that allow for radical improvements in how networked businesses can work in the future. Those solutions will be demonstrated and tested through early trials on experimentation sites across Europe. The project results will be open to the FI PPP program and the general public, and the pro-active engagement of larger user communities and external solution providers will foster innovation and industrial uptake planned for Phase 3 of the FI PPP.

Project Consortium

- DLO; Netherlands
- ATB Bremen; Germany
- IBM; Israel
- KocSistem; Turkey
- Aston University; United Kingdom
- ENoLL; Belgium
- KTBL; Germany
- NKUA; Greece
- Wageningen University; Netherlands
- PlusFresc; Spain
- FloriCode; Netherlands
- Kverneland; Netherlands
- North Sea Container Line; Norway
- LimeTri; Netherlands
- BO-MO; Slovenia
- MOBICS; Greece
- Fraunhofer IML; Germany
- Q-ray; Netherlands
- FINCONS; Italy
- Kühne + Nagel; Switzerland
- University Duisburg Essen; Germany
- ATOS; Spain
- The Open Group; United Kingdom
- CentMa; Germany
- iMinds; Belgium
- Marintek; Norway
- University Politecnica Madrid; Spain
- Arcelik; Turkey
- EuroPoolSystem; Germany
- GS1 Germany; Germany
- Mieloo & Alexander; Netherlands
- OPEKEPE; Greece
- Innovators; Greece
- CIT; Spain
- SDZ; Germany
- Snoopmedia; Germany
- EECC; Germany
- CBT; Spain

More Information

Harald Sundmaeker (coordinator)
 Bert Vermeer (deputy coordinator)
 Project Website

e-mail: sundmaeker@atb-bremen.de
 e-mail: bert.vermeer@wur.nl
 Web link: <http://www.flspace.eu/>

Dissemination Level

PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	X
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Change History

Version	Notes	Date
001	Creation of the document SDK User and Developer Guide	09.12.2014
002	Update of the contents, update of the introduction section, adding a paragraph, updated text to clarify, improve and narrow the context Update of the number of Flspace plugin (15 to 17) Update of the figures in several sections Update of the Browser pane section Added the new sections related to new plugin (sketches, import and export plugin) Added the RCP app archetype section Update of the Flspace Studio Tool section Minor changes related to rephrasing and typo error Update of the abbreviation table, update of the references table	23.01.2015
003	Internal review process, checking URL links to the Flspace Web online documentation Update of the abbreviation table, update of the references table Final version ready for submission of approved document	06.02.2015
004	Update of the coordinator information in the section " <i>More Information</i> " Added Flspace development repository and documentation references, formatting improvement Final version ready for submission to EC	27.02.2015
005		
006		

Abbreviations

AAA	Authentication, Authorisation, and Accounting	IDE	Integrated Development Environment
ACSI	Artifact-Centric Service Interoperation	IDM	Identity Management
AdvB	Advisory Board	i.e.	id est = that is to say
AJAX	Asynchronous JavaScript + XML	IE	Integration Environment
API	Application Programming Interface	IEC	International Electrotechnical Commission
App	Software Application	IETF	Internet Engineering Task Force
B2B	Business-to-business	I/O	Input / Output
B2C	Business-to-Consumer	IoT	Internet of Things
BCM	Business Collaboration Module in Flspace	IP	Intellectual Property
BCO	Business Collaboration Objects in Flspace	IP (protocol)	Internet Protocol
BE	Business Entities	IPR	Intellectual Property Rights
BPPC	Business Process Participant Configuration	IPsec	Internet Protocol Security
BSS	Business Support Systems	IT	Information Technology
CDR	Charging Detailed Records	ITU	International Telecommunication Union
CEP	Complex Event Processing	ISO	International Standardization Organisation
CSB	Cloud Service Bus	J2SE	Java 2 Platform, Standard Edition
CSS	Cascading Style Sheets	JDK	Java Development Kit
CSV	Comma-Separated Values	JDT	Related to Eclipse Java Development Tools
D	Deliverable	JMX	Java Management Extensions
DAO	Data Access Object	JRE	Java Runtime Environment
DB	Database	JS	JavaScript
DoW	Description of Work	JSON	JavaScript Object Notation
EC	European Commission	JSP	Java Server Page
EDI	Electronic Data Interchange	JVM	Java Virtual Machine
EE	Experimentation Environment	KPI	Key Performance Indicator
e.g.	Exempli gratia = for example	LPA	Logistics Planning Application
EPA	Event Processing Agent	M	Month
EPM	Event Processing Module in Flspace	MTBF	Mean Time Between Failures
ESB	Enterprise Service Bus	MVC	Model–View–Controller
EU	European Union	OASIS	Organization for the Advancement of Structured Information Standards
FIA	Future Internet Assembly	OAuth	Open standard Authentication protocol
FI-PPP	Future Internet Public Private Partnership	OMG	Object Management Group
FP7	Framework Programme 7	OSS	Operational Support Systems
GA	Grant Agreement	P2P	Peer-to-peer
GE	Generic Enabler	PaaS	Platform as a Service
GUI	Graphical User Interface	PDE	Related to Eclipse Java Development Tools
HTML	HyperText Markup Language	PE	Production Environment
IaaS	Infrastructure as a Service	PIA	Product Information App
ICT	Information and Communication Technology		

PIE	Preliminary Integration Environment	SWT	Standard Widget Toolkit
PKI	Public Key Infrastructure	T	Task
PM	Person Month	TCP	Transmission Control Protocol
POM	Project Object Model (used by maven tools)	TIC	Tailored Information for Consumers
Proton	IBM Proactive Technology Online	TLS	Transport Layer Security
QoS	Quality of Service	TPM	Transport Planning Module
RBAC	Role-Based Access Control	UAA	User Management, Authentication and Authorisation
RCP	Rich Client Platform	UI	User Interface
REST	Representational State Transfer	UML	Unified Modeling Language
RFC	Request for Comments	URI	Universal Resource Identifier
RSS	Revenue Sharing System	URL	Universal Resource Locator
RTD	Research and Technological Development	USDL	Unified Service Description Language
SaaS	Software as a Service	VM	Virtual Machine
SDI	System and Data Integration layer in Flspace	VPN	Virtual Private Network
SDK	Software Development Kit	W3C	World Wide Web Consortium
SME	Small and Medium Sized Enterprise	WADL	Web Application Description Language
SOA	Service Oriented Architecture	WLAN	Wireless Local Area Network
SOAP	Simple Object Access Protocol	WP	Work Package
SOA-RM	(OASIS) Reference Model for Service Oriented Architecture	WS	Web Service
SPT	Security, Privacy and Trust Framework	WSDL	Web Services Description Language
SSH	Secure Shell	XLS/XLSX	Microsoft Excel file Format
SSL	Secure Sockets Layer	XML	eXtensible Markup Language
SSO	Single Sign On	XSD	XML Schema Definition
ST	Sub-Task		

Table of Contents

1	Introduction	11
1.1	Scope	11
1.2	Intended audience	16
2	SDK User/Developer Guide	17
2.1	Goal and objectives	17
2.2	Overview	17
2.3	Flspace-bcoeditor Plugin	19
2.4	Flspace-converter Plugin	20
2.4.1	Prerequisites	20
2.4.2	Using the plugin	20
2.4.3	Converting formats	20
2.5	Flspace-doc Plugin	24
2.6	Flspace-epmeditor Plugin.....	30
2.7	Flspace-export Plugin.....	31
2.8	Flspace-graphic Plugin	34
2.8.1	The Browser Pane	35
2.8.2	Flspace Graphics Toolbar	50
2.9	Flspace-import Plugin	52
2.10	Flspace-perspective Plugin	53
2.11	Flspace-preferences Plugin	54
2.12	Flspace-pubsub Plugin	58
2.12.1	Prerequisites	58
2.12.2	Using the plugins.....	58
2.12.3	Publish a message.....	58
2.12.4	Subscribe a message.....	59
2.13	Flspace-REST Plugin	60
2.14	Flspace-sdi-connector Plugin.....	63
2.14.1	Using the SDI Connector	64
2.14.2	The SDI Rest Tool	65
2.15	Flspace-Sketches Plugin	68
	Using the plugin.....	68
2.16	Flspace-SOAP Plugin	74
2.17	Flspace-uploader Plugin.....	77
2.18	Flspace-wizard Plugin	78
2.18.1	Widget Project	80
2.18.2	Compact Project	81
2.18.3	Decoupled Project	83
2.18.4	Node.js Project.....	84
2.18.5	JavaFX Project	85
2.18.6	RCP app archetype.....	87
2.19	Flspace-xmleditor Plugin.....	89
2.20	Flspace-logging Plugin.....	92
2.21	Flspace Studio Tool.....	96
3	Glossary	98
3.1	Terms and definitions	98
4	References	105

List of Figures

Figure 1: Flspace SDK features overview.....	13
Figure 2: Flspace SDK screenshot overview	18
Figure 3: Menu item to access the BCO Editor.....	19
Figure 4: Flspace BCO editor	19
Figure 5: Flspace converter	20
Figure 6: Format converter dialog (I).....	20
Figure 7: Format converter dialog (II).....	21
Figure 8: Format converter dialog (III).....	21
Figure 9: Format converter dialog (IV)	22
Figure 10: Format converter dialog (V)	22
Figure 11: Converting XLSX to JSON	23
Figure 12: Flspace doc generator	24
Figure 13: Flspace doc generation results.....	24
Figure 14: Flspace Doc dialog (I).....	25
Figure 15: Flspace Doc dialog (II).....	25
Figure 16: Flspace Doc Workspace tab (I)	26
Figure 17: Flspace Doc Workspace tab (II)	26
Figure 18: Flspace Doc Workspace tab (III)	26
Figure 19: Flspace Doc Direct tab (I).....	27
Figure 20: Flspace Doc Direct tab (II).....	27
Figure 21: Flspace Doc Direct tab (III).....	28
Figure 22: Flspace Doc Direct tab (IV)	28
Figure 23: External Generation result	29
Figure 24: Embedded Generation result	29
Figure 25: Flspace Doc validations (I)	29
Figure 26: Flspace Doc validations (II)	30
Figure 27: Flspace EPM editor	30
Figure 28: Flspace Export.....	31
Figure 29: Flspace Export Options (I)	31
Figure 30: Flspace Export Options (II)	32
Figure 31: Flspace Export Wizard	32
Figure 32: Flspace Project to be exported.....	33
Figure 33: Flspace Graphics perspective	34
Figure 34: Flspace Graphics Overview.....	34
Figure 35: Flspace Graphic Explorer	35
Figure 36: Web Services Element	35
Figure 37: REST client from file.....	36
Figure 38: REST client form (I)	36
Figure 39: REST client from URL	37
Figure 40: REST client form (II)	37
Figure 41: SOAP client from file.....	38
Figure 42: SOAP client form.....	38
Figure 43: Rest client from URL	39
Figure 44: SOAP client form.....	39
Figure 45: Publish/Subscribe Elements	39
Figure 46: Publish/Subscribe on Flspace Canvas.....	40
Figure 47: Publisher Form.....	41

Figure 48: Option Connect.....41

Figure 49: Connect Publisher/Subscriber41

Figure 50: Publisher/Subscriber Form42

Figure 51: Publisher and Subscriber not connected42

Figure 52: Fulfilling the Publisher/Subscriber settings manually43

Figure 53: Click on Subscribe button43

Figure 54: Endpoints elements43

Figure 55: Endpoints on Flspace Visual44

Figure 56: Widget Form – Tab Widget Settings.....44

Figure 57: Widget Form – Tab Graphic Settings.....45

Figure 58: Transformer formats45

Figure 59: Type of formats on Flspace Visual.....45

Figure 60: Format/Element XML Form46

Figure 61: XML Format – Insert by Direct Input46

Figure 62: XML Format – Insert by File Upload47

Figure 63: XML Format – Output.....47

Figure 64: XML Format – Graphic Settings48

Figure 65: XML Format (Source) and JSON Format (Target)48

Figure 66: Direct Input.....49

Figure 67: Connect the elements49

Figure 68: Target Output50

Figure 69: Flspace Graphic Toolbar50

Figure 70: Flspace Graphics Persistent Files.....51

Figure 71: Deleting Elements.....51

Figure 72: Flspace import plugin52

Figure 73: Flspace import wizard52

Figure 74: Flspace SDK Perspectives View entry53

Figure 75: Flspace Graphics perspective53

Figure 76: Flspace Preferences Settings Form54

Figure 77: Archetypes Settings Form.....54

Figure 78: BCO Settings Form55

Figure 79: Logging Settings Form55

Figure 80: Proxy Settings Form.....56

Figure 81: SDI Settings Form.....56

Figure 82: Uploader Settings Form.....57

Figure 83: Flspace publisher-subscriber58

Figure 84: Publisher dialog58

Figure 85: Subscriber dialog59

Figure 86: Flspace REST tool.....60

Figure 87: Flspace REST tool from file (I).....60

Figure 88: Flspace REST tool from file (II).....61

Figure 89: Flspace REST tool from URL (I)61

Figure 90: Flspace REST tool from URL (II)62

Figure 91: Button location to launch the SDI Connector plugin.....63

Figure 92: Menu item to launch the SDI Connector.....63

Figure 93: Flspace SDI Rest Tool (I).....63

Figure 94: Flspace SDI Rest Tool (II).....64

Figure 95: Flspace SDI Rest Tool view66

Figure 96: Flspace SDI Rest Tool file dialog	67
Figure 97: Menu item to launch MySketches plugin.....	68
Figure 98: Basic options in MySketches	68
Figure 99: Text option in MySketches (I).....	69
Figure 100: Text option in MySketches (II).....	69
Figure 101: Rubber option in MySketches	69
Figure 102: Filling options in MySketches (I).....	70
Figure 103: Filling options in MySketches (II).....	70
Figure 104: Lines in MySketches.....	71
Figure 105: Text Tool Font option in MySketches.....	71
Figure 106: Load option in MySketches	72
Figure 107: Save option in MySketches.....	72
Figure 108: Colour option in MySketches	73
Figure 109: Airbrush options in MySketches.....	73
Figure 110: Flspace SOAP tool.....	74
Figure 111: Flspace SOAP tool from file (I).....	74
Figure 112: Flspace SOAP tool from file (II).....	75
Figure 113: Flspace SOAP tool from URL (I)	75
Figure 114: Flspace SOAP tool from URL (II)	76
Figure 115: Flspace Widget Uploader	77
Figure 116: Flspace Widget generation results.....	77
Figure 117: Flspace Project Wizard	78
Figure 118: Flspace Project Wizard Form.....	79
Figure 119: Waiting for Flspace project creation	79
Figure 120: Flspace Proxy Settings	80
Figure 121: Flspace Widget project.....	81
Figure 122: Flspace Widget project structure	81
Figure 123: Flspace Compact project	82
Figure 124: Flspace Compact project structure	82
Figure 125: Flspace Decoupled project	83
Figure 126: Flspace Decoupled project structure	84
Figure 127: Flspace Node.js project	84
Figure 128: Flspace Node.js project structure.....	85
Figure 129: Flspace JavaFX project.....	86
Figure 130: Flspace JavaFX project structure	86
Figure 131: Flspace RCP project	87
Figure 132: Flspace RCP project structure	87
Figure 133: Launching Flspace RCP project.....	88
Figure 134: Default RCP project GUI	88
Figure 135: Flspace XML-editor.....	89
Figure 136: Flspace XML Editor Form Platform preferences form.....	89
Figure 137: Flspace XML Editor Form Platform State Properties form.....	90
Figure 138: Flspace XML Editor Form Platform Wiring form	90
Figure 139: Flspace XML Editor Text	91
Figure 140: Logging Settings.....	92
Figure 141: Flspace Logging View.....	93
Figure 142: Configuring projects to use Logging Plugin	94
Figure 143: Flspace Logging preferences (defaults).....	94

Figure 144: Flspace Logging log server95
 Figure 145: Flspace Logging filtering by Username.....95
 Figure 146: Flspace Studio available versions96
 Figure 147: Flspace Studio splash screen96
 Figure 148: Flspace About Dialog96
 Figure 149: Flspace Intro page97
 Figure 150: FlspaceStudio Tool.....97

List of Tables

Table 1: Flspace SDK resources and online documentation13
 Table 2: Other Flspace and FIWARE resources.....14
 Table 3: Wirecloud online documentation.....14
 Table 4: Store online documentation.....15
 Table 5: External development tools references15
 Table 6: Bitbucket collaborative environment for Flspace development.....16

1 Introduction

The aim of this document is to describe the main functionalities of the Flspace Software Development Kit (SDK) presented as a user and developer guideline related to the environment of development and corresponding Integrated Development Environment (IDE) based on Eclipse [20] SDK tools for Java. This document describes the functionalities, wizards, Flspace Eclipse plugins developed and provided by Flspace SDK to facilitate the tasks of the developers through a set of graphical user interface (GUI) which allows the developer to interact with the tools, following the necessary steps to achieve the desired work or carry out some tasks and testing purposes.

It is foreseen for the next releases, a chapter for the **libraries** that are being implemented in several programming languages, in order to provide the Flspace user with the appropriate tools to connect to the platform.

This document is an annex of the main deliverable D200.7 “Flspace Integrated Release V3” in addition to other documents and resources related to the Flspace Project ([7][8][9][10][11][12][13][14]). The online documentation for SDK is also available in the following web site: <http://dev.flspace.eu/doc/wiki/sdk>

The following sections are dedicated providing a user guideline and describing the steps using the features or functionalities related to each plugin which conform the Flspace SDK. It is structured as follows

- A short description of the plugin and corresponding prerequisites.
- A set of pictures displaying the user interaction with the Graphical User Interface (GUI) by using the corresponding plugin.
- A set of pictures and descriptions presenting the steps to carry out the desired functions as well as providing indications and settings at user level.

1.1 Scope

Flspace SDK is a collection of plugins developed for the Eclipse platform which together offers an IDE for the development of applications in Flspace. In this environment it's possible to edit different codes, scripts with syntax highlight, code completion, compilation errors, code and comment folding, script execution with pre-configured interpreters, alongside with the tools which are already provided by the Eclipse platform.

Seventeen Flspace plugins have been already implemented. The **FlspaceStudio Tool** is also available. This binary distribution of Flspace SDK is represented by an Eclipse Rich Client Application. The distribution provides the required user interface for all of the pre-installed Flspace SDK functionalities along with a customised approach of the Eclipse IDE reaching the specific needs of Flspace platform.

Several **libraries to access Flspace platform** are being implemented in different programming languages (C#, JavaScript and Python libraries are in progress. PHP library is also planned). The aim of these libraries is to provide the corresponding access to Flspace services based on OAuth2 protocol [26].

The following plugins are available for the application developer:

- **Flspace Preferences:** special dialog pages to indicate how FlspaceStudio is going to operate.

- **Flspace Perspective:** defines the initial set of Flspace collection and arrangement of views, and its editor area.
- **Flspace Views:** the set of views are provided with the FlspaceStudio.
- **Create a project:** a Flspace wizard has been implemented. Five archetypes have been defined to generate the different Flspace projects.

Exclusively for projects based on the Widget nature (widget projects and widget part in decoupled projects), and once the development of the widget is completed, the user will be able to validate the project and then to upload it to Flspace Store.

- **Generate project documentation:** provides the Java doc generation by Flspace using maven ([21]) capabilities in Eclipse.
- **Using the Acsi Editor - BCO (Business Collaboration Objects):** provides the ACSI (Artifact-Centric Service Interoperation [24]) editor tool to Flspace developers. ACSI editor is a web based tool used by BCM module of B2B component for the creation of business entities.
- **Using the Proton Editor – EPM:** to access to PROTON (IBM Proactive Technology Online) [25], a web-based tool used by EPM module of B2B component for the creation of partners and the corresponding events.
- **A File Editor for the Widget Config.xml :** allows the user to define its app config properties in a friendly way.
- **The SDI Connector with the SDI Rest Tool:** allows the final user to experiment with SDI module *locally*. This feature is provided with an Eclipse view, allowing end-users to test different services in different machines, not only focusing on SDI connector plugin.
- **Publish and Subscribe Messages:** shows the capabilities of information transmission from different layers and systems through RabbitMQ (open source message broker software that implements the Advanced Message Queuing Protocol –AMQP-).
- **A Logging Plugin:** enables users to store, and classify the logs generated by Flspace apps in order to make easy the app development process. Now, a back-end to store all generated logs in a database is provided. (This back-end is developed in Nodejs and database in MongoDB [28])
- **Transforming File Formats:** several standard/widely-used data formats can be interchanged by means of this plugin.
- **Sketches plugin:** allows the user to draw and make some simple diagrams that can be useful to depict icons, flow diagrams, component analysis, etc.
- **REST client applications:** generates REST clients for the application developers in an easy way.
- **SOAP client applications:** for generating SOAP clients in an automatic and transparent way for the end user.
- **Flspace Graphics View:** this plugin allows the app developer/designer to use in a graphical way the next Flspace plugins: REST/SOAP clients, Publish-

er/Consumer facilities, and the Flspace Data Transformer. *Widget* graphic management is ready for apps being uploaded.

- **Import plugin:** new plugin implemented for importing Flspace projects (widget and decoupled ones).
- **Export plugin:** new plugin implemented for exporting Flspace projects.

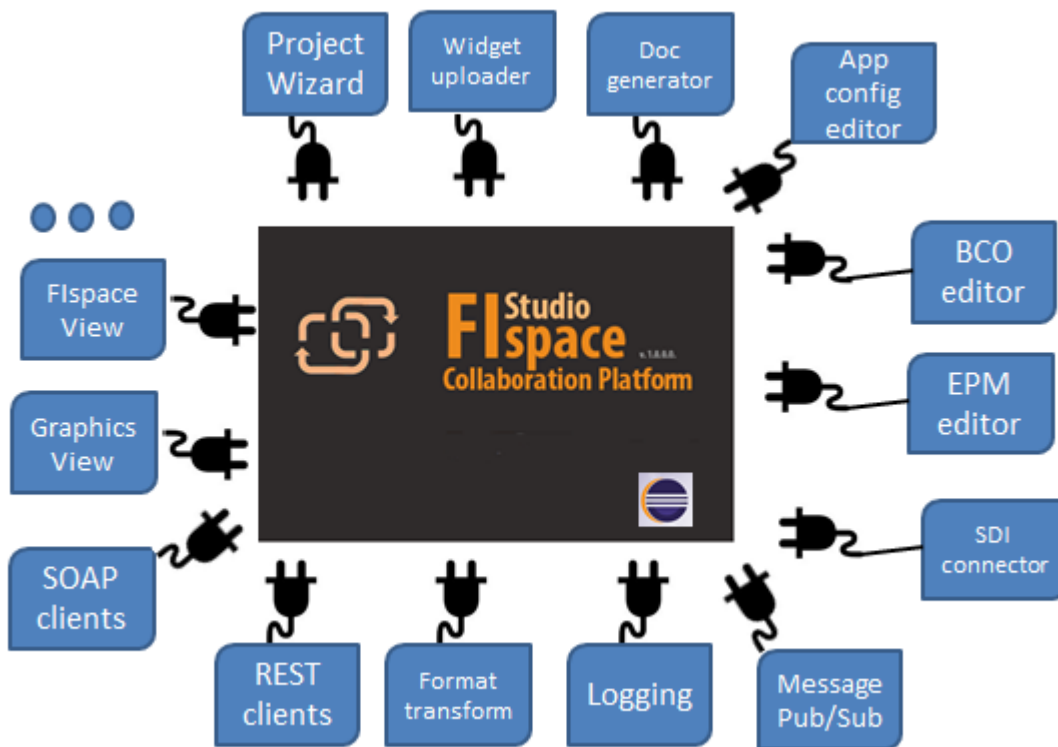


Figure 1: Flspace SDK features overview

Table 1 shows the links to the SDK resources and online documentation.

Description	Link
Flspace App Developer Intro web site	http://dev.fispace.eu/doc/wiki/App%20Developer%20Intro
Flspace SDK Guide web site	http://dev.fispace.eu/doc/wiki/sdk
Flspace apps for newbies	https://bitbucket.org/fispace/apps/wiki/Flspace%20apps%20for%20newbies

Table 1: Flspace SDK resources and online documentation

Table 2 shows the links to other online resources related to FIspace project and FIWARE.

Description	Link
FIspace Business collaboration web site	http://www.fispace.eu/
FIspace Developer Documentation web site	http://dev.fispace.eu/doc/wiki/Home
FIspace Deliverables web site	http://www.fispace.eu/deliverable.html
FIspace Tutorial web site	http://www.fispace.eu/tutorials.html
FIWARE web site	http://www.fi-ppp.eu/projects/fi-ware/
FIWARE Catalogue of the Generic Enablers (GEs)	http://catalogue.fi-ware.org/
FIWARE community web site	http://www.fi-ware.org/community/

Table 2: Other FIspace and FIWARE resources

Table 3 shows the links to the Wirecloud online documentation.

Description	Link
FIWARE - Catalogue - Application Mashup - Wirecloud	http://catalogue.fi-ware.org/enablers/application-mashup-wirecloud
FIWARE - Catalogue - Application Mashup - Wirecloud Documentation	http://catalogue.fi-ware.org/enablers/application-mashup-wirecloud/documentation
FIWARE - Application Mashup - Wirecloud - User and Programmer Guide	https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Application_Mashup_-_Wirecloud_-_User_and_Programmer_Guide
Dashboard - Wirecloud home page	http://conwet.fi.upm.es/wirecloud/
Dashboard - The WireCloud Mashup Platform	http://conwet.fi.upm.es/docs/display/wirecloud/The+WireCloud+Mashup+Platform
Dashboard - Welcome to CoNWeT-Wirecloud Confluence	http://conwet.fi.upm.es/docs/dashboard.action
Dashboard - User Guide	http://conwet.fi.upm.es/docs/display/wirecloud/WireCloud+User%27s+Guide
Dashboard - WireCloud Installation and Administration Guide	http://conwet.fi.upm.es/docs/display/wirecloud/Wire-Cloud+Installation+and+Administration+Guide

Table 3: Wirecloud online documentation

Table 4 shows the links to the WStore online documentation.

Description	Link
FIWARE - Catalogue - Store - WStore	http://catalogue.fi-ware.org/enablers/store-wstore
FIWARE - Catalogue - Store - WStore Documentation	http://catalogue.fi-ware.org/enablers/store-wstore/documentation
FIWARE - Store - W-Store - User and Programmer Guide	https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Store_-_W-Store_-_User_and_Programmer_Guide
FIWARE - Store - W-Store - Store - W-Store - Installation and Administration Guide	https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Store_-_W-Store_-_Installation_and_Administration_Guide

Table 4: Store online documentation

Table 5 shows the external development tools references.

Description	Link
Java Environment, JVM, JRE, JDK (Oracle)	http://www.oracle.com/technetwork/java/javase/downloads/index.html
Eclipse IDE (Integrated Development Environment)	https://www.eclipse.org/ , https://www.eclipse.org/downloads/
Maven	http://maven.apache.org/ , http://maven.apache.org/download.cgi

Table 5: External development tools references

Table 6 shows the Flspace development repository and documentation references based on the bitbucket tools for collaborative development.

Bitbucket is a hosting site for the distributed version control systems (DVCS) Git (<http://git-scm.com/>) and Mercurial (<http://mercurial.selenic.com/>). The service offering includes an [issue tracker](#) and [wiki](#), as well as integration with a number of popular [services](#) such as Basecamp, Flowdock, and Twitter.

Description	Link
Bitbucket Flspace repository home page	https://bitbucket.org/flspace

Bitbucket FIspace core component home page	https://bitbucket.org/fispace/core/wiki/Home
Bitbucket FIspace Roadmap page	https://bitbucket.org/fispace/core/wiki/roadmap

Table 6: Bitbucket collaborative environment for FIspace development

This document is the initial guideline for the SDK. More updated information can be found on the FIspace online documentation.

Online documentation for SDK [12][13]: <http://dev.fispace.eu/doc/wiki/sdk>

1.2 Intended audience

The main interest groups of this document are the software developers or users who are developers of application for FIspace or by extension they are App developers on top of the FIspace platform. App Developers are the actual software and system providers who offer “packaged” / componentized solutions and applications in form of Apps.

2 SDK User/Developer Guide

2.1 Goal and objectives

The Software Development Toolkit (SDK) provides tool-support for the development of Flspace Apps. The SDK will ease the work of App developers during the implementation of the Apps, providing specific tools and hiding the complexity of the platform. Particularly, the SDK will include:

- Tooling (specifically an Integrated Development Environment – IDE), which is built on Eclipse IDE [20]. Eclipse is widely adopted by the development community and supported by the Eclipse foundation. The Flspace SDK will offer functionalities such as (1) integration of Eclipse JDT (Eclipse Java Development Tools) (e.g., classpath containers) or Eclipse PDE (Eclipse Plug-in Development Environment); (2) providing access to Javadoc for all referenced elements (Flspace modules) and auto-completion support; (3) visual management of components and case modelling will be provided.
- Libraries to link with the respective modules of the Flspace, such as security, privacy and trust, or the System and Data Integration layer in Flspace (SDI).

Complementing the SDK, there will be a set of tools targeted to business architects for customizing and extending the Flspace to the individual needs of Users. This includes tools for authoring of Business Entities and Event Rules (see B2B Core), as well as configuring mediators and connectors to backend systems (see System and Data Integration).

2.2 Overview

As an initial precondition it is recommended to start working on a new Eclipse [20]. When trying to install the **Flspace SDK plugins** into older versions of Eclipse, dependency conflicts may occur during installation. Therefore it is highly recommended to download a current version of Eclipse from <https://www.eclipse.org/downloads/>, and to remove your local repository.

Once the Eclipse product is running, the first step will be to install the Flspace plugins using the Flspace update site (<http://37.131.251.110:8080/flspace-updatesite/>): select Help -> Install New Software -> Install the Flspace Update Site.

Figure 2 shows the FIspace SDK screenshot as an overview.

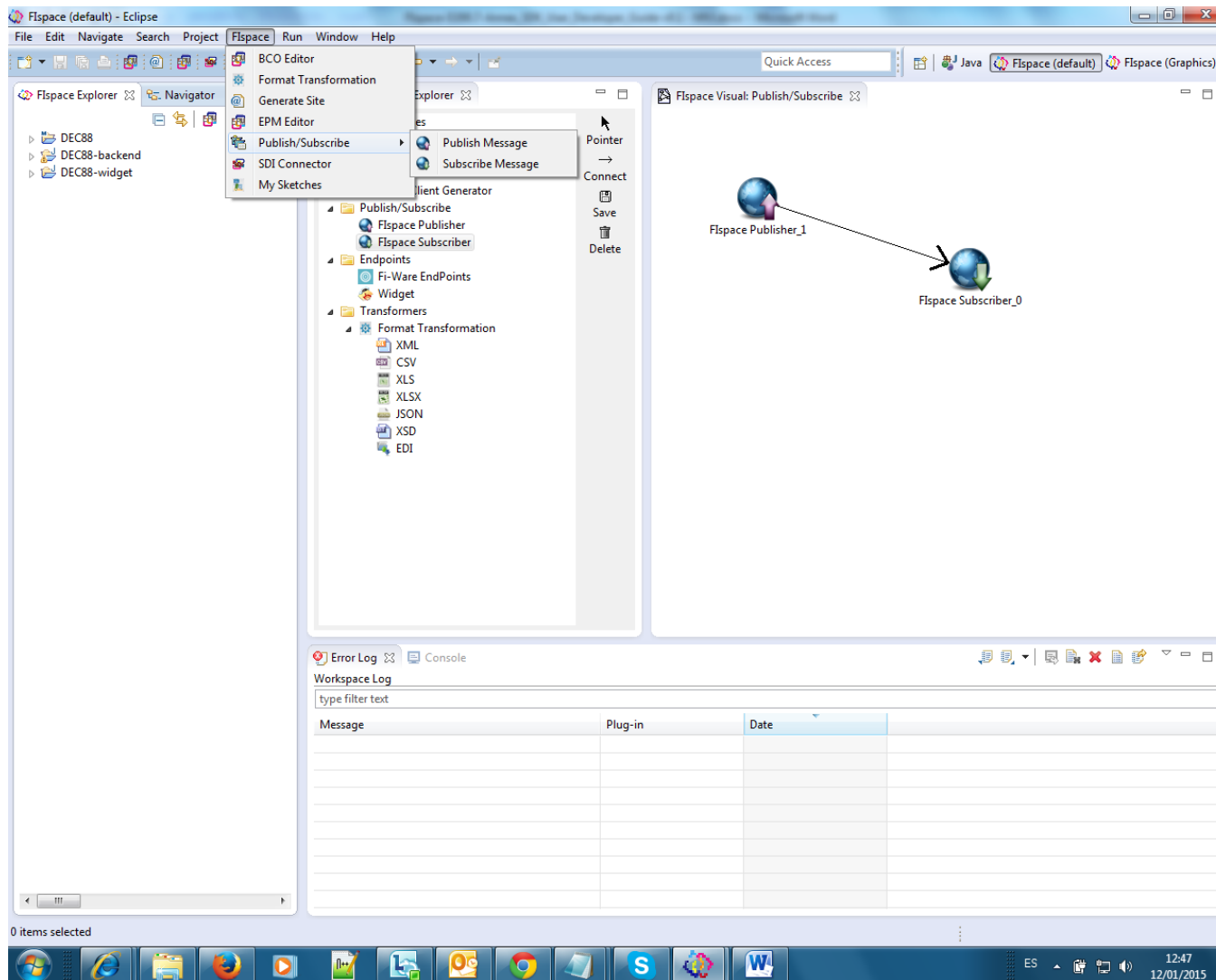


Figure 2: FIspace SDK screenshot overview

2.3 Flspace-bcoeditor Plugin

BCO (Business Collaboration Objects) can be accessed by an icon on the Eclipse IDE toolbar as well as with an entry under “Flspace” menu category.

The user clicks either on the toolbar button or on the menu entry in order to trigger the Eclipse internal browser which automatically loads the ACSI [24] tool web page, as it is shown in the following figures:

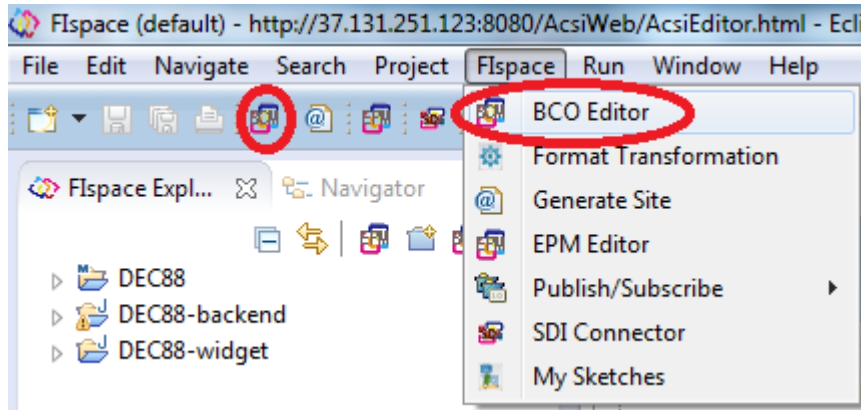


Figure 3: Menu item to access the BCO Editor

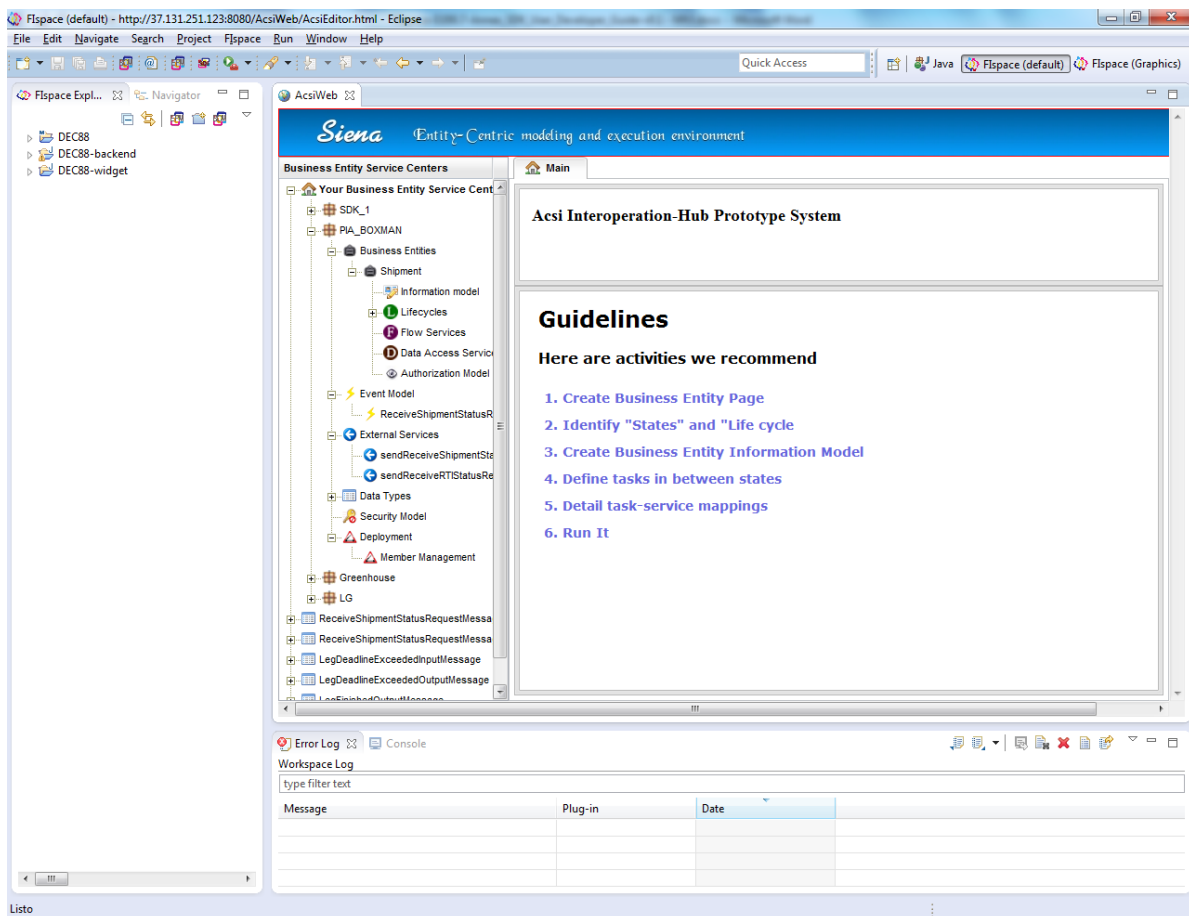


Figure 4: Flspace BCO editor

2.4 FIspace-converter Plugin

2.4.1 Prerequisites

We have the possibility of convert among different file formats like XML, JSON, CSV, XLS/XLSX (Microsoft Excel file Format) and EDI, between others. The only prerequisite is to have one of these documents to convert Copying & Pasting into the text area or uploading the file from the local disk.

2.4.2 Using the plugin

From the FIspace menu bar, select the option Format Converter.

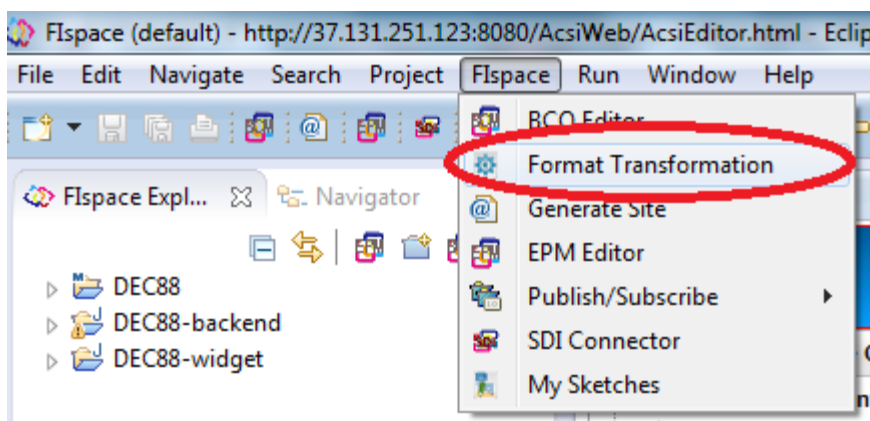


Figure 5: FIspace converter

2.4.3 Converting formats

This tool offers the opportunity of choosing among different formats in order to converting them into others. The source format can be selected from the main combo box.

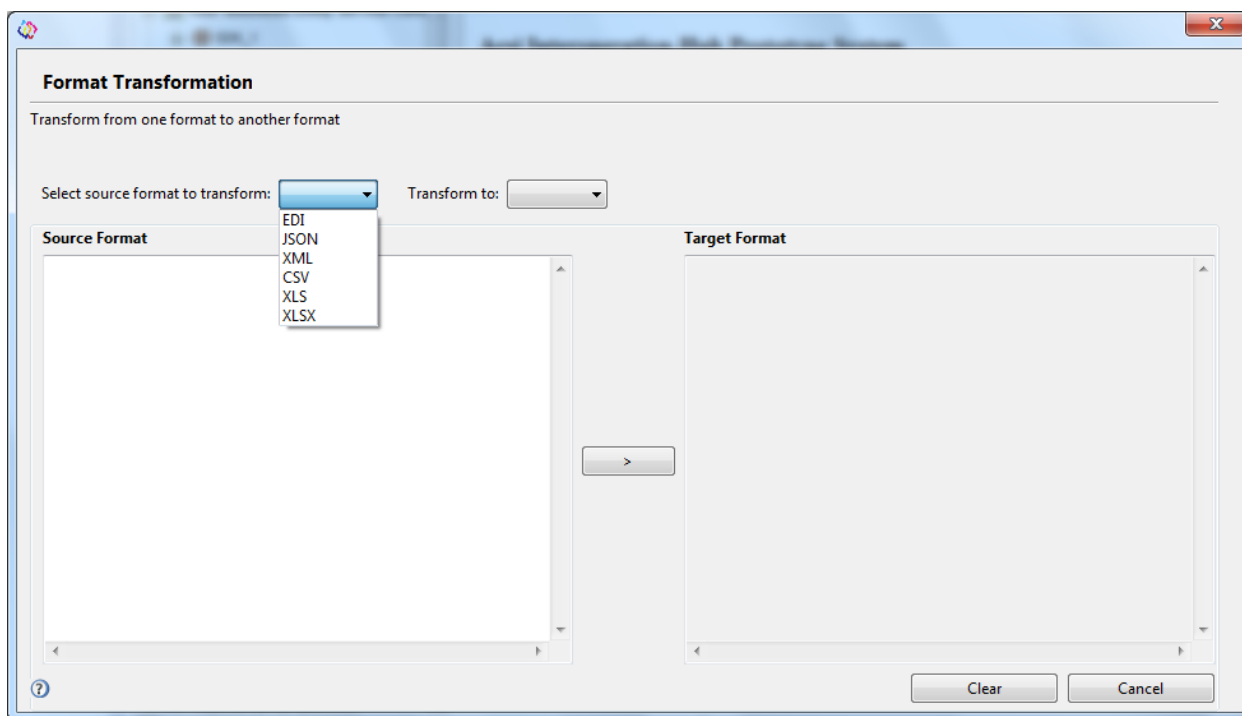


Figure 6: Format converter dialog (I)

Once the source format has been selected, another combo box has to be used for selecting the target format in the “Conversions” section.

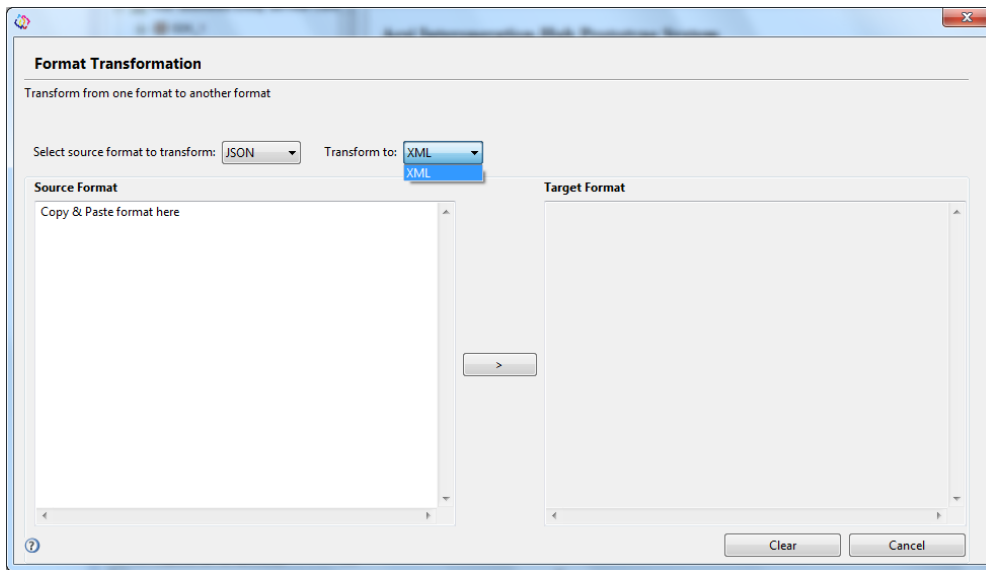


Figure 7: Format converter dialog (II)

In order to make the format transformation, we must copy and paste the data to transform in the text area placed at left.

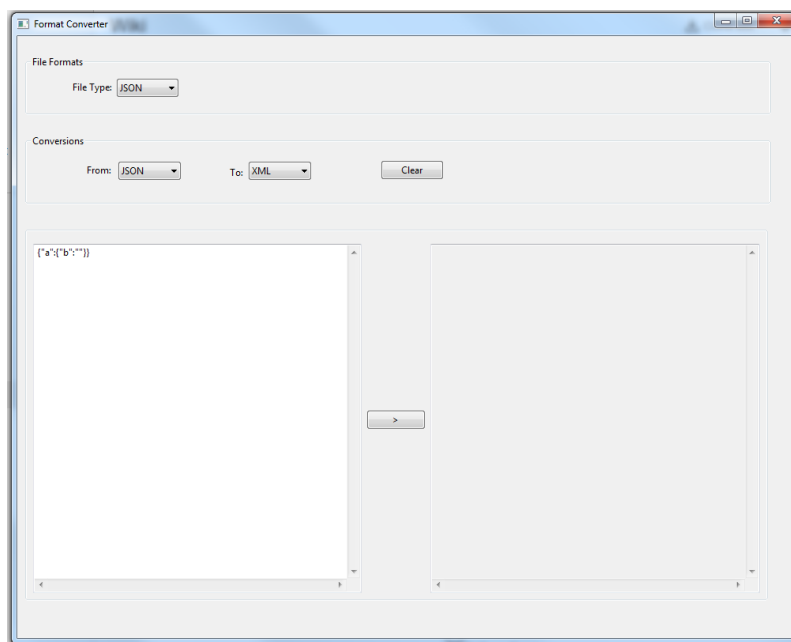


Figure 8: Format converter dialog (III)

At this moment we can push the button to convert the format. The result will appear into the text area placed at the right.

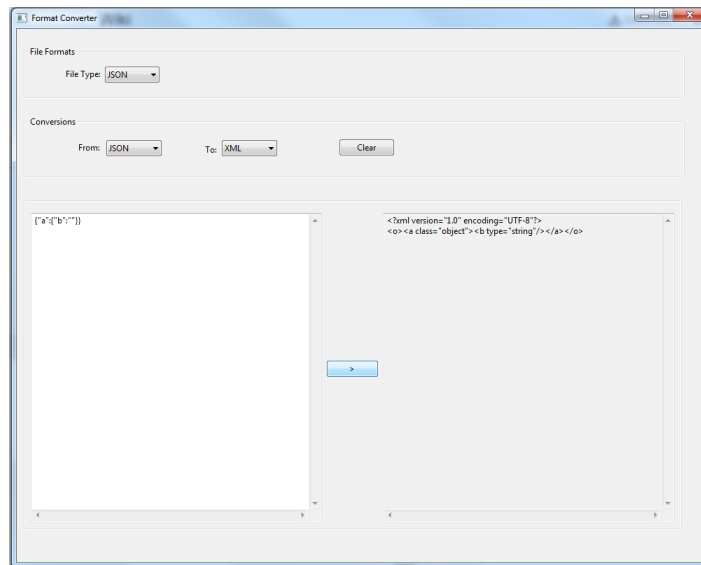


Figure 9: Format converter dialog (IV)

In any moment, we can clear the content of the controls using the Clear button.

The operation to convert Excel formats to other formats works differently. We can't do copy & paste from an Excel file into the text area. For this reason, we need to upload the file from the local directory using the conversion button (arrow button) placed between the text areas. After of this, the file uploaded is converted to the desired format.

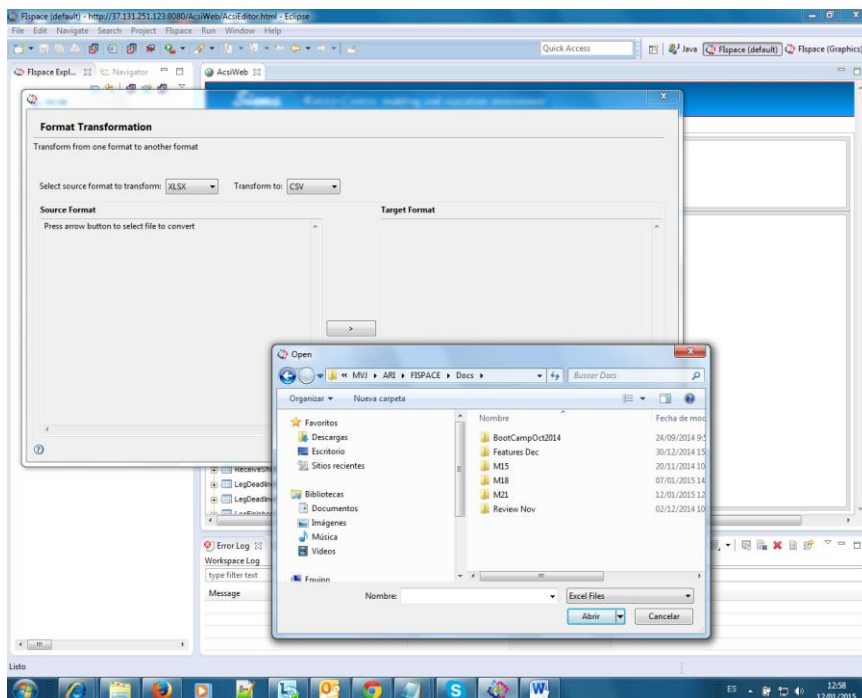


Figure 10: Format converter dialog (V)

The result of the conversion is showed into the text area at right.

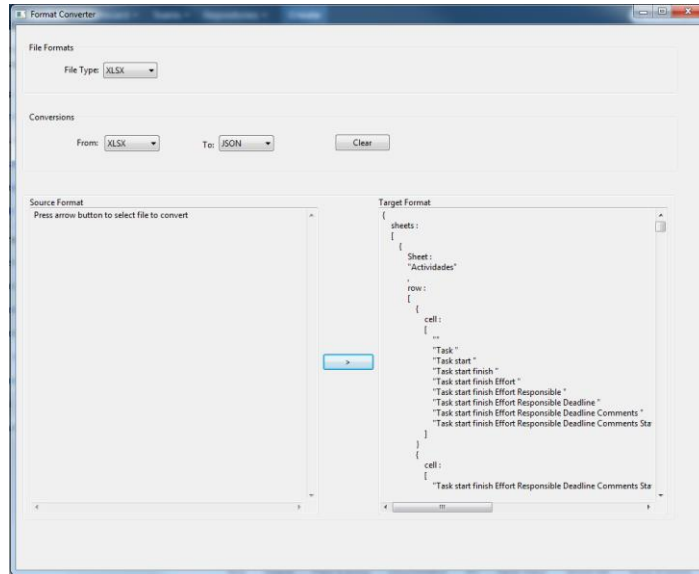


Figure 11: Converting XLSX to JSON

2.5 FISpace-doc Plugin

FISpace Doc is an Eclipse Plugin for automatically adding Maven [21] Site and POM file headers to your default workspace or directly to any folder. It optionally generates initial comments from the elements name by using Velocity templates for Javadoc and file headers.

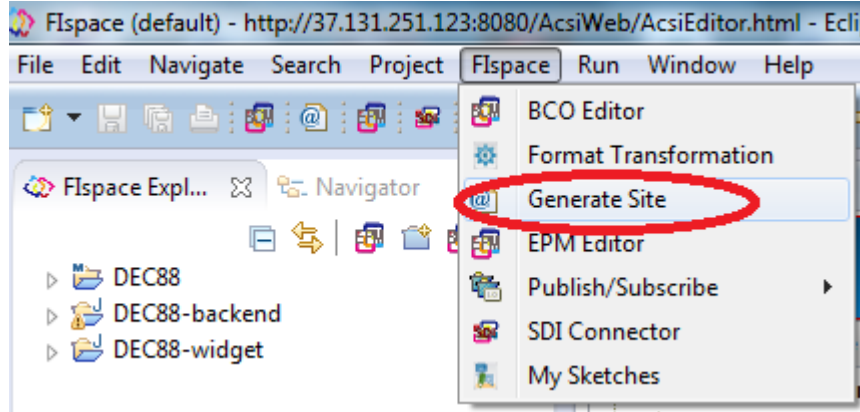


Figure 12: FISpace doc generator

The results are shown in next figure.

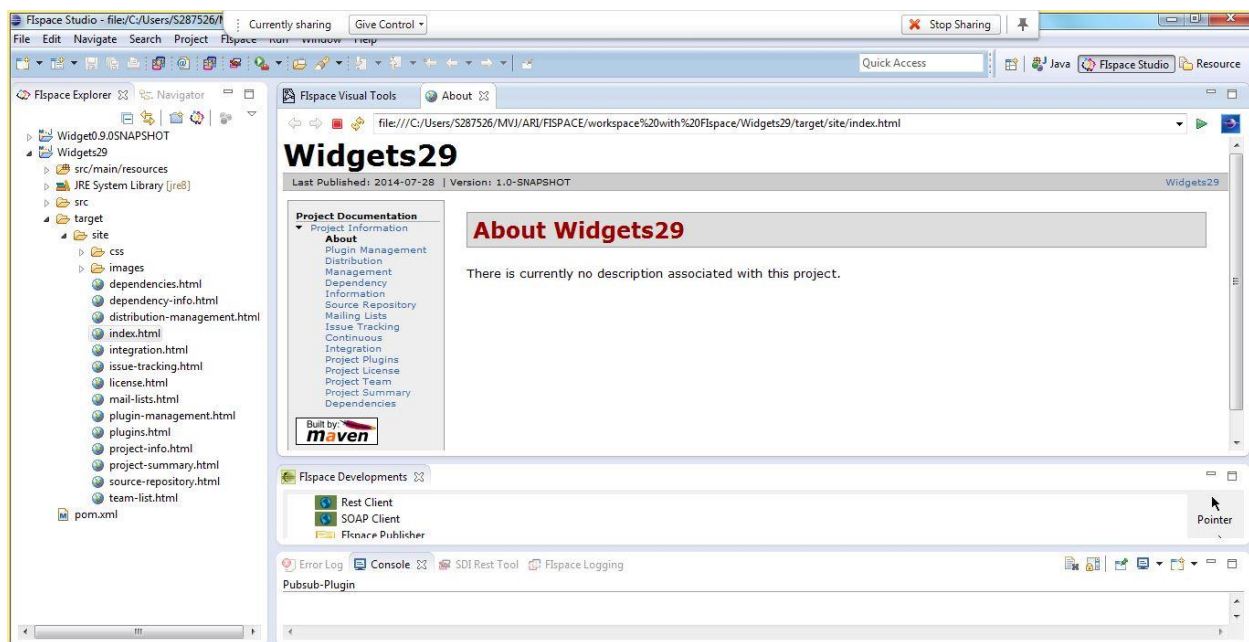


Figure 13: FISpace doc generation results

Setting of the project Workspace.

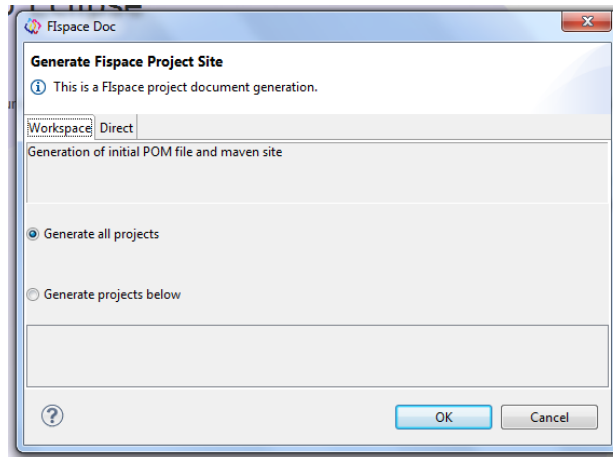


Figure 14: FIspace Doc dialog (I)

Setting of the maven tools related to the project.

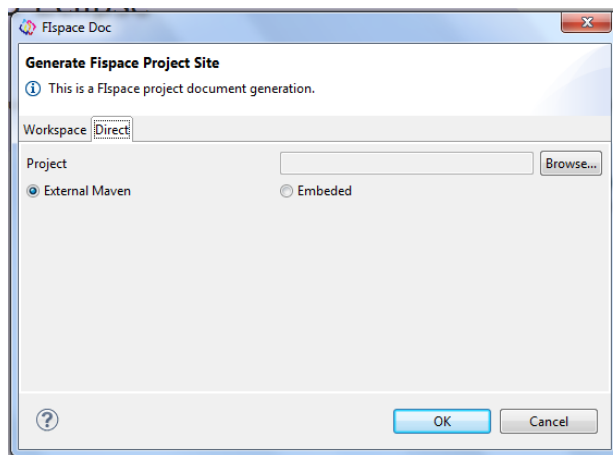


Figure 15: FIspace Doc dialog (II)

FIspace Doc plugin has the following features:

- **Workspace Tab:** Generate the documentation directly of the workspace projects.

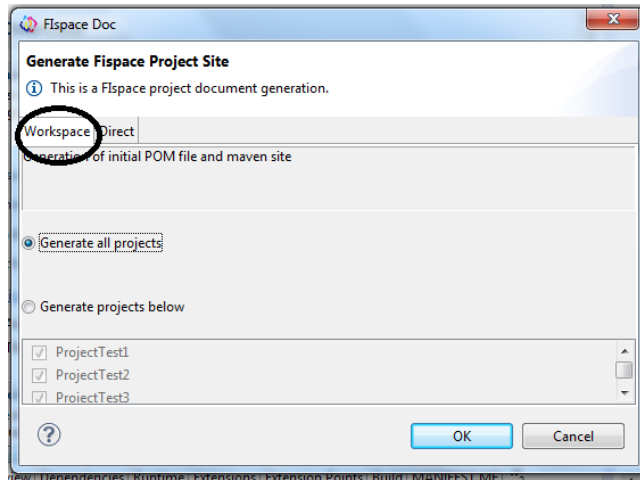


Figure 16: FIspace Doc Workspace tab (I)

- **Generate all projects:** Calls the maven site build on every project.

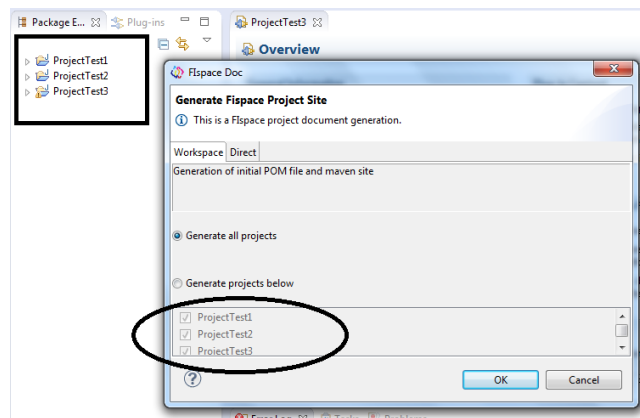


Figure 17: FIspace Doc Workspace tab (II)

- **Generate projects:** Calls the maven site on selected projects.

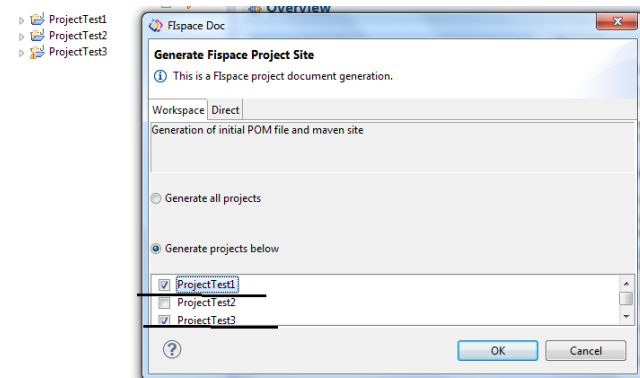


Figure 18: FIspace Doc Workspace tab (III)

- Direct Tab options.

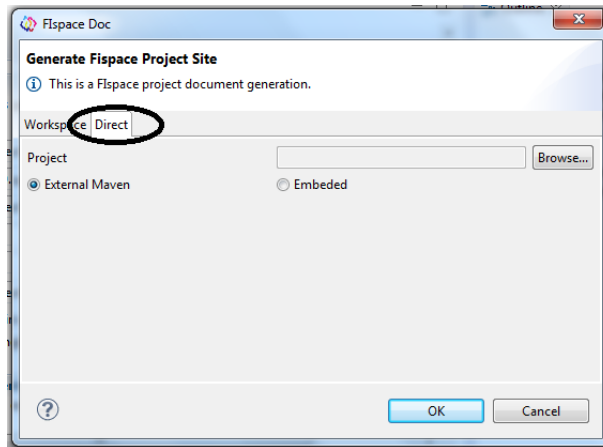


Figure 19: FIspace Doc Direct tab (I)

- Project or Folder: Generate the maven site directly to any folder. Existing site will be totally replaced by the generated.
- Generation Options:
 - External: It generates documentation by External Maven.
 - Embedded: It generates documentation by Embedded Maven.
 - Usage: Browse the project location

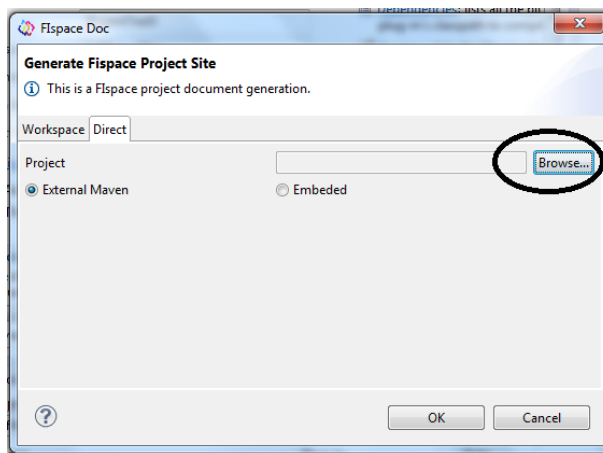


Figure 20: FIspace Doc Direct tab (II)

Action to select the Generation type:

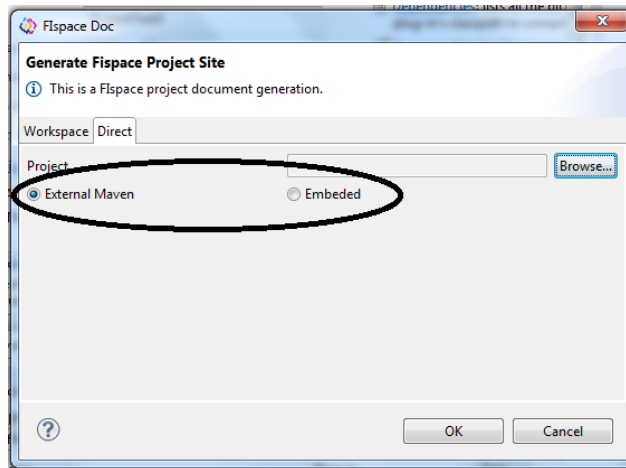


Figure 21: FIspace Doc Direct tab (III)

Press OK Button to validate the options and generate the documentation.

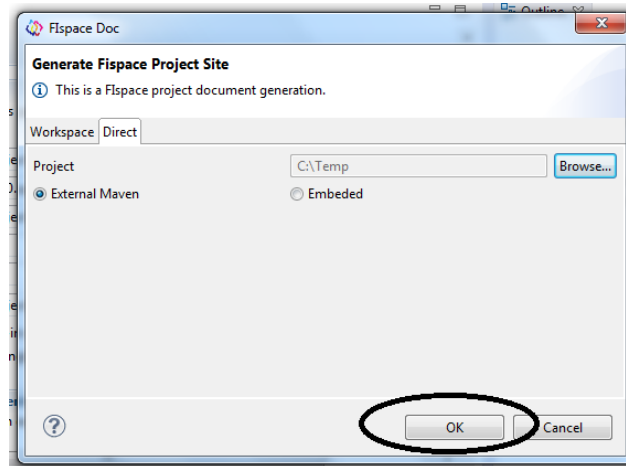


Figure 22: FIspace Doc Direct tab (IV)

If everything was successful the results appear like follows:

- External

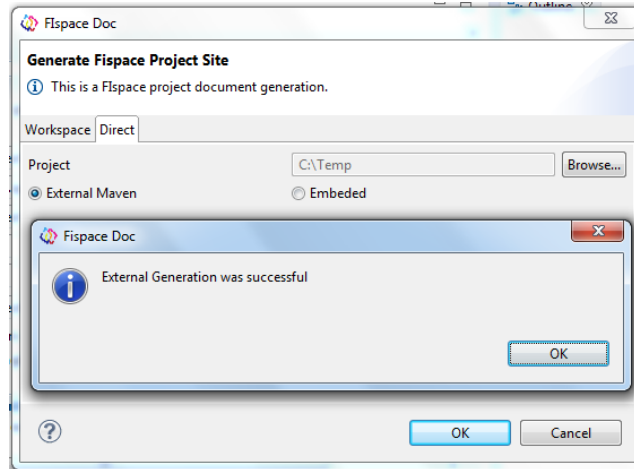


Figure 23: External Generation result

- Embedded

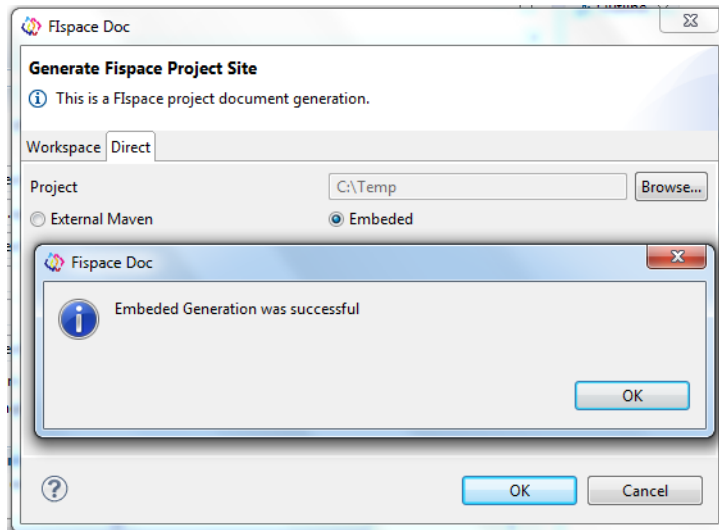


Figure 24: Embedded Generation result

Validations: The "Project" field is mandatory.

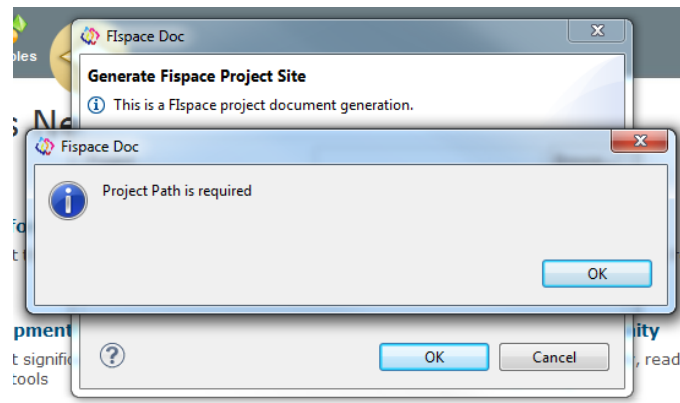


Figure 25: Fispace Doc validations (I)

Set the MAVEN_HOME environment variable to point to the Maven installation for generation with External Maven.

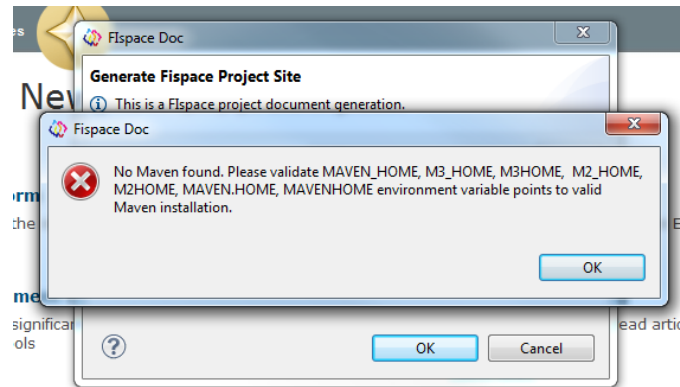


Figure 26: FIspace Doc validations (II)

2.6 FIspace-epmeditor Plugin

EPM editor can be accessed by an icon on the Eclipse IDE toolbar as well as with an entry under “FIspace” menu category.

The user can click either on the toolbar button or on the menu entry in order to trigger the Eclipse internal browser which automatically loads the Proton tool web page [25].

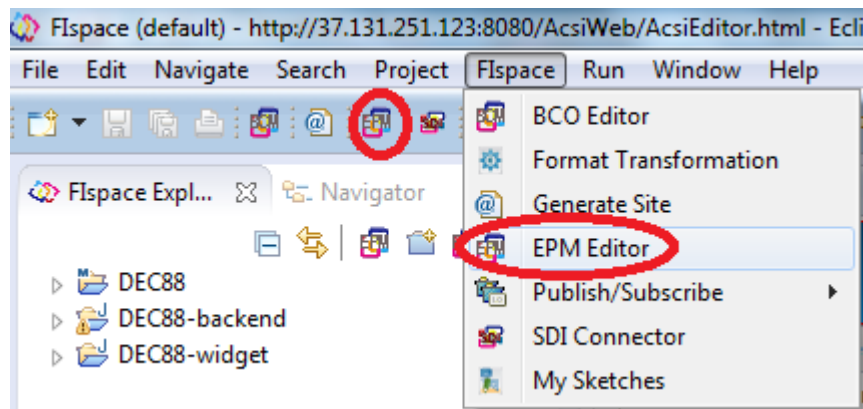


Figure 27: FIspace EPM editor

Notice that the EPM editor could not be opened with the Eclipse internal web browser (more memory resource could be needed). Try instead with an external web browser.

2.7 Flspace-export Plugin

To export an entire Flspace project into an archive or directory, follow these steps:

Select the project then right click over the project:

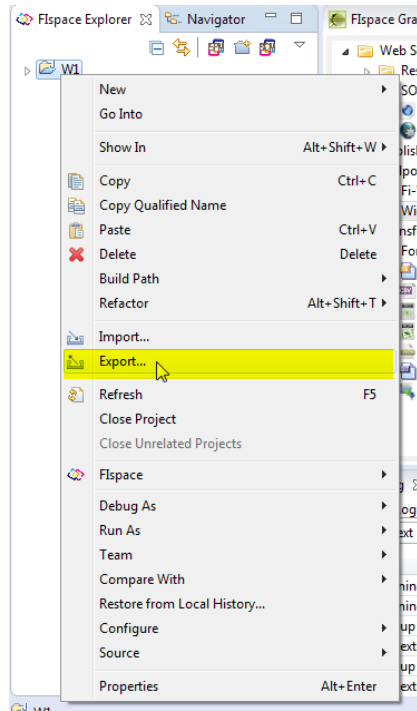


Figure 28: Flspace Export

After clicking over *Export* option, the following dialog will appear:

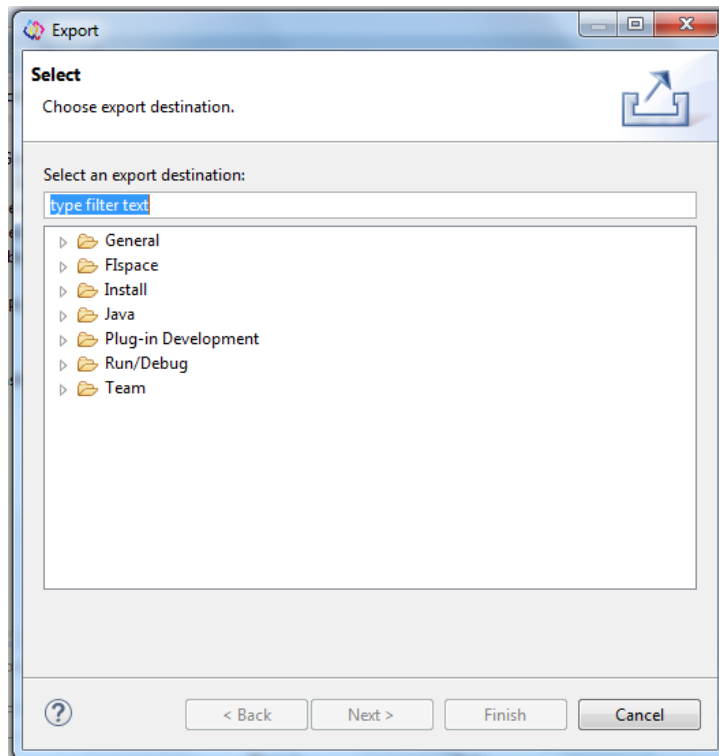


Figure 29: Flspace Export Options (I)

The user should select *FIspace* then *Export FIspace Project*.

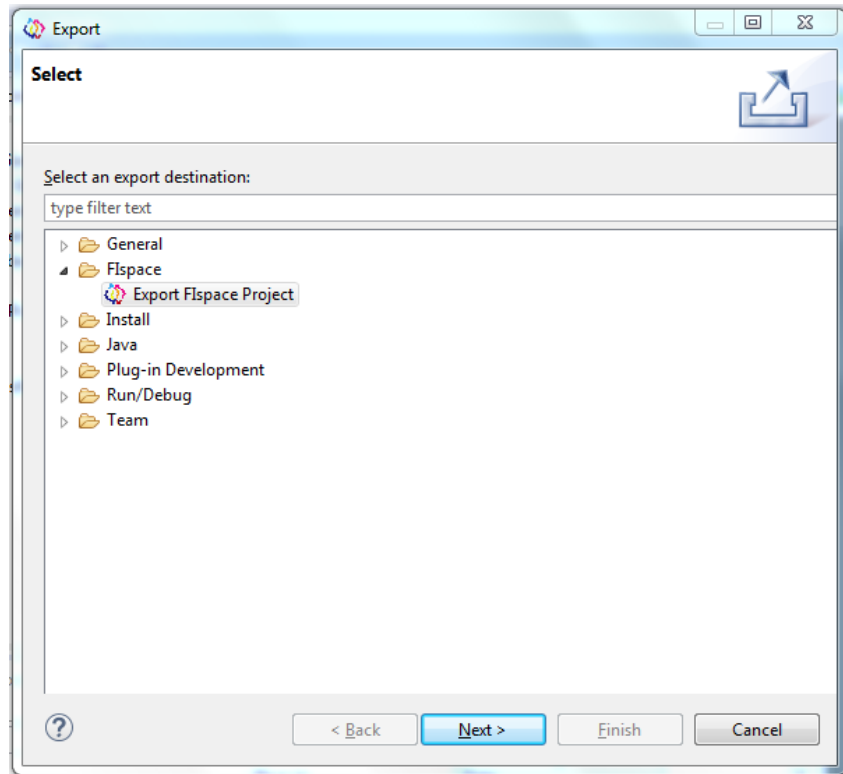


Figure 30: FIspace Export Options (II)

By clicking on button *Next*, a form will appear:

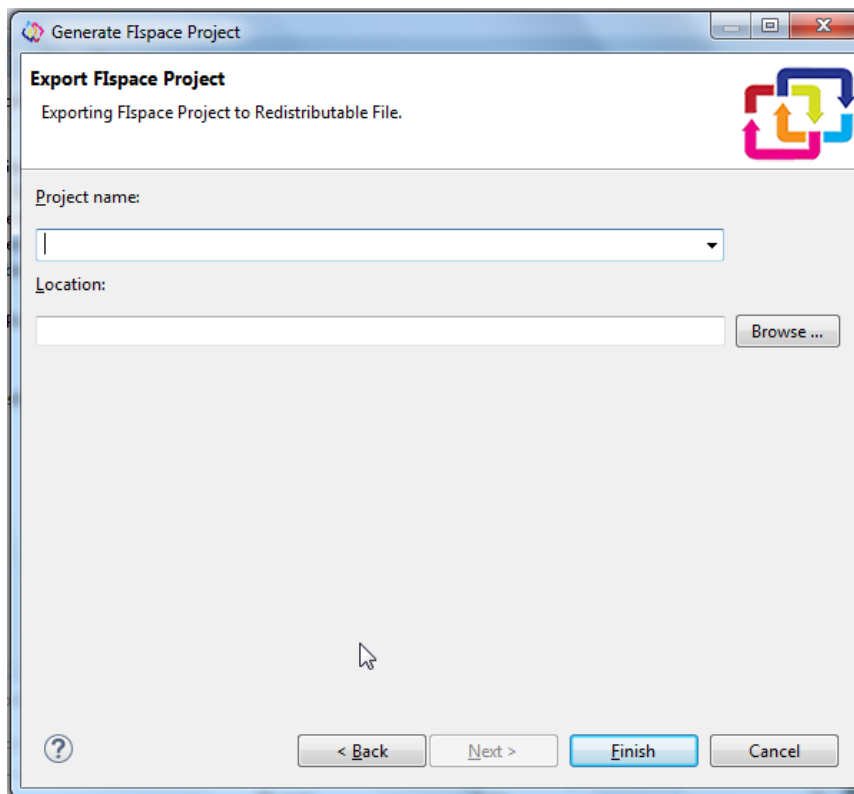


Figure 31: FIspace Export Wizard

The user should fulfill the form then click on button *Finish*:

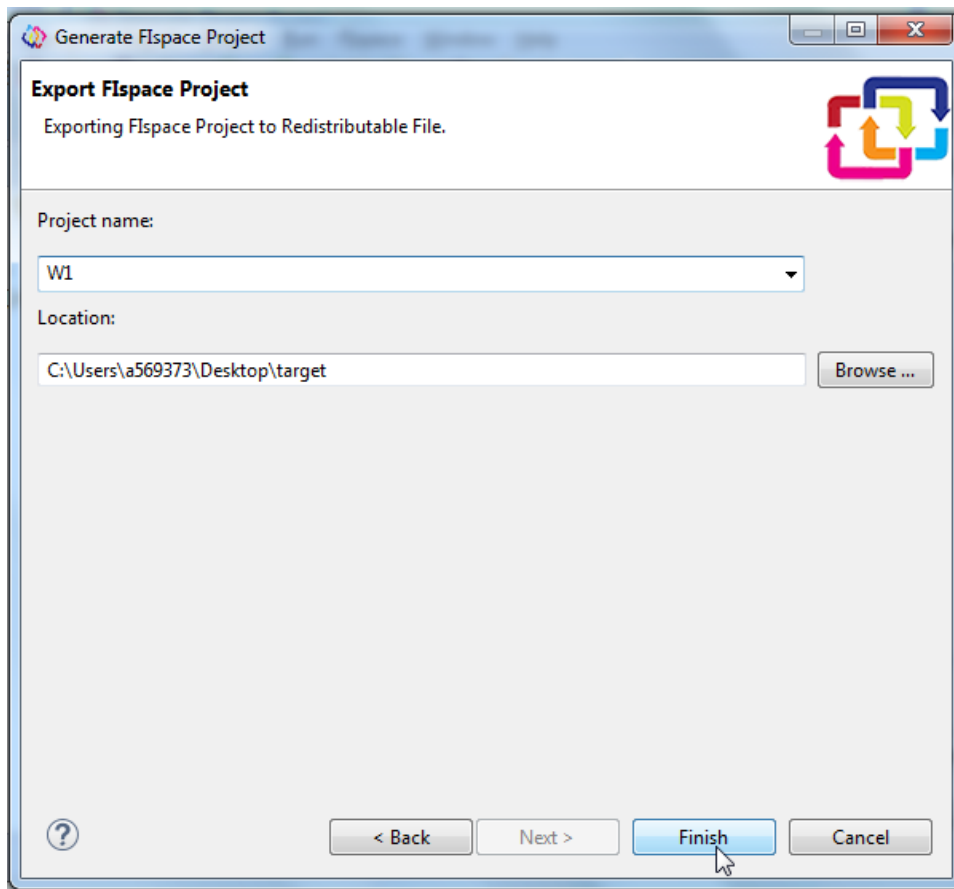


Figure 32: FIspace Project to be exported

2.8 Flspace-graphic Plugin

The Flspace Graphic is a fancy and easy-to-use graphic plugin that will help you to enhance the level of detail of how to use the SDK.

Open the Flspace graphic Perspective by using the standard **Window -> Open Perspective -> Other.Command** and selecting "Flspace (Graphics)" from the list.

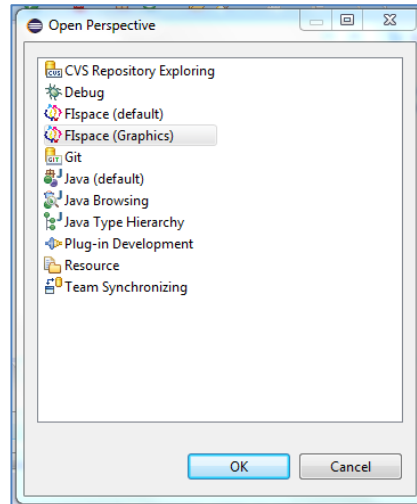


Figure 33: Flspace Graphics perspective

The perspective will open two views (see next figure):

- A "Flspace Graphic" browser view to the left containing different Flspace elements.
- A "Flspace Visual:" canvas view to the right containing the elements dropped from Flspace Graphic Explorer.

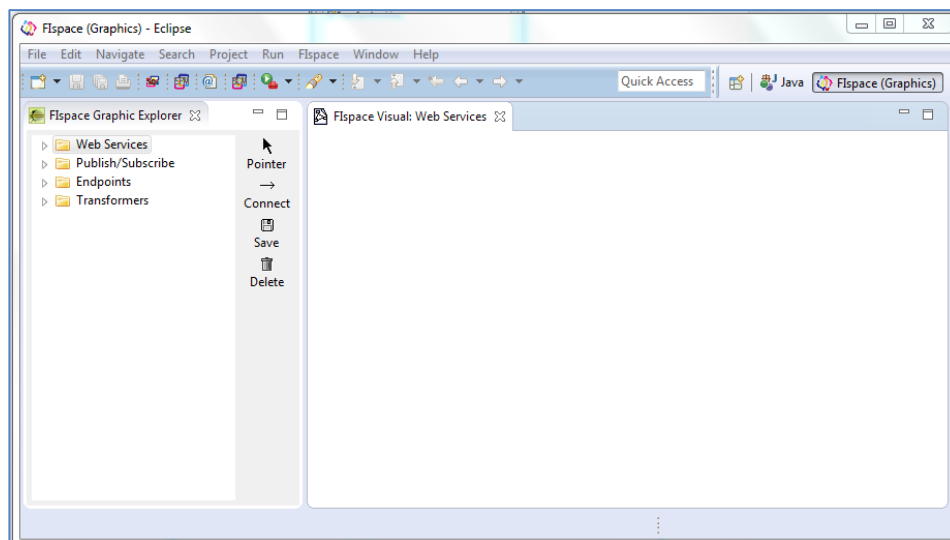


Figure 34: Flspace Graphics Overview

2.8.1 The Browser Pane

This is the browser pane, where you can navigate through different Flspace elements. This pane list all the elements of Flspace as a tree view also the User could drag and drop the elements from the browser to the Canvas and then start to interact with these (see next figure).

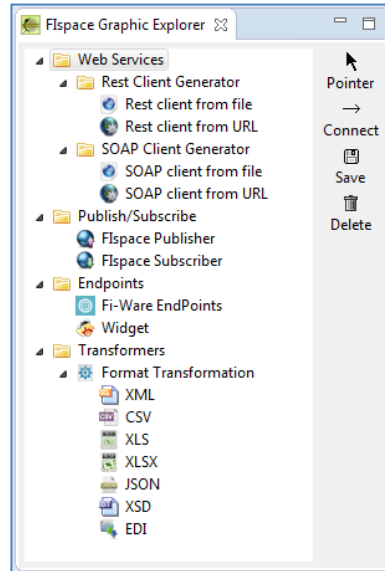


Figure 35: Flspace Graphic Explorer

2.8.1.1 Web Services

This element helps to interact with the REST and SOAP plugins.

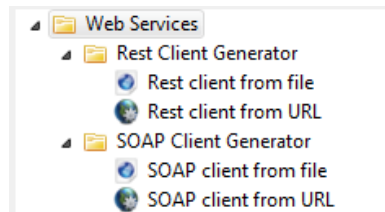


Figure 36: Web Services Element

REST Client Generator

Through this element the user could create, consume and test REST web services.

REST Client from file

Dragging and dropping a REST client from file:

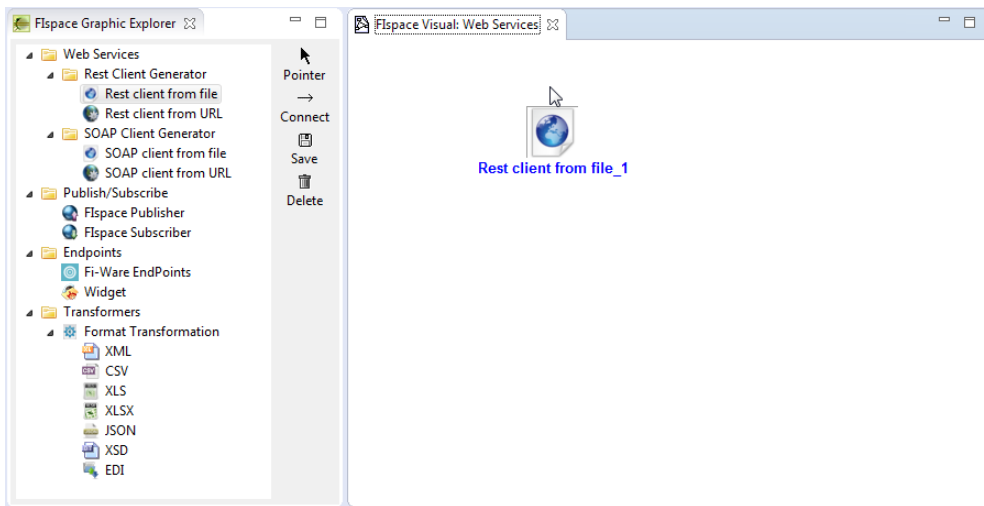


Figure 37: REST client from file

After double clicking over the graphic element came up a form with REST web service details:

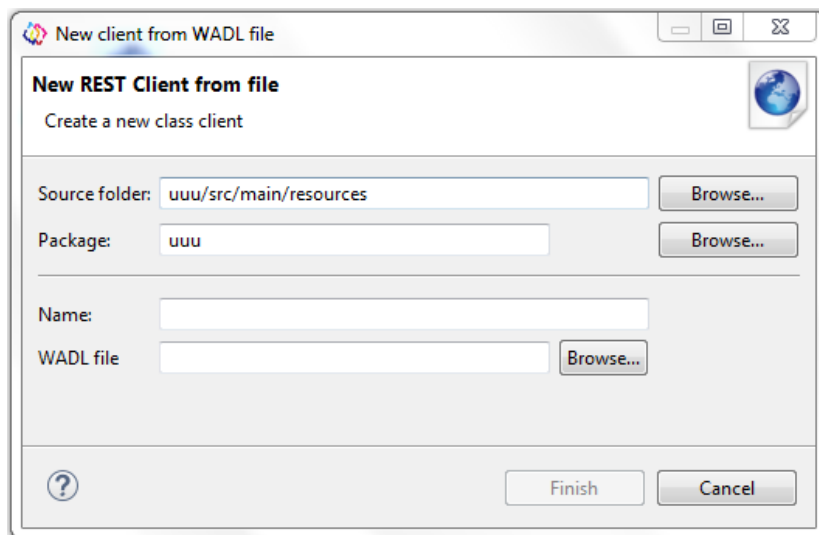


Figure 38: REST client form (I)

After setting all data from the form, Flspace generate the REST web services from the file (WADL Endpoint - Browse).

REST Client from URL

Dragging and dropping a REST client from URL:

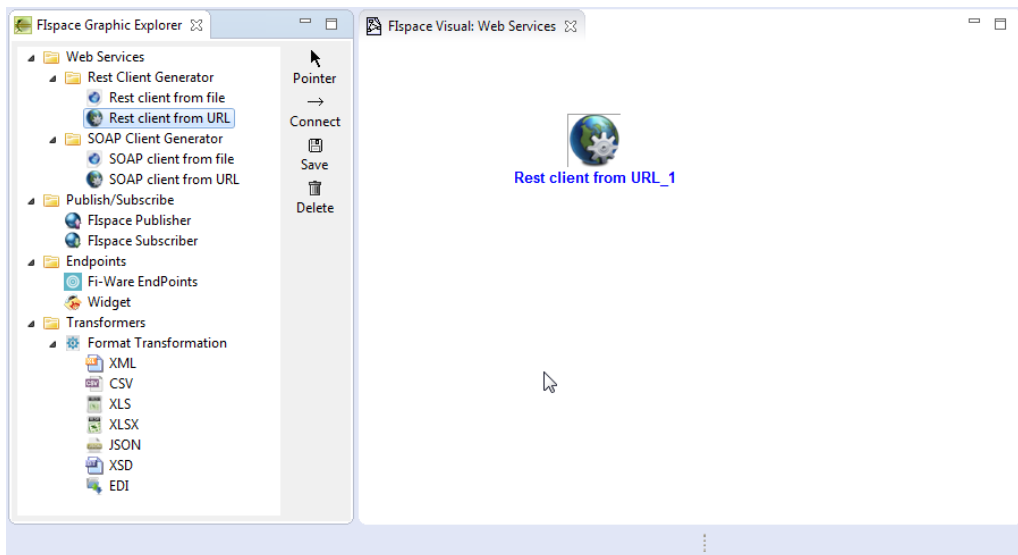


Figure 39: REST client from URL

After double clicking over the graphic element came up a form with REST web service details:

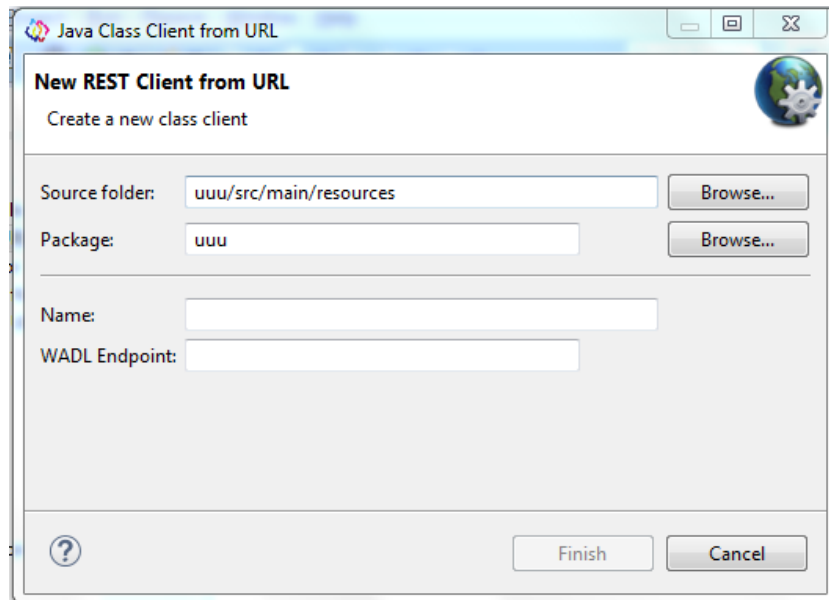


Figure 40: REST client form (II)

After set all data from the form, Flspace generates the REST web services from the URL (WADL Endpoint).

SOAP Client Generator

Through this element the user could create, consume and test SOAP web services.

SOAP Client from file

The picture of the SOAP Client on the Flspace Visual: Web Services:

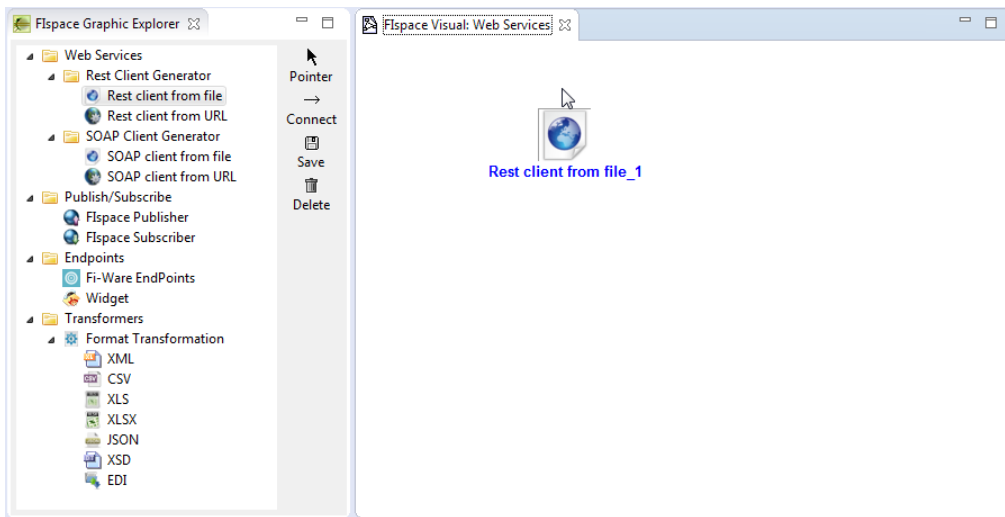


Figure 41: SOAP client from file

After double clicking over the graphic element came up a form with SOAP web service details:

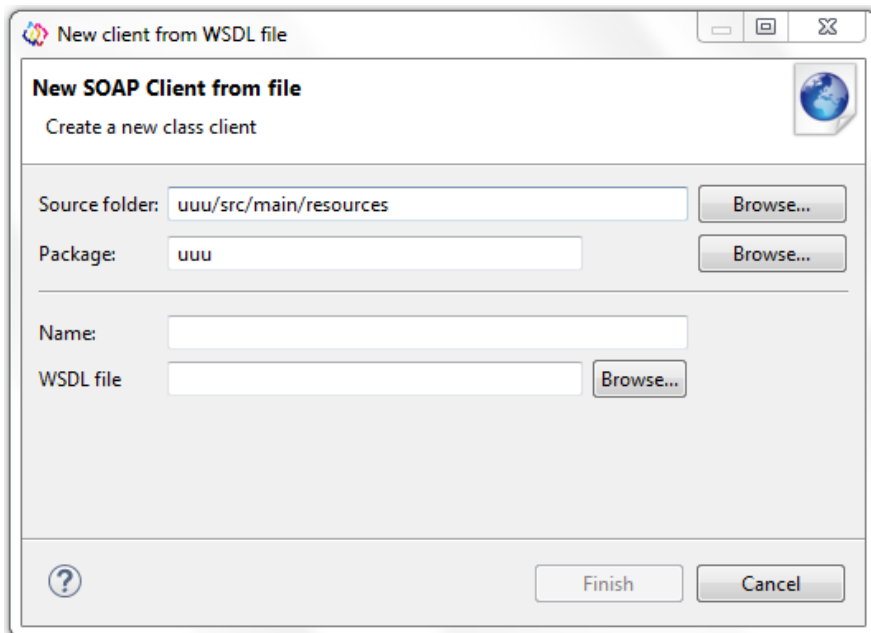


Figure 42: SOAP client form

After setting all data from the form, FIspace generates the SOAP web services from the file (WADL Endpoint - Browse).

SOAP Client from URL

The picture of the SOAP Client on the FIspace Visual: Web Services:

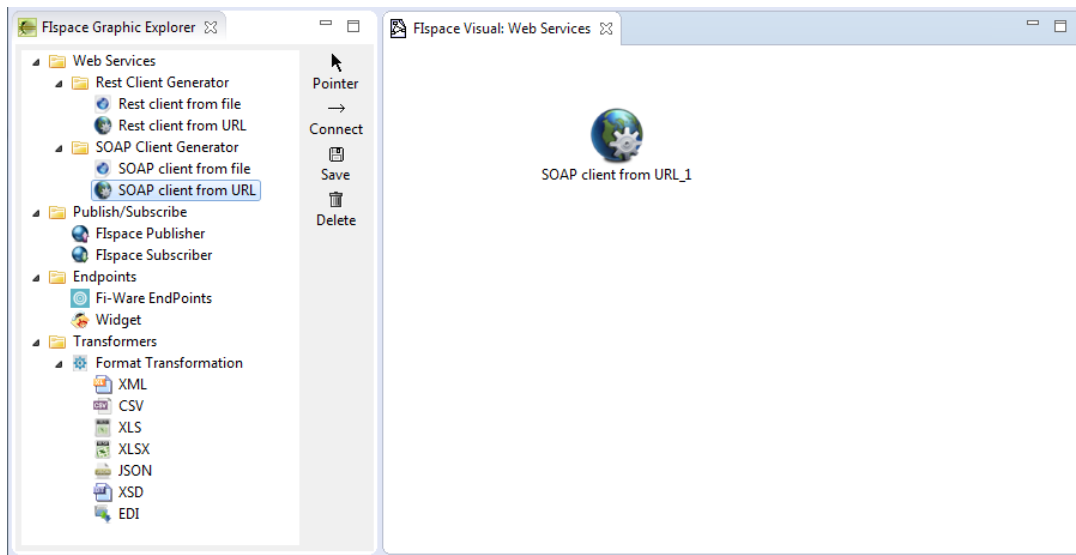


Figure 43: Rest client from URL

After double click over the graphic element came up a form with SOAP web service details:

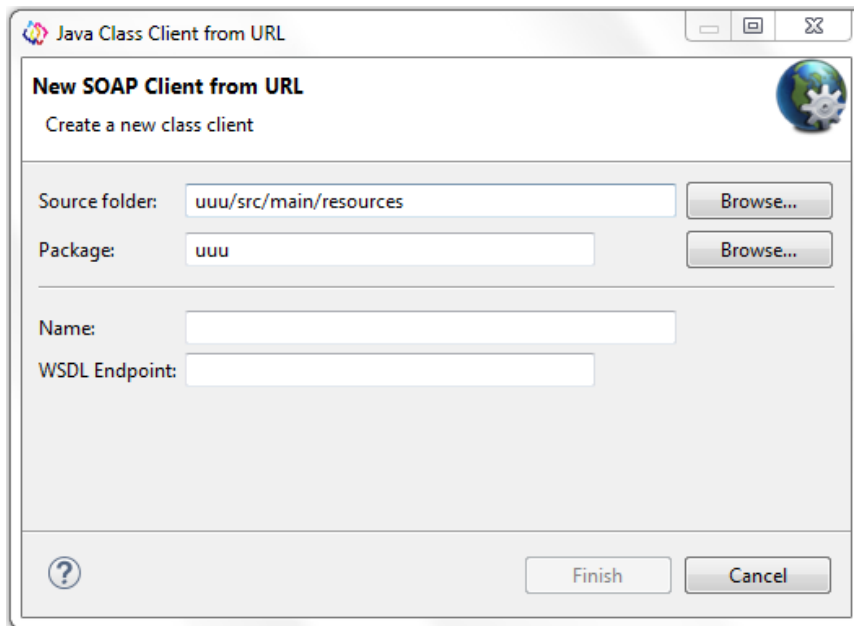


Figure 44: SOAP client form

After setting all data from the form, FIspace generates the SOAP web services from the URL (WADL Endpoint).

2.8.1.2 Publish/Subscribe

The main objective of distributed event-based systems is to decouple the system’s components (producers/consumers) therefore FIspace has added this feature (Publisher and Subscriber).

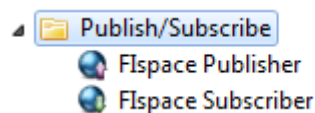


Figure 45: Publish/Subscribe Elements

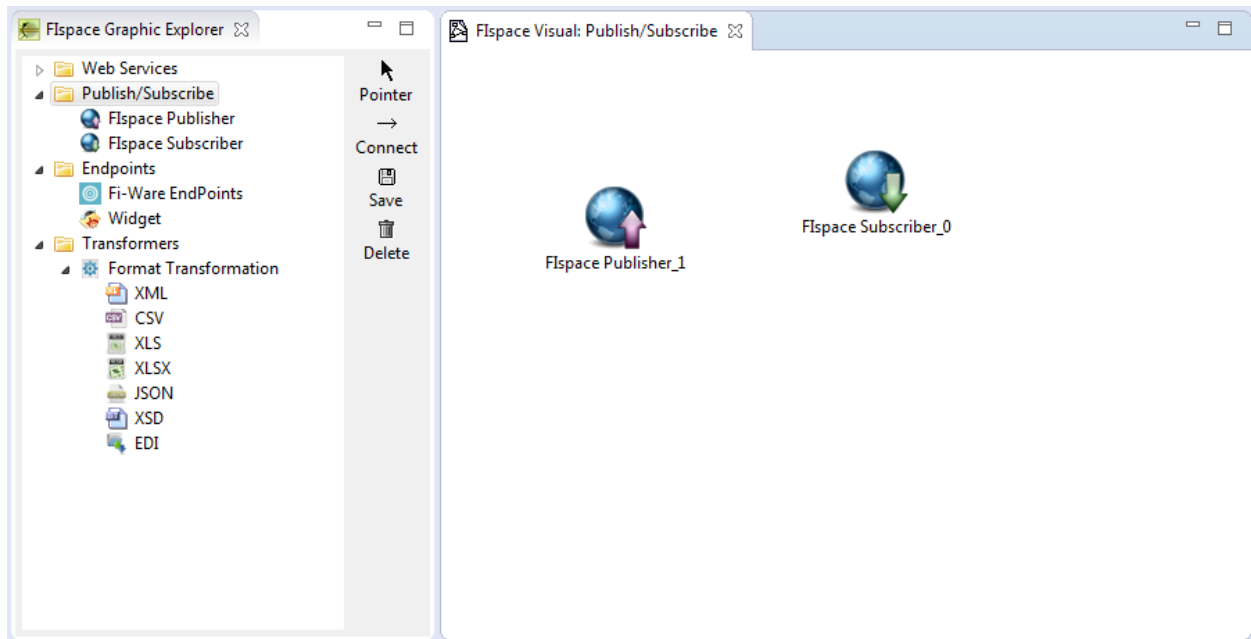


Figure 46: Publish/Subscribe on Flspace Canvas

Prerequisites

Messages are published and subscribed by use RabbitMQ Server. In case of work with a local server is necessary to download and install RabbitMQ from:

<http://www.rabbitmq.com/download.html>

Flspace Publisher

In order to publish the user should double click over the Flspace_Publisher graphic element then a form appears with the parameters details.

Figure 47: Publisher Form

After set all data from the form, the user should send the information to the server (RabbitMQ) by clicking on the Publish button.

Flspace Subscriber

Exist two ways how to subscribe the messages:

Through the Visual Canvas:

Selecting the Publisher and Subscriber by the option connect from the toolbar:



Figure 48: Option Connect

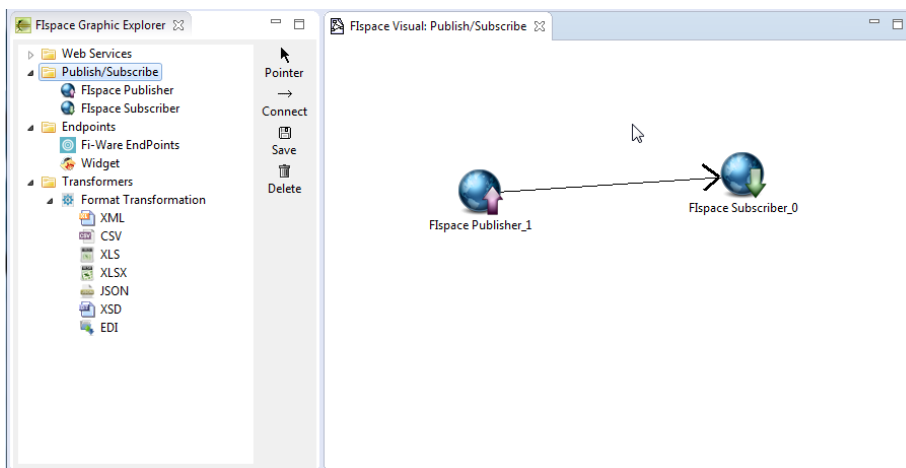


Figure 49: Connect Publisher/Subscriber

After connecting the user should insert the Publisher data then the data in the Subscriber is going to be updated automatically:

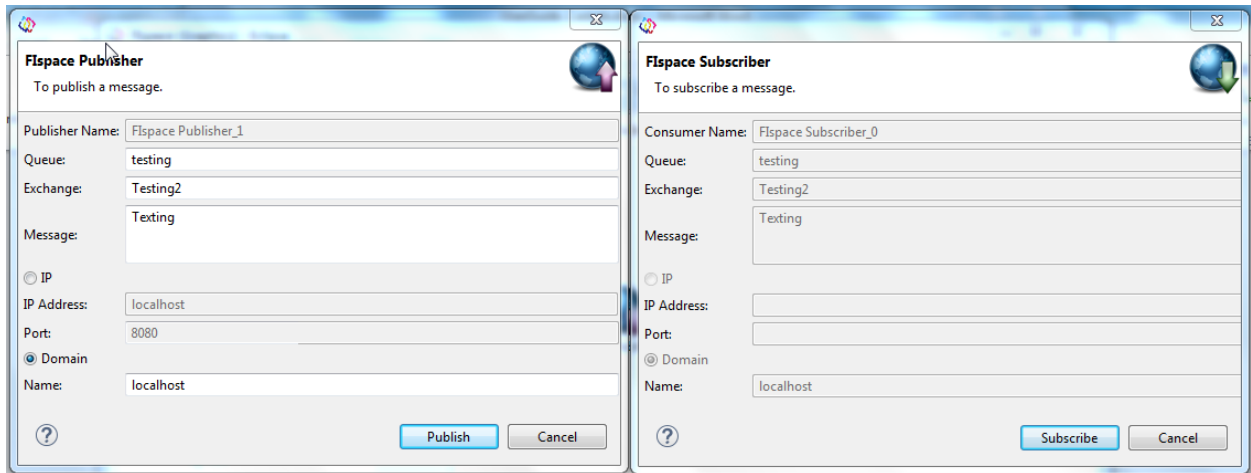


Figure 50: Publisher/Subscriber Form

Through Setting the Data Manually:

The user should insert the Publisher data the Subscriber data manually:

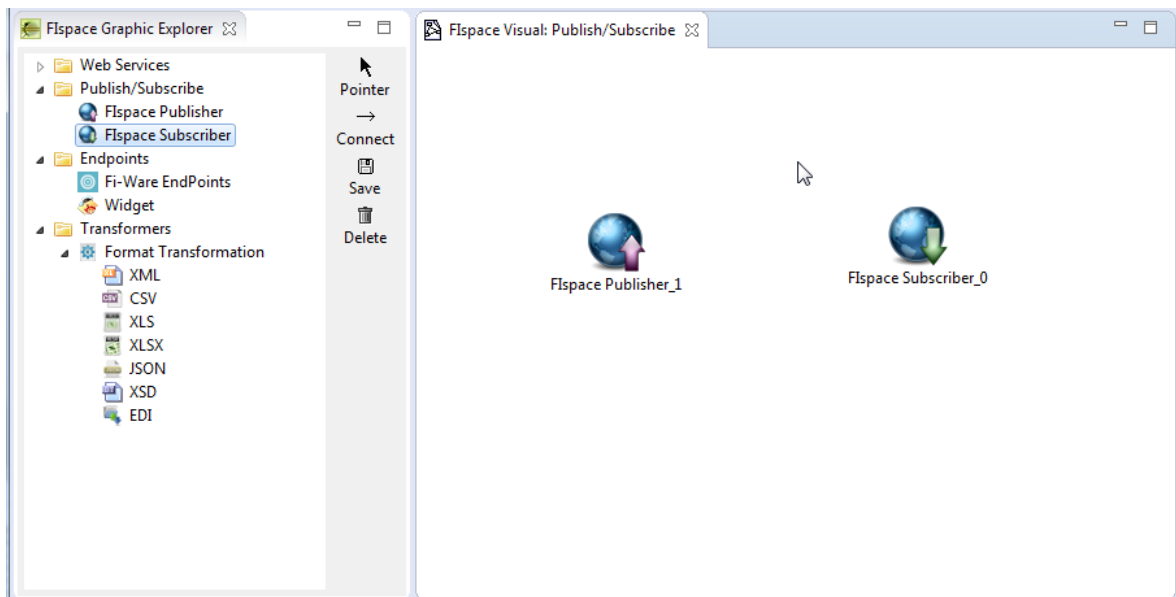


Figure 51: Publisher and Subscriber not connected

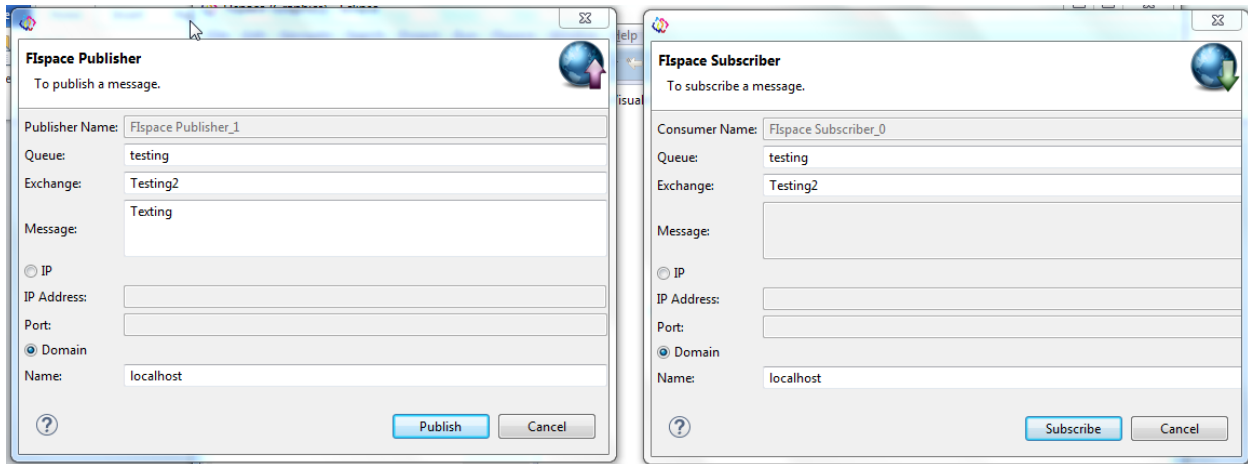


Figure 52: Fulfilling the Publisher/Subscriber settings manually

Clicking the button *Subscribe* the user is going to receive the message:

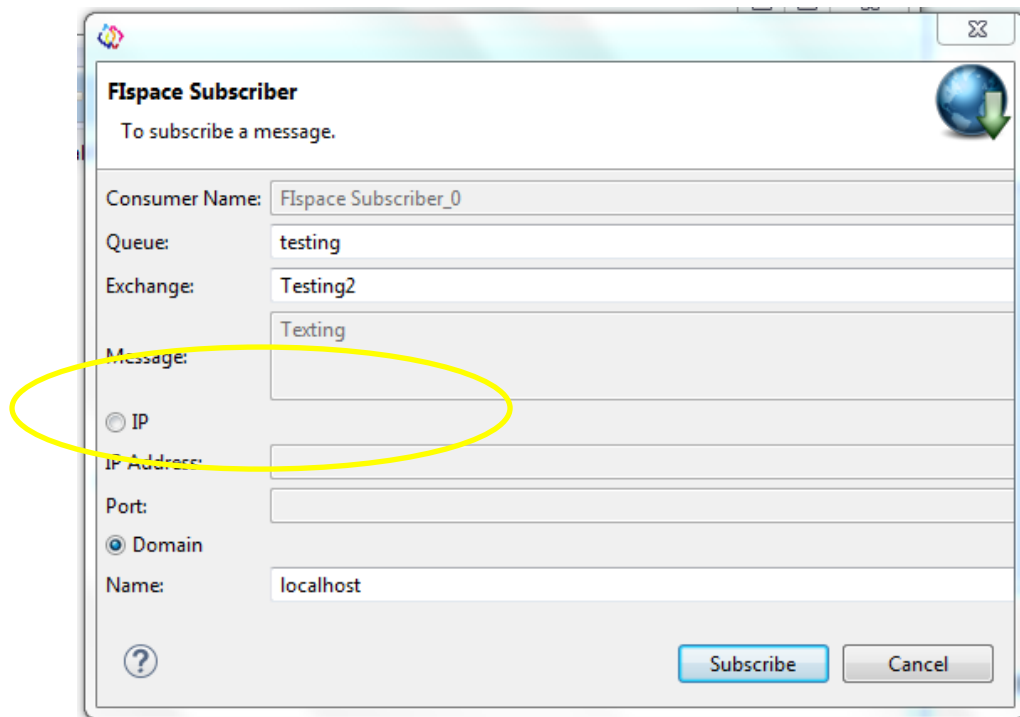


Figure 53: Click on Subscribe button

2.8.1.3 Endpoints

The Flspace Endpoints provide a reference or specification that is used to define a group or family of message addressing properties and give end-to-end message characteristics.



Figure 54: Endpoints elements

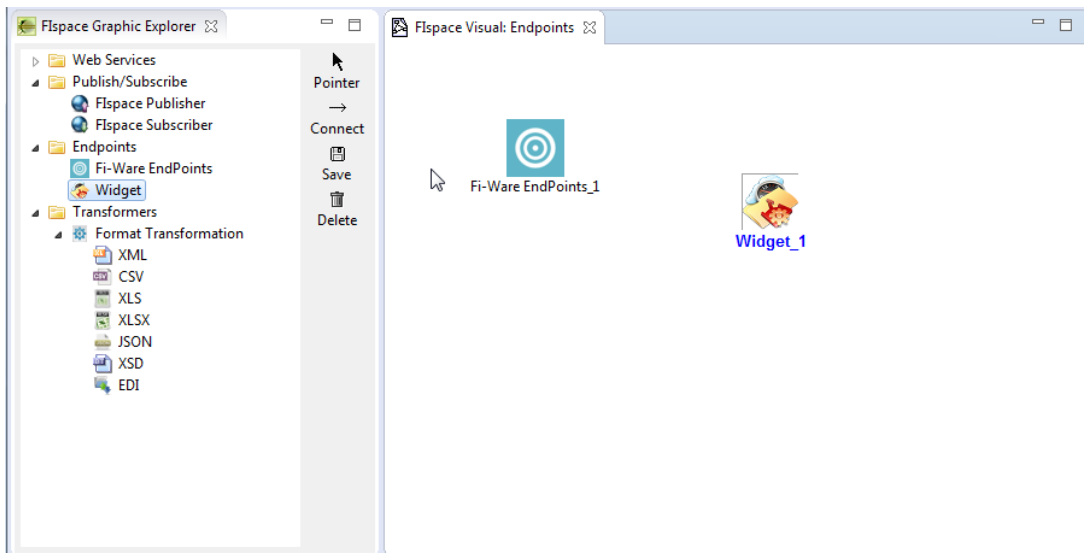


Figure 55: Endpoints on Flspace Visual

2.8.1.3.1 Types of Endpoints

Flspace supports two types of Endpoints FI-Ware and Widget.

Fi-Ware Endpoints

Through this element the user could reach and work with FI-Ware Endpoints.

Widgets

Through this element the user could reach and work with Widgets. The Flspace Widget element Form has two TabFolders: Widget Settings and Graphic Settings.

The user could specify Widget settings:

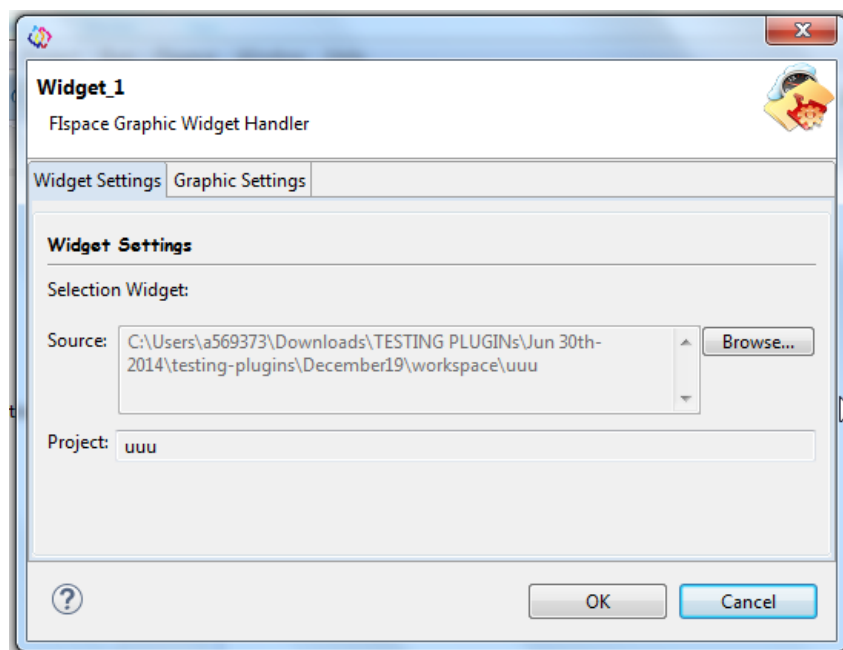


Figure 56: Widget Form – Tab Widget Settings

The user could specify Graphic settings:

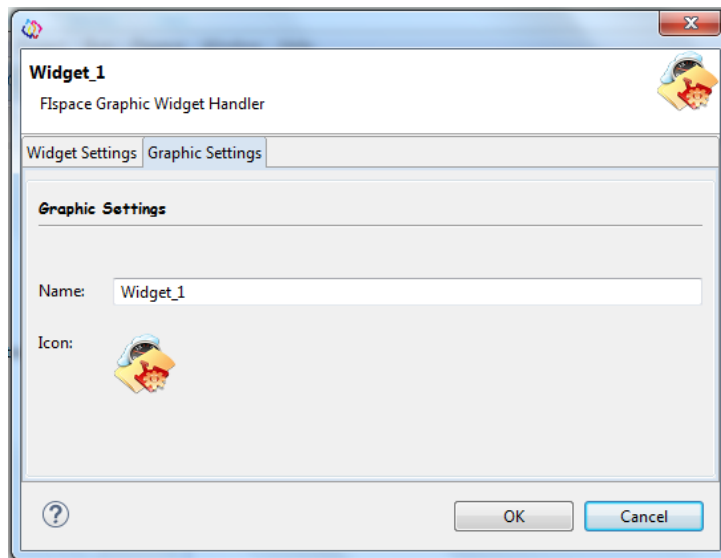


Figure 57: Widget Form – Tab Graphic Settings

2.8.1.4 Transformers

Format Transformation helps the user to convert from several formats into others.:

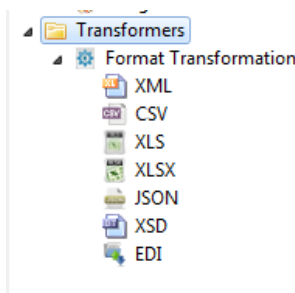


Figure 58: Transformer formats

The following figure shows the types of formats/elements that Flspace Graphic supports.

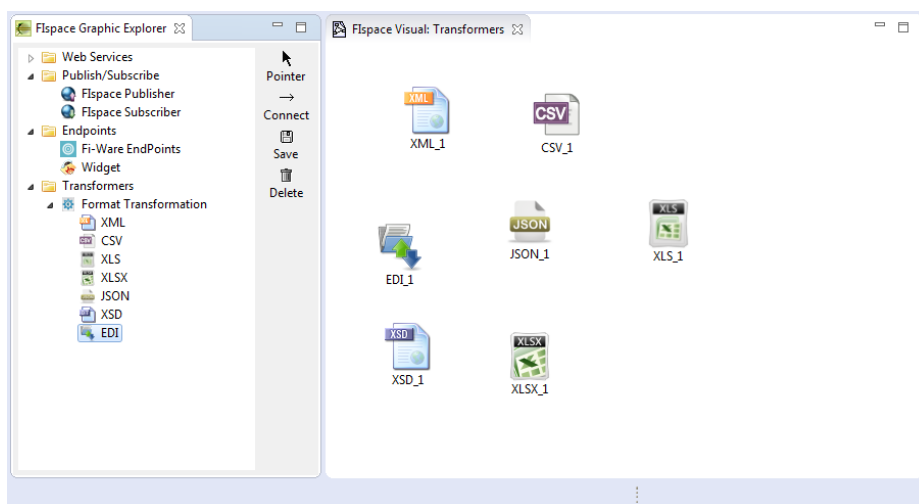


Figure 59: Type of formats on Flspace Visual

2.8.1.4.1 Type of formats

Flspace supports conversion for XML, CSV, EDI, JSON, XLS, XSD, and XLSX.

2.8.1.4.2 Form Details

After double click over an element on the Flspace Visual a form appears:

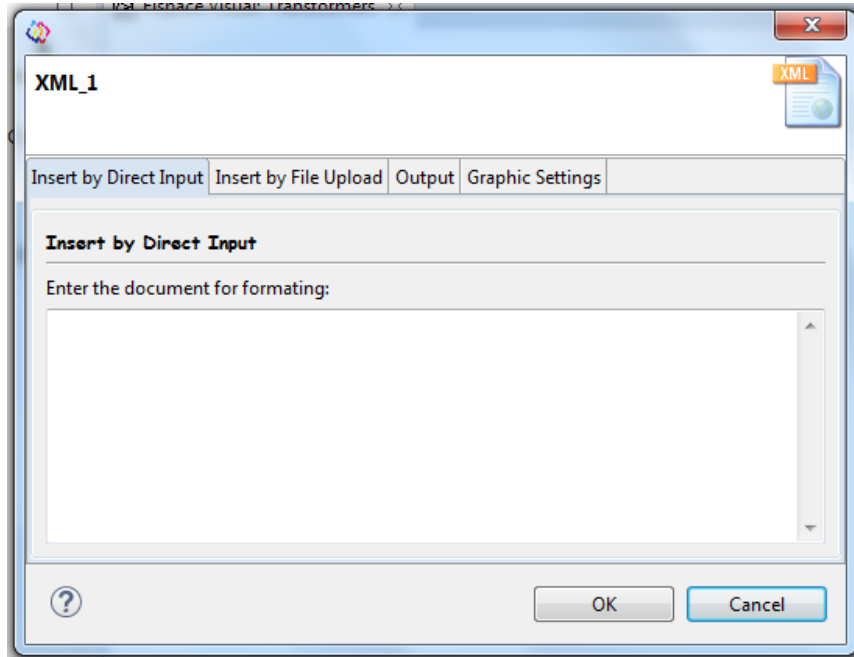


Figure 60: Format/Element XML Form

All Elements of the Flspace Transform Form got the same TabFolders.

Insert by Direct Input

Basically here the user Cut and Paste the data to transform, see Figure 57.

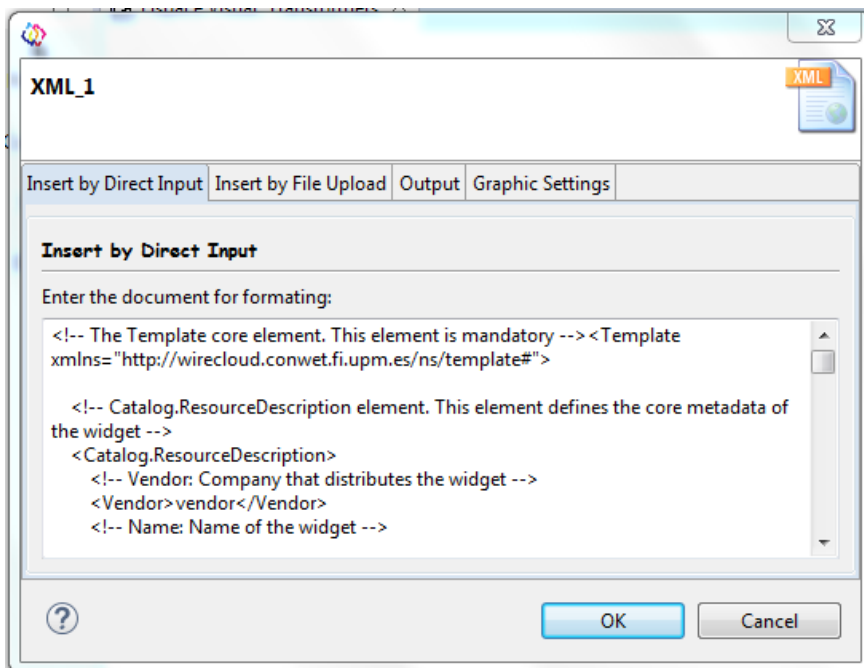


Figure 61: XML Format – Insert by Direct Input

Insert by File Upload

Basically here the user adds the file to transform:

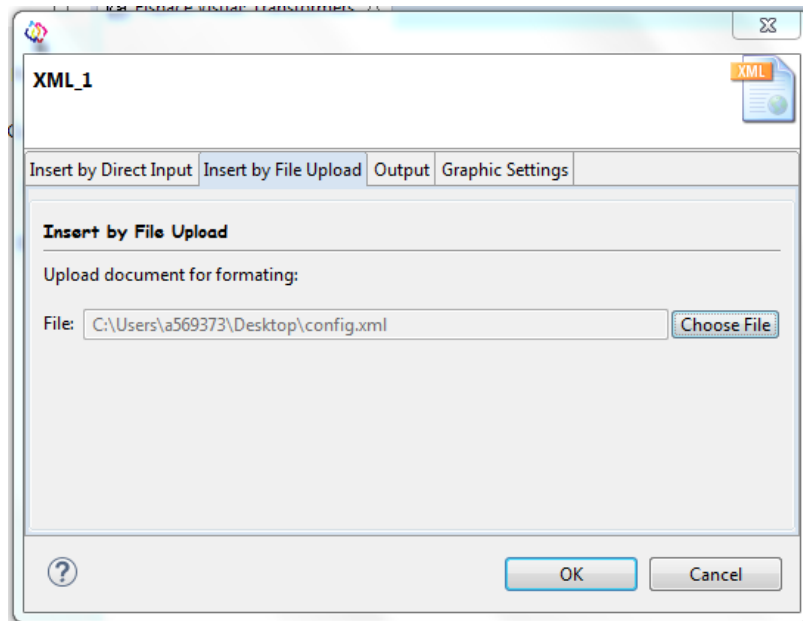


Figure 62: XML Format – Insert by File Upload

Output

If the element is connected already and it is the target format, the user could see the result of the transformation here.

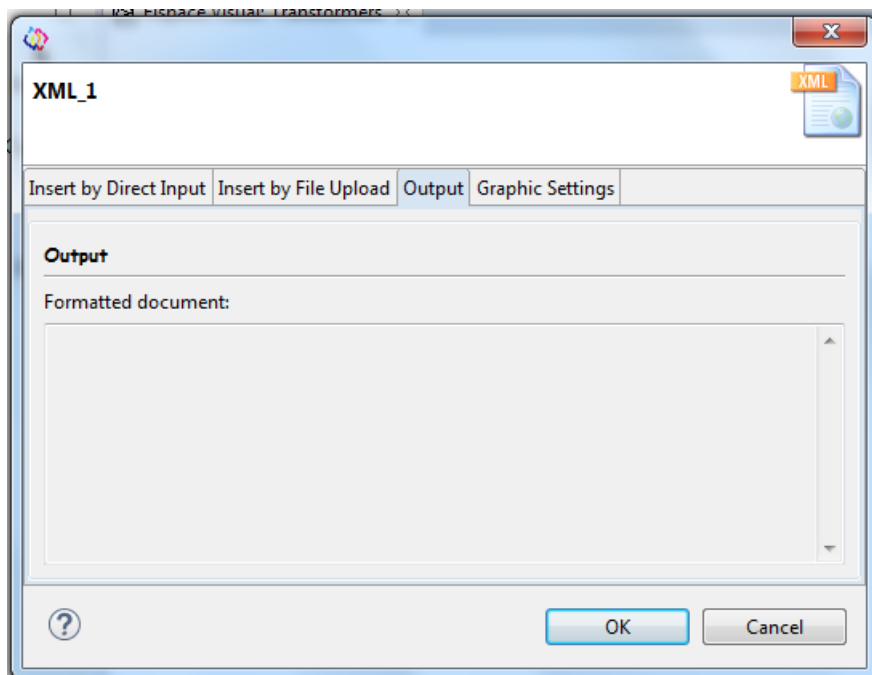


Figure 63: XML Format – Output

Graphic Settings

Through this form the user could change the graphic settings.

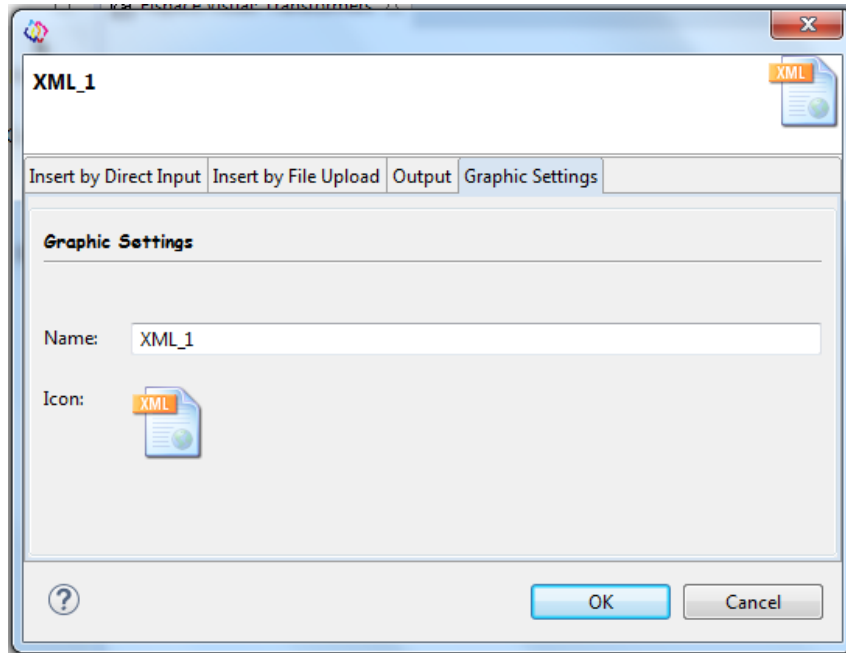


Figure 64: XML Format – Graphic Settings

2.8.1.4.3 How to Transform Data

The next steps are the basic steps to transform any format:

Drag & Drop

You have to drag & drop the element to transform from the browser into the Visual FIspace also the target element. In this case the source is XML and target JSON. See next figure:

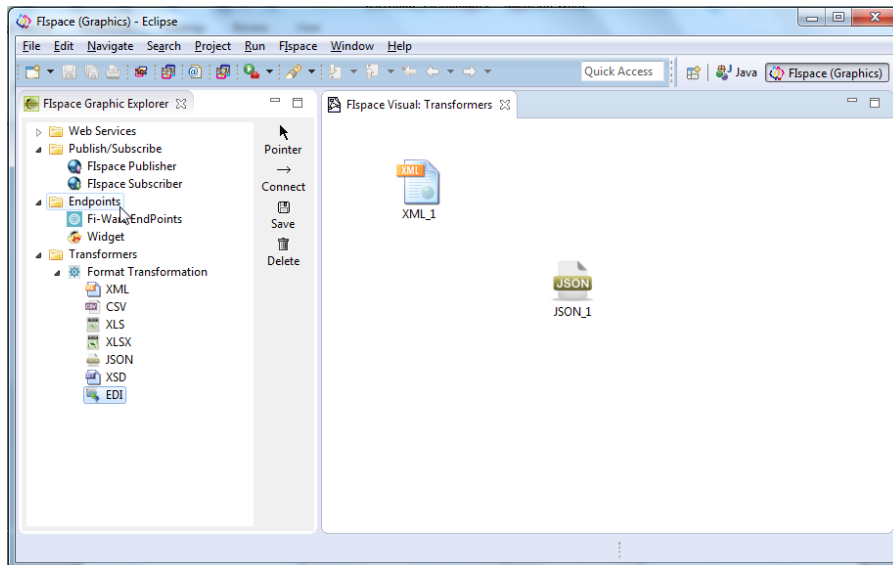


Figure 65: XML Format (Source) and JSON Format (Target)

Add the Data to Transform

The user must add the data. Exist two ways how to add the data to transform: Insert by Direct Input or Insert by File Upload. In this case The Insert by Direct input was used.

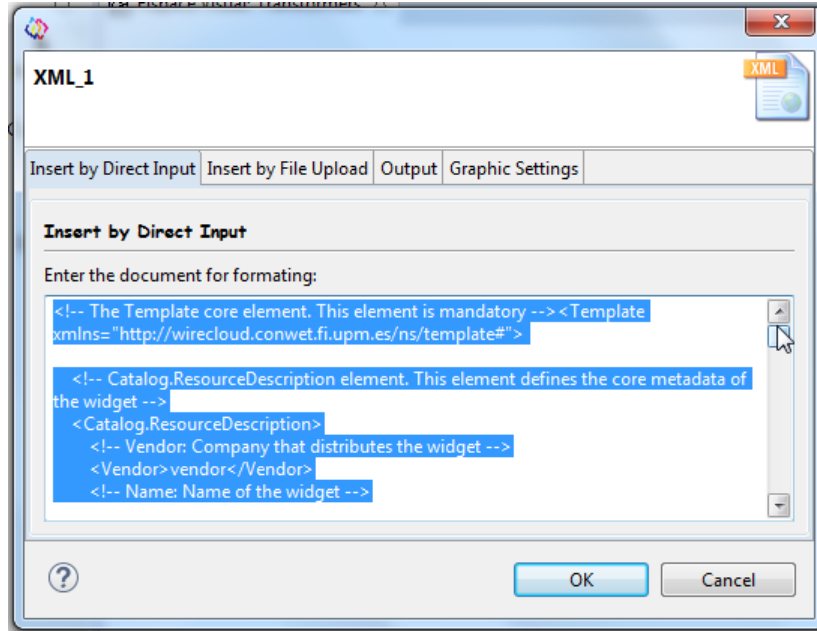


Figure 66: Direct Input

Connect the Elements

Selecting the Source (XML) and Target (JSON) by the option “Connect” from the toolbar, then the data in the Target is going to be transformed automatically, see Figure 63.

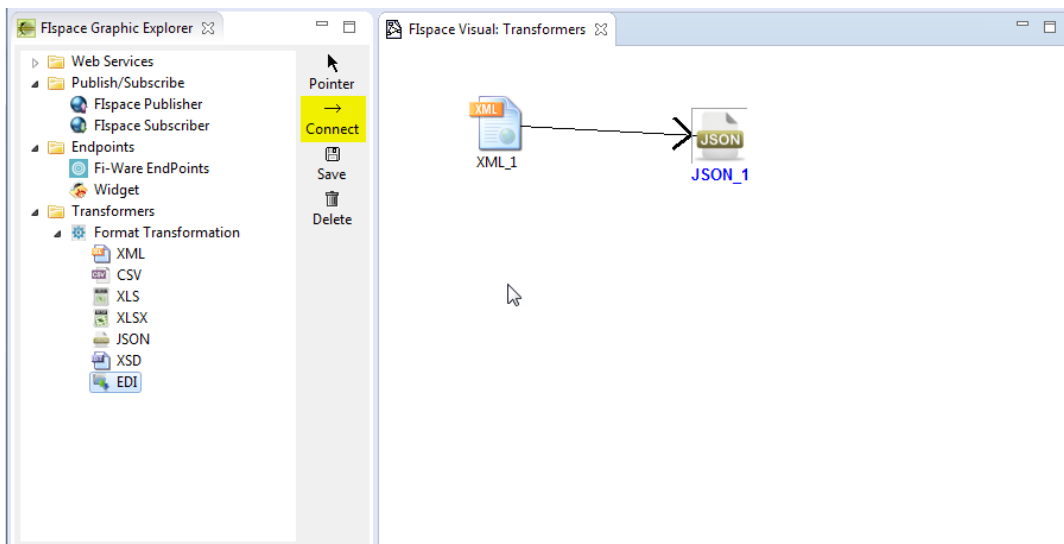


Figure 67: Connect the elements

Target Tab Output

After connecting the user should check the Tab Output from the target:

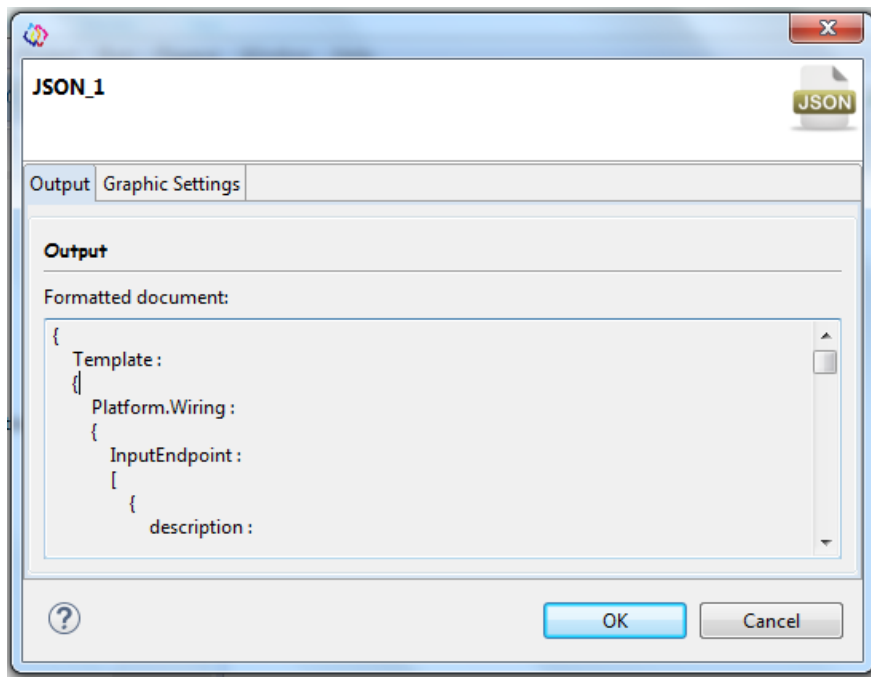


Figure 68: Target Output

The transformation from XML to JSON was successfully done by the Graphic Plugin.

2.8.2 Flspace Graphics Toolbar

The Flspace Graphic Toolbar allows the user to work and interact with the Flspace Graphic Elements.

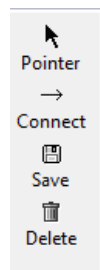


Figure 69: Flspace Graphic Toolbar

The Flspace Graphic toolbar elements are:

Button *Pointer*

Through this button the user could deselect the current element selected on the Visual Graphic Area, basically this is the on-focus.

Button *Connect*

Through this button the user could connect different elements on the Visual Graphic Area.

Button *Save*

Store all the elements on in the Visual Graphic Area and its relations in XML Format.

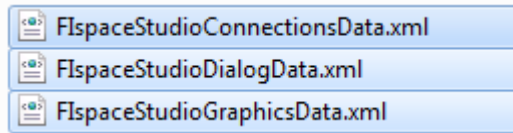


Figure 70: FIspace Graphics Persistent Files

The description of the storage files is following:

FIspaceStudioConnectionsData.xml File:

It is produced by flushing the relations between elements on the Visual Graphic Area to disk.

FIspaceStudioDialogData.xml File:

It is produced by flushing the Dialog data of the elements on the Visual Graphic Area to disk.

FIspaceStudioGraphicsData.xml

It is produced by flushing the Graphical data of the elements on the Visual Graphic Area to disk.

Button *Delete*

Through this button the user could delete different elements on the Visual Graphic Area:

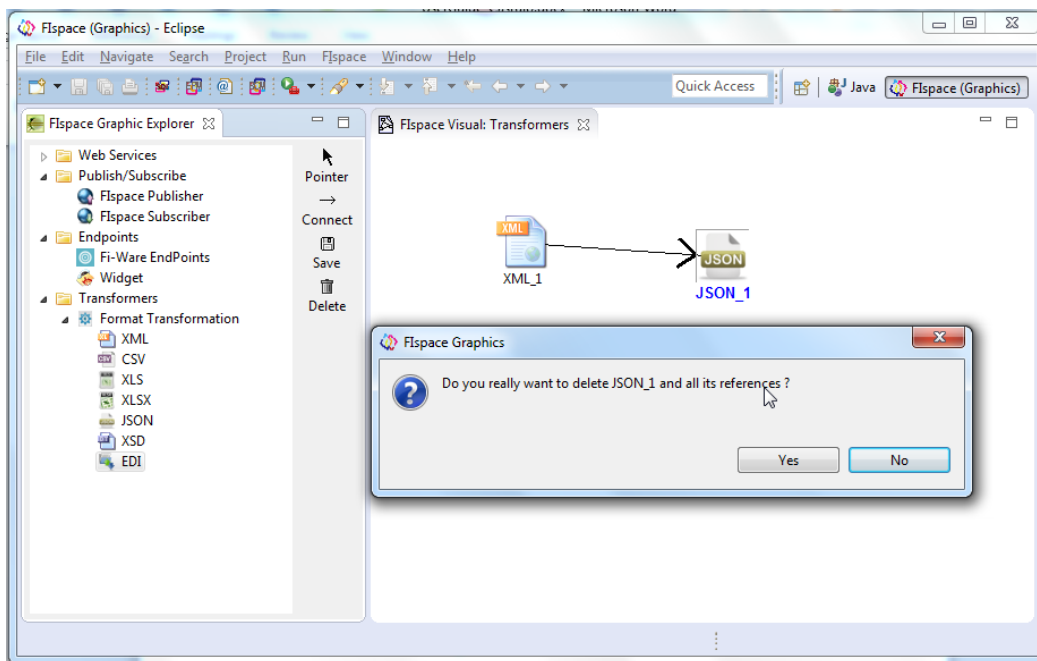


Figure 71: Deleting Elements

2.9 Flspace-import Plugin

You can import Flspace projects in your workspace by means of this plugin extension. Select from the main menu *File* → *Import* → *Flspace* → *Existing Flspace Project*.

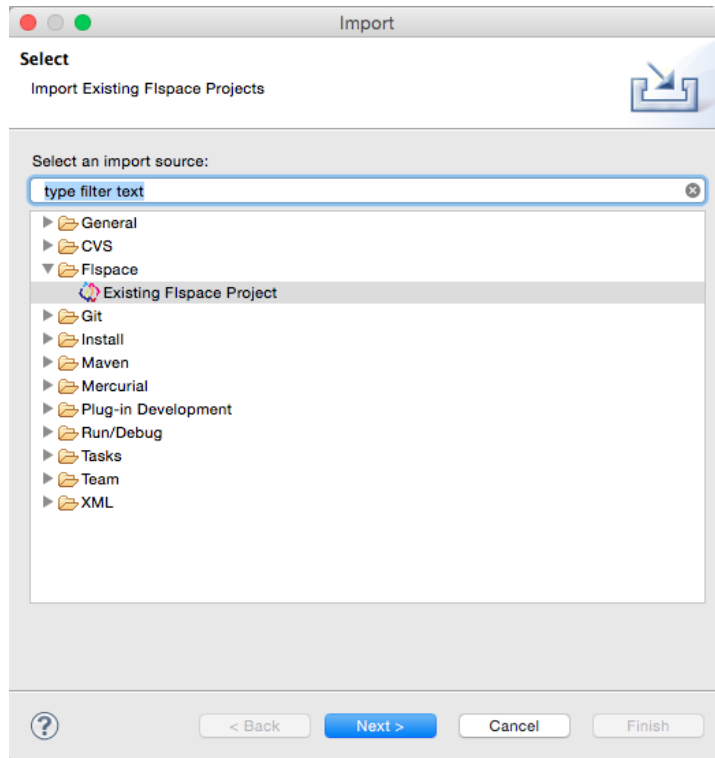


Figure 72: Flspace import plugin

It will then appear a wizard for browsing and selecting the path of the Flspace project to be imported on workspace. Once you have chosen the path you can safely import the selected project on workspace by clicking on *Finish* button.

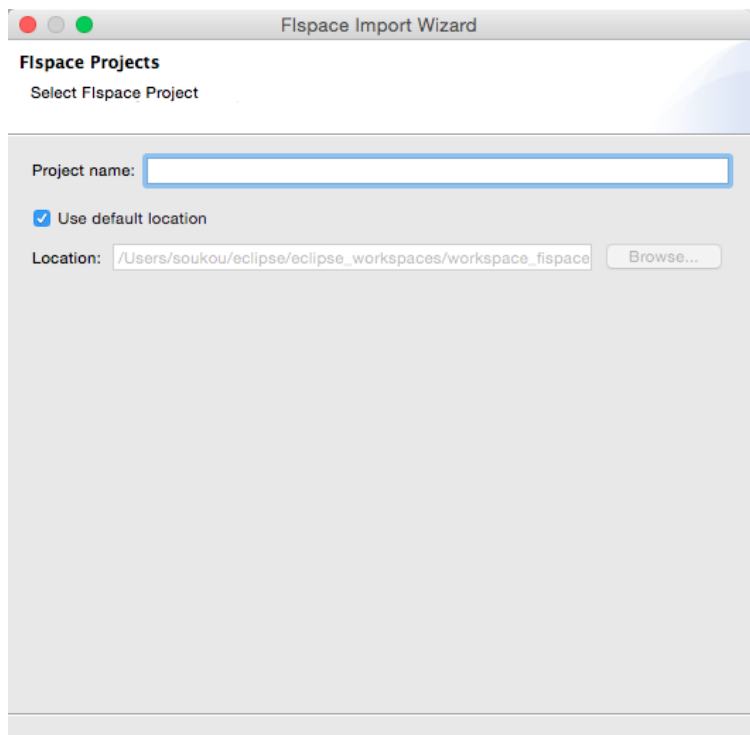


Figure 73: Flspace import wizard

2.10 Flspace-perspective Plugin

The Flspace perspectives can be opened by clicking on the menu bar: Window -> Perspective -> Other -> Flspace.

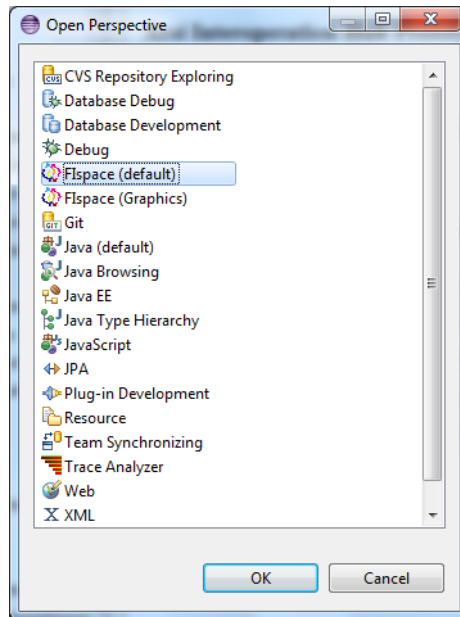


Figure 74: Flspace SDK Perspectives View entry

If the user selects the Flspace *Graphics* one, the layout changes to the following:

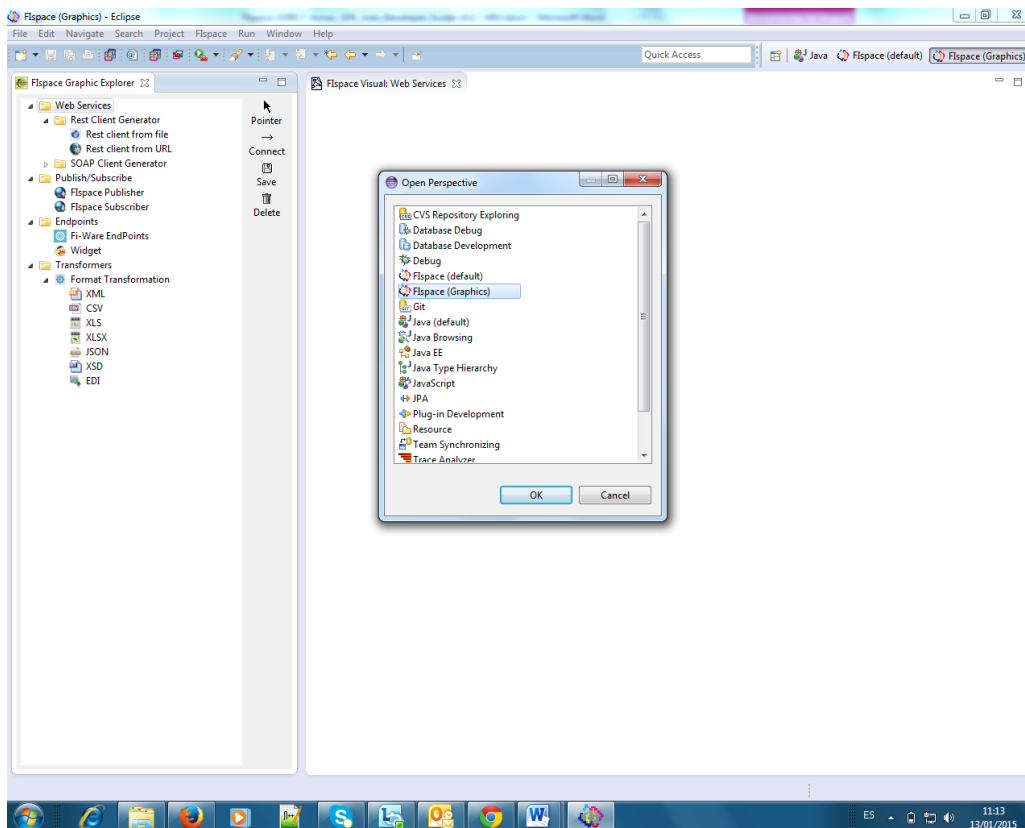


Figure 75: Flspace Graphics perspective

2.11 Flspace-preferences Plugin

Flspace has a number of preference settings under *Window* → *Preferences* → *Flspace Preferences*.

The Flspace recommended settings are:

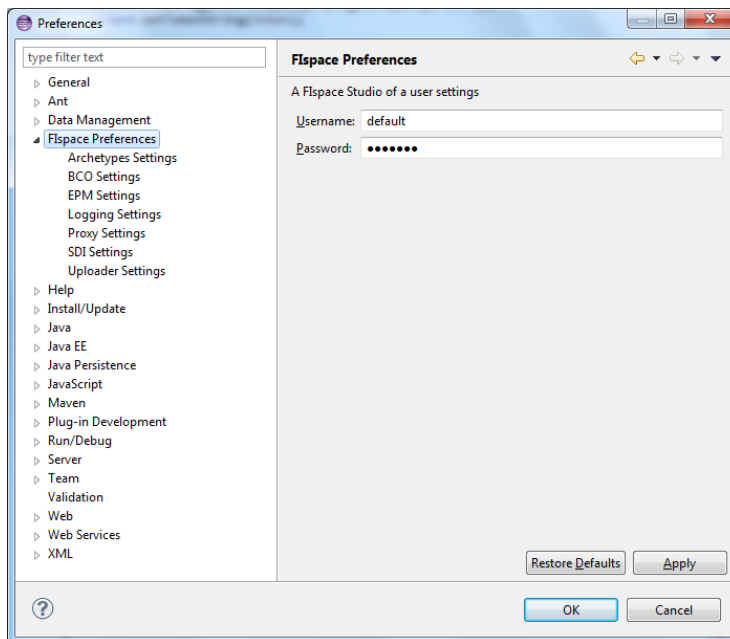


Figure 76: Flspace Preferences Settings Form

Archetypes Settings:

In short, *archetype* is a Maven project templating toolkit. The simplest way to get started with Flspace is to generate a new Flspace project using the Flspace Wizard.

In the “Archetype Settings” form the user can provide the configuration of the each Flspace Project.

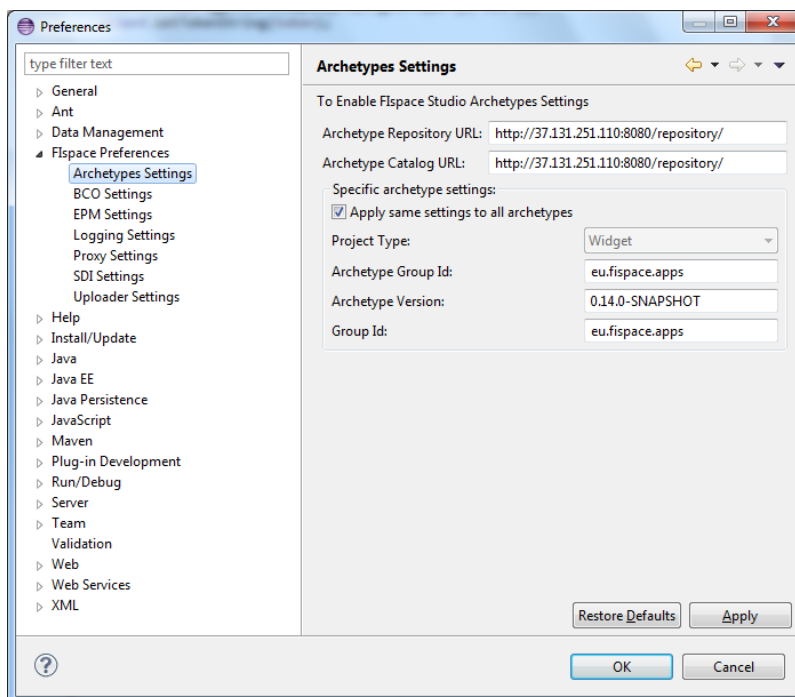


Figure 77: Archetypes Settings Form

BCO Settings:

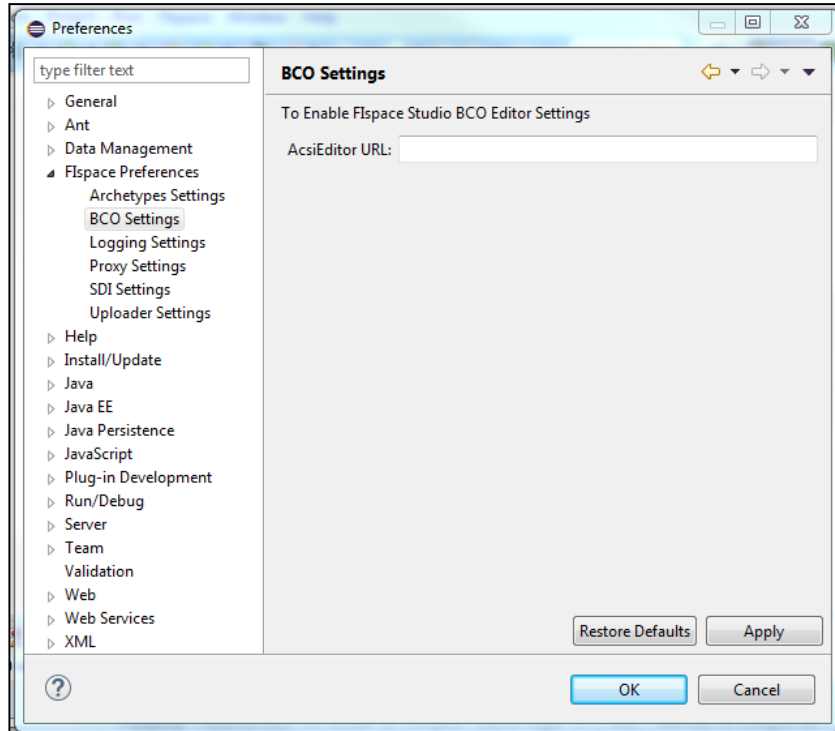


Figure 78: BCO Settings Form

Logging Settings:

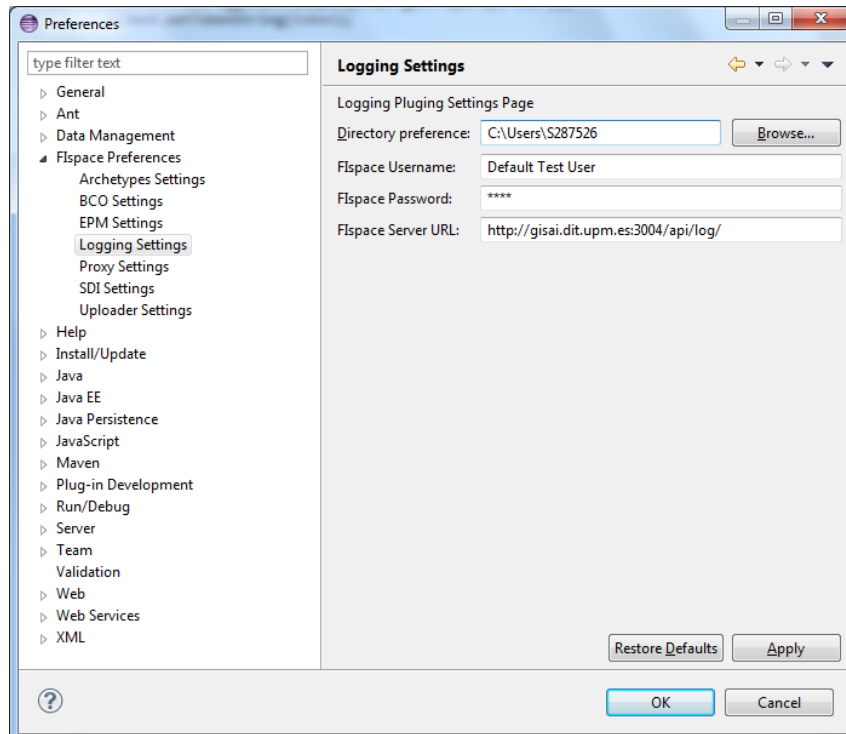


Figure 79: Logging Settings Form

Proxy Settings:

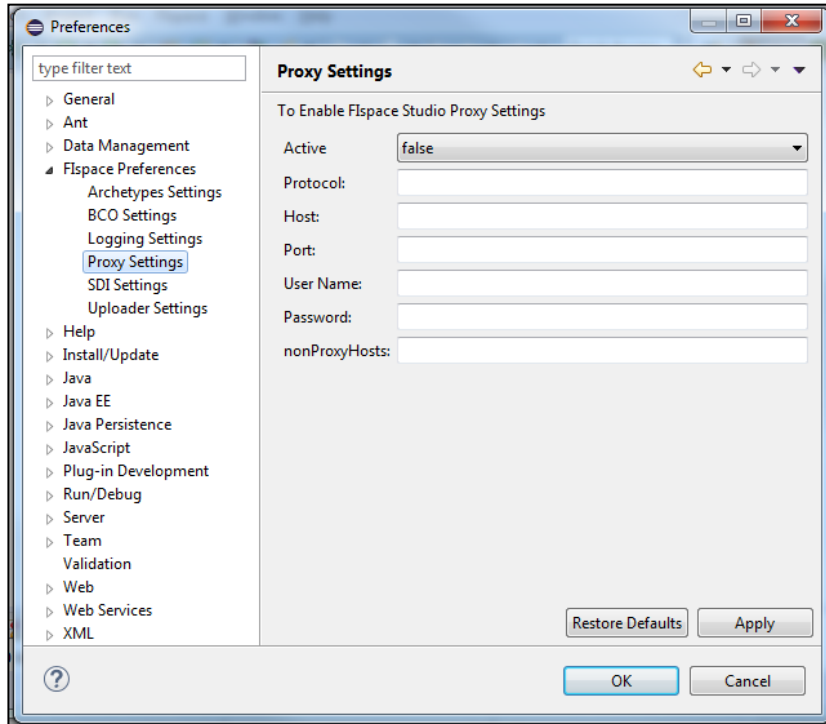


Figure 80: Proxy Settings Form

SDI Settings:

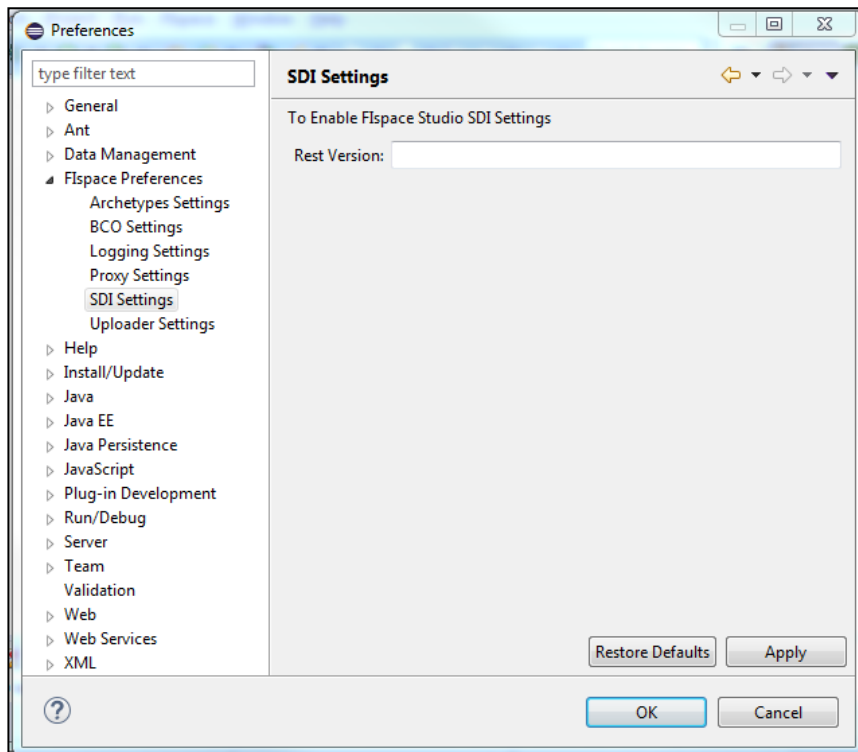


Figure 81: SDI Settings Form

Uploader Settings:

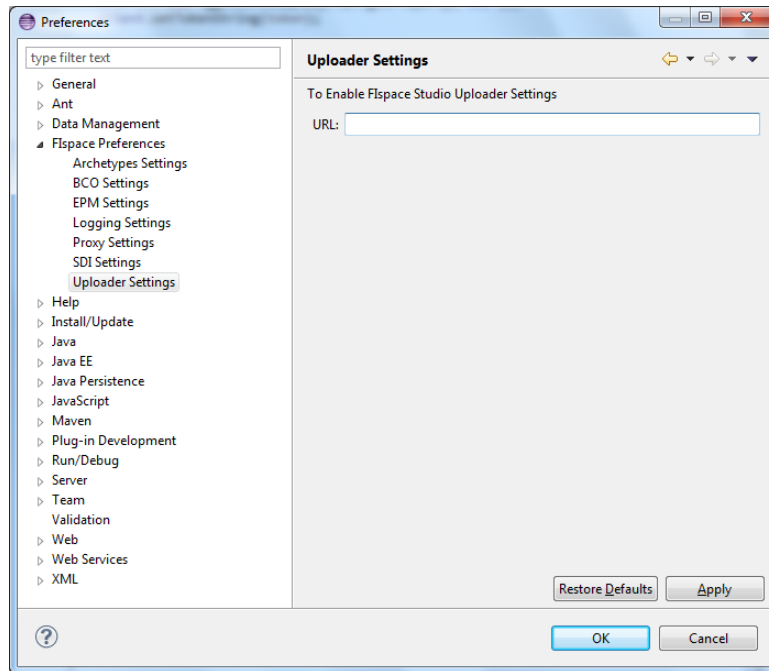


Figure 82: Uploader Settings Form

2.12 Flspace-pubsub Plugin

2.12.1 Prerequisites

Messages are published and subscribed using RabbitMQ Server [22]. We can work with a remote server specifying an IP address and port or using a domain name, or using a local server. In case of work with a local server is necessary to download and install RabbitMQ from: <http://www.rabbitmq.com/download.html>

2.12.2 Using the plugins

From the Flspace menu bar, select the option **Publish/Subscribe**. This option is subdivided in two options, **Publish Message** or **Subscribe Message**.

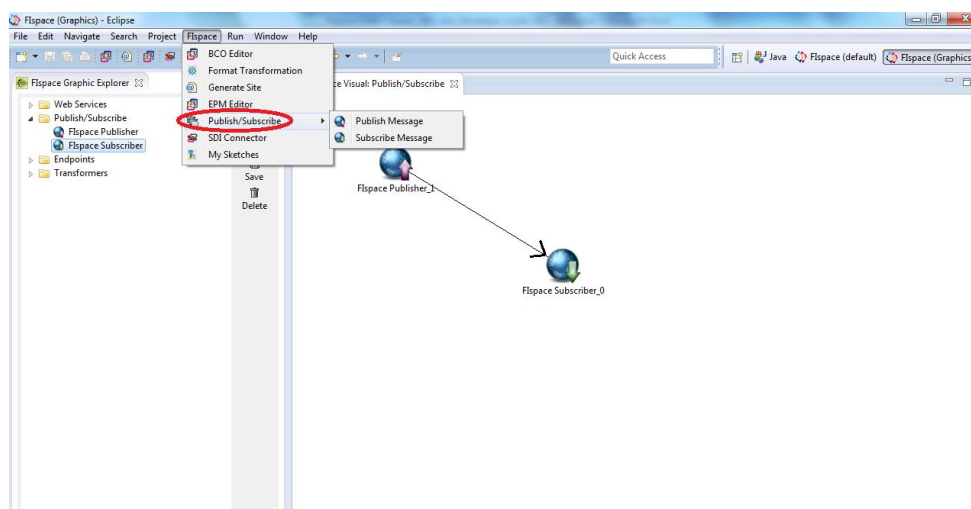


Figure 83: Flspace publisher-subscriber

Publish and Subscribe are separate plugins. They are working independently each by itself.

2.12.3 Publish a message

In order to publish a message, we need to specify the following parameters: a queue name, an exchange name, the message and an IP with a port or a domain name. See the next figure:

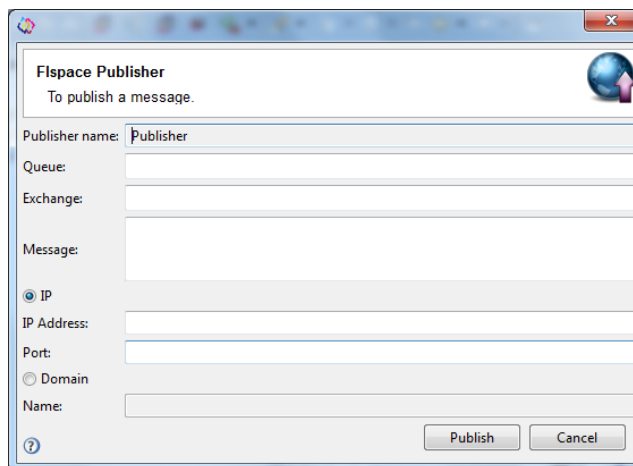


Figure 84: Publisher dialog

2.12.4 Subscribe a message

In order to subscribe messages, we need also to specify some information, a queue name, an exchange name, the message and an IP with a port or a domain name such is showed in the next image. Every time a message is sent to the RabbitMQ, the consumer or subscriber captures it and shows the message just received.

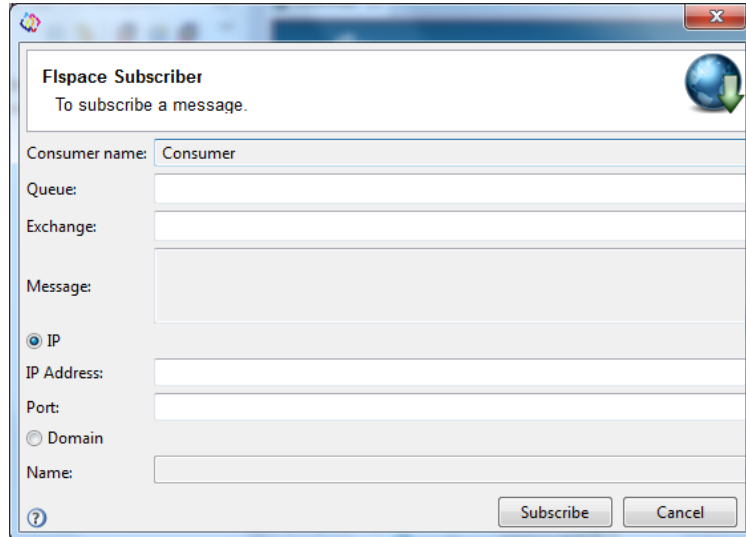


Figure 85: Subscriber dialog

2.13 Flspace-REST Plugin

From the Bar menu select **File -> New Project -> Rest Client Generator**. This wizard will be opened:

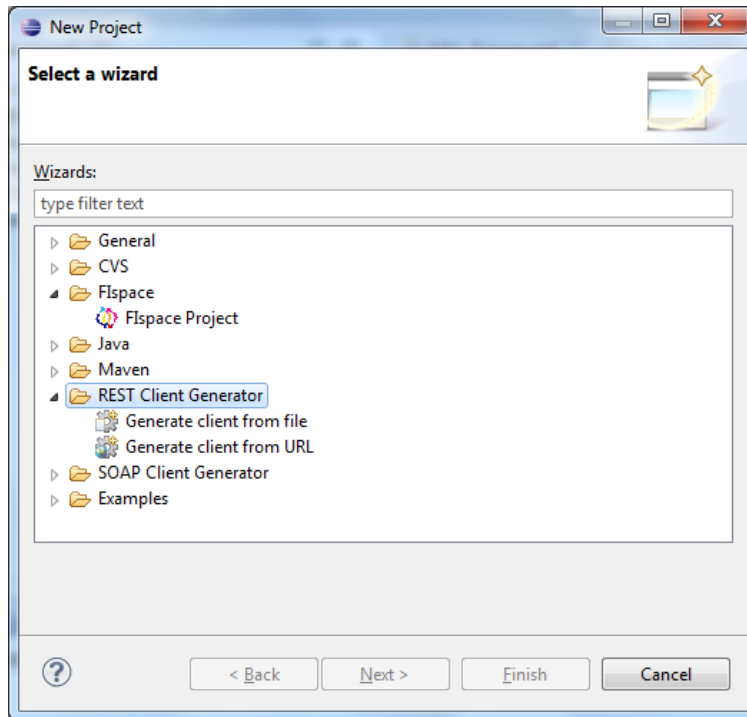


Figure 86: Flspace REST tool

If you want to generate a REST client from a file, choose the corresponding item.

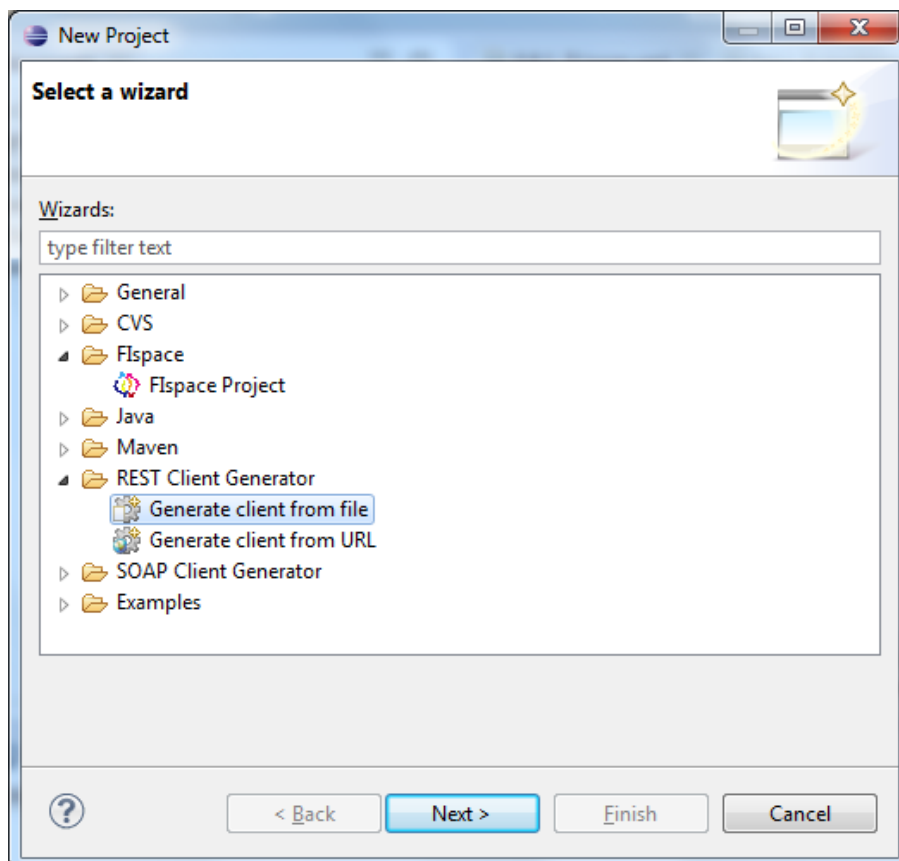


Figure 87: Flspace REST tool from file (I)

Clicking on Next button the following window is shown (Figure 88).

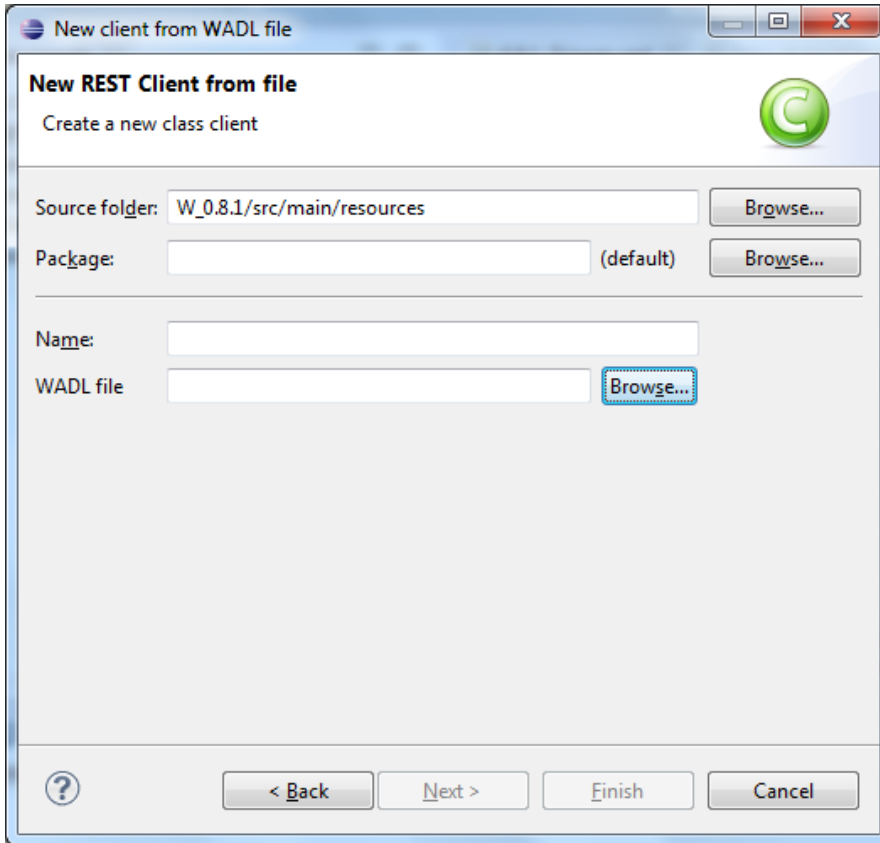


Figure 88: Flspace REST tool from file (II)

If you want to generate a REST client from an URL, choose the corresponding item.

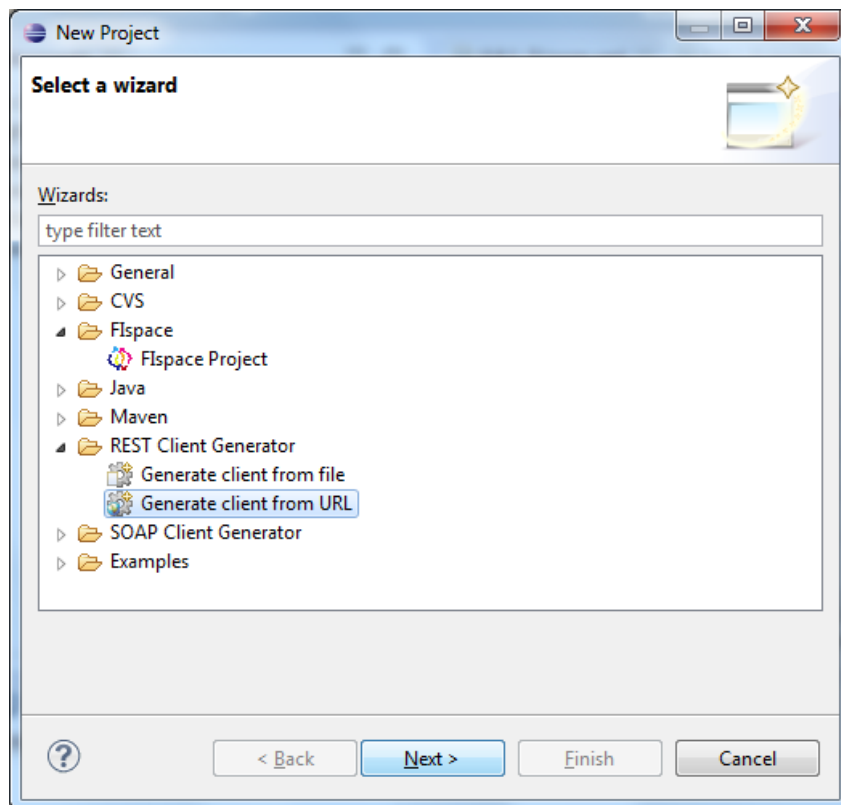


Figure 89: Flspace REST tool from URL (I)

Click on Next button to continue with the wizard.

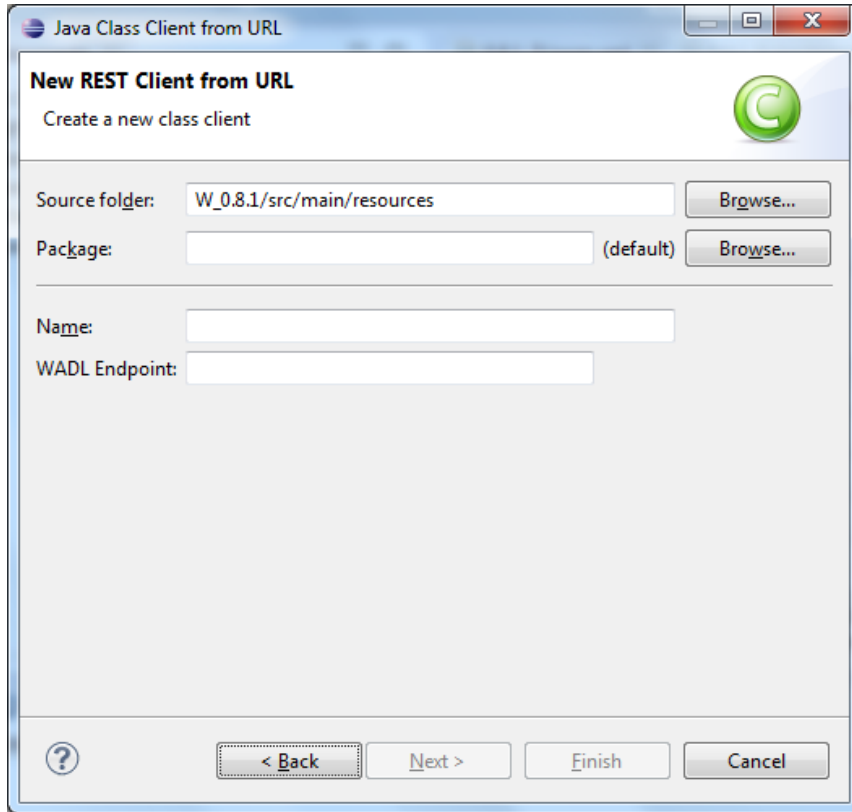


Figure 90: FIspace REST tool from URL (II)

2.14 Flspace-sdi-connector Plugin

Once the SDI Connector plugin is installed into the workbench, new elements are added to the workbench. **System and Data Integration** layer is a component of Flspace architecture.

A new icon (SDI Connector) will be enabled to launch the SDI Connector plugin. This icon can be showed in the following picture.



Figure 91: Button location to launch the SDI Connector plugin

Also in Flspace menu entry:

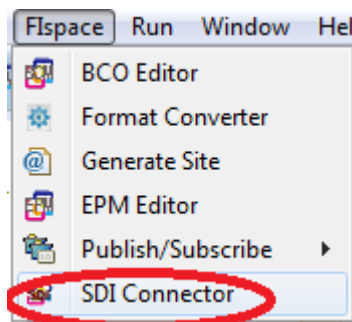


Figure 92: Menu item to launch the SDI Connector

A new Eclipse view associated to the SDI Connector plugin will be accessible. To open this new view provided by the plugin, the next steps must be followed:

Into the main menu select Window -> Show View -> Other; once opened, the Flspace view will be available. This view contains a tool called SDI Rest Tool, which will provide access to Rest Services for the SDI Connector.

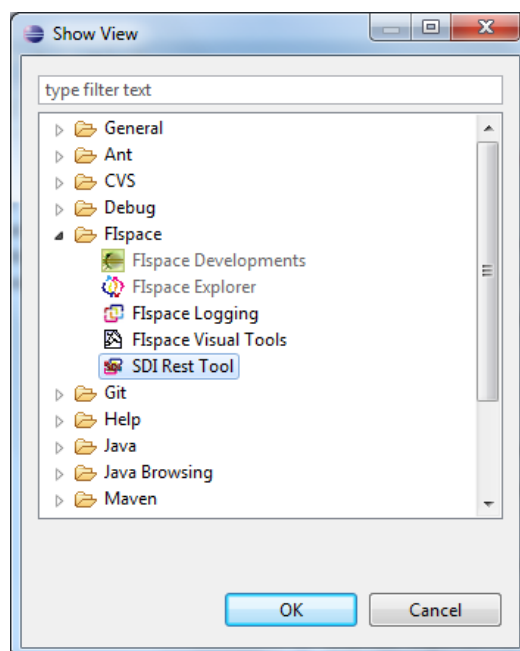


Figure 93: Flspace SDI Rest Tool (I)

Selected the view, a new item appears into the workbench. This contains a console view to check the status of the SDI plugin and the SDI Rest Tool to manage REST calls.

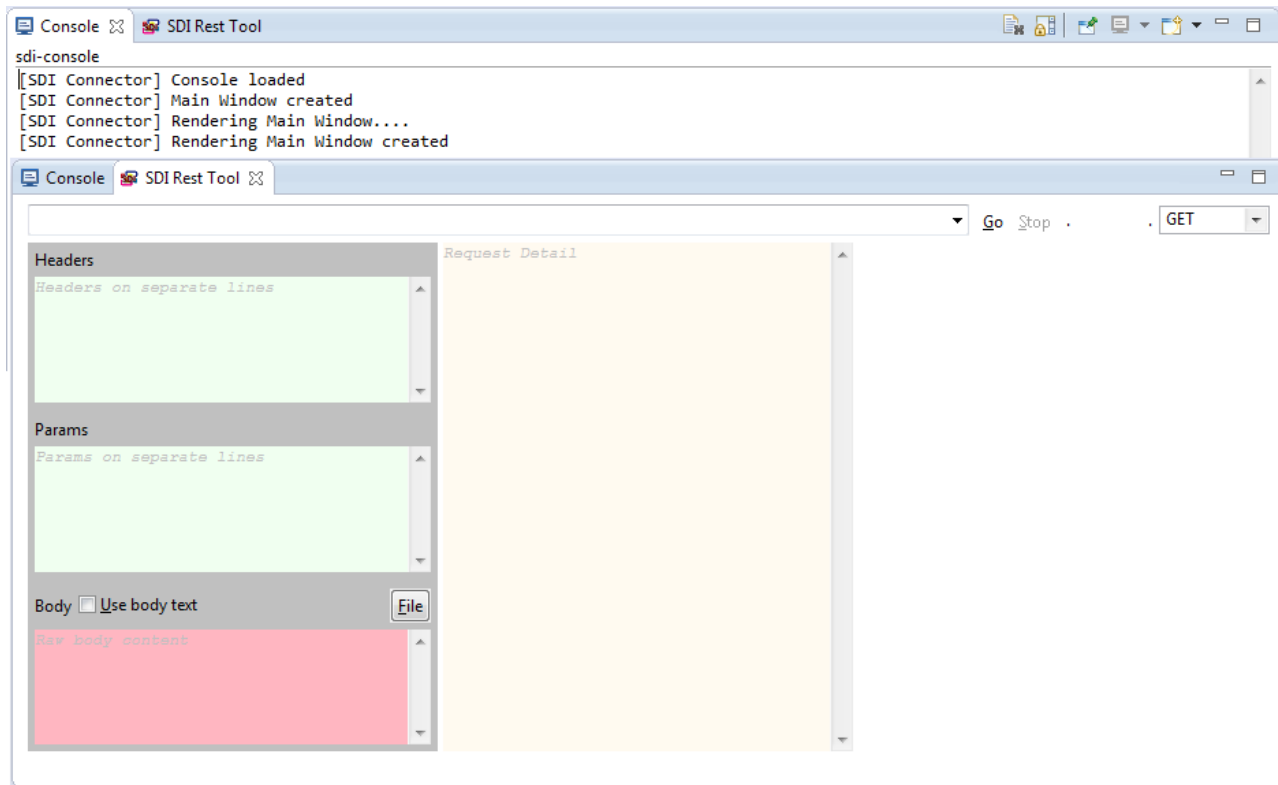



Figure 94: Flspace SDI Rest Tool (II)

The behaviour and use of this tool will be explained in the paragraph 2.14.2 “*The SDI Rest Tool*”.

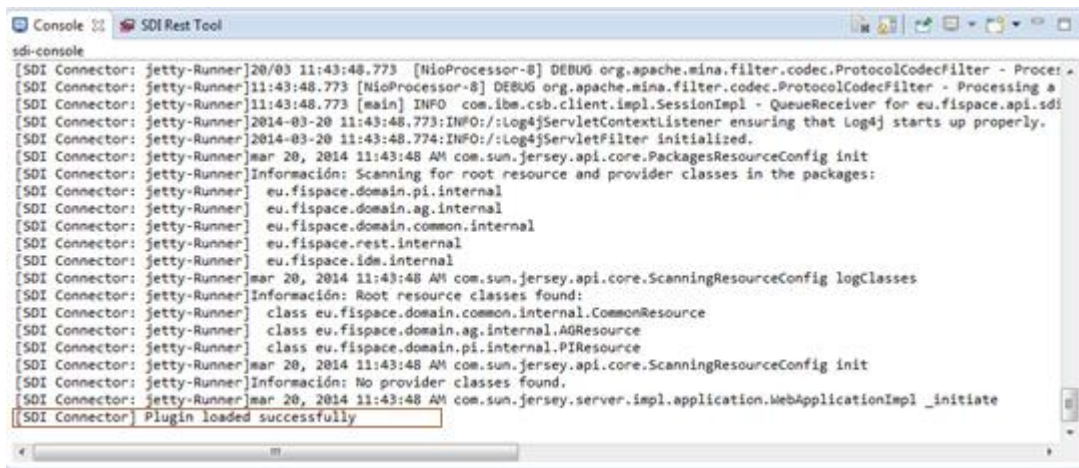
2.14.1 Using the SDI Connector

Once explained the items provided by the plugin described in the previous chapter, we can launch the SDI Connector by clicking the button defined for that purpose .

After that, a new window, will inform us about the progress of that launching.

The SDI connector plugin will perform all the activities required for download, install and setup the plugin. During this process it is possible to check its progress by glancing the console. If the whole process ends without errors, two events have occurred:

- The console’s output will show all the information related to the SDI installation. It means, therefore any errors occurred as the successful installation. It is shown in the next figure:



```

sdi-console
[SDI Connector: Jetty-Runner] 20/03 11:43:48.773 [NioProcessor-8] DEBUG org.apache.mina.filter.codec.ProtocolCodecFilter - Process
[SDI Connector: Jetty-Runner] 11:43:48.773 [NioProcessor-8] DEBUG org.apache.mina.filter.codec.ProtocolCodecFilter - Processing a
[SDI Connector: Jetty-Runner] 11:43:48.773 [main] INFO com.ibm.csb.client.impl.SessionImpl - QueueReceiver for eu.fispace.api.sdi
[SDI Connector: Jetty-Runner] 2014-03-20 11:43:48.773:INFO:/:/Log4jServletContextListener ensuring that Log4j starts up properly.
[SDI Connector: Jetty-Runner] 2014-03-20 11:43:48.774:INFO:/:/Log4jServletFilter initialized.
[SDI Connector: Jetty-Runner] mar 20, 2014 11:43:48 AM com.sun.jersey.api.core.PackagesResourceConfig init
[SDI Connector: Jetty-Runner] Información: Scanning for root resource and provider classes in the packages:
[SDI Connector: Jetty-Runner] eu.fispace.domain.pi.internal
[SDI Connector: Jetty-Runner] eu.fispace.domain.ag.internal
[SDI Connector: Jetty-Runner] eu.fispace.domain.common.internal
[SDI Connector: Jetty-Runner] eu.fispace.rest.internal
[SDI Connector: Jetty-Runner] eu.fispace.idm.internal
[SDI Connector: Jetty-Runner] mar 20, 2014 11:43:48 AM com.sun.jersey.api.core.ScanningResourceConfig logClasses
[SDI Connector: Jetty-Runner] Información: Root resource classes found:
[SDI Connector: Jetty-Runner] class eu.fispace.domain.common.internal.CommonResource
[SDI Connector: Jetty-Runner] class eu.fispace.domain.ag.internal.AGResource
[SDI Connector: Jetty-Runner] class eu.fispace.domain.pi.internal.PIResource
[SDI Connector: Jetty-Runner] mar 20, 2014 11:43:48 AM com.sun.jersey.api.core.ScanningResourceConfig init
[SDI Connector: Jetty-Runner] Información: No provider classes found.
[SDI Connector: Jetty-Runner] mar 20, 2014 11:43:48 AM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate
[SDI Connector] Plugin loaded successfully

```

Note that in this case, the plugin has been loaded successfully and no previous errors have been produced.

- Moreover, a new window with the entire rest specification (the WADL file) will be shown to the user in aim to provide him the whole REST specification associated to the SDI Connector. The picture below shows a snippet of the content of this file.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <application xmlns="http://wadi.dev.java.net/2009/02">
  <doc xmlns:jersey="http://jersey.java.net/" jersey:generatedBy="Jersey: 1.17.1 02/28/2013 12:47 PM" />
  <grammars />
- <resources base="http://localhost:9090/api/">
- <resource path="/domain/pi">
  - <resource path="PushShipmentMessage">
    - <method id="postPushShipmentMessage" name="POST">
      - <request>
        <representation mediaType="application/xml" />
        <representation mediaType="application/json" />
      </request>
      - <response>
        <representation mediaType="text/html" />
      </response>
    </method>
  </resource>
</resources>

```

Note: It's also possible to check the content of this file, accessing with a web browser the URL associated to localhost machine: <http://localhost:9090/api/application.wadl>

2.14.2 The SDI Rest Tool

The SDI connector also provides a REST tool for testing purpose. This tool let users test REST services against different machines or servers. In this case, this tool let us check the SDI services deployed into the local machine.

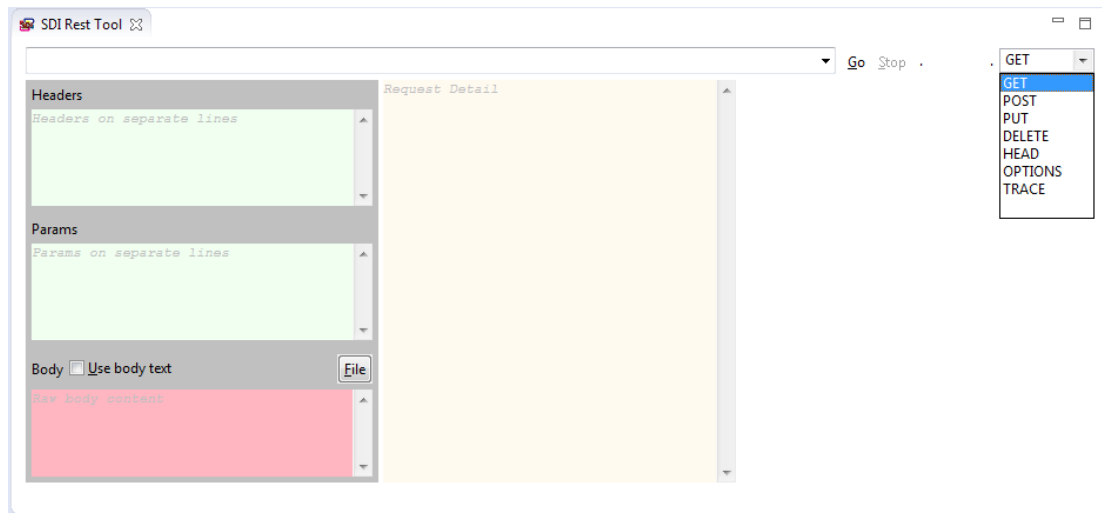


Figure 95: FIspace SDI Rest Tool view

Several items are described as follows:

- **URL:** The REST URL to test. This can be a local one or remote one.
- **Go:** Start the request defined by URL, method, headers and parameters.
- **Stop:** Stop the current request.
- **Type of request:** The request can be done using different kind of methods. Further information about request methods can be found in the Hypertext Transfer Protocol -- HTTP/1.1 at W3C.org.
- **Headers:** Defines (if necessary) the headers for the request. If more than one is defined, each one must be defined in separate lines. Further information about Headers can be found in the Hypertext Transfer Protocol -- HTTP/1.1 at W3C.org.
- **Params:** Defines, as the previous section, the parameters for the formal REST request. In the same way like headers, each parameter should be defined in separate lines. Further information about protocol parameters can be found in the Hypertext Transfer Protocol -- HTTP/1.1 at W3C.org.
- **Body:** HTTP Body Data is the data bytes transmitted in an HTTP transaction message immediately following the headers if there is any. Enabling the “Use Body Text” option Use body text, user can define it.

It's possible to use an existing file to provide the content of the body into the request. In this case, clicking the “File” button , the user can specify the file, and the type of from the local machine.

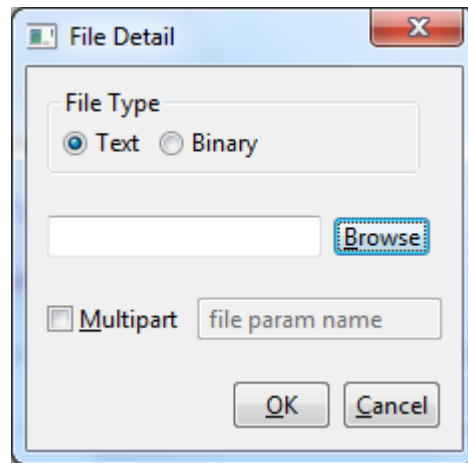


Figure 96: Flspace SDI Rest Tool file dialog

Further information about message types can be found in the Hypertext Transfer Protocol -- HTTP/1.1 at W3C.org.

Request details panel: Shows information about the kind of request done by the user.

```
GET / HTTP/1.1
Host: www.google.es
Connection: Keep-Alive
User-Agent: SDI-Rest Tool
```

Response Panel: Represent the response obtained after the user request.

<pre>GET /api//domain/ag/TestPublishToCSB_B2BRequestForGreenho useAdvice HTTP/1.1 Host: localhost:9090 Connection: Keep-Alive User-Agent: SDI-Rest Tool</pre>	<pre>TestPublishToCSB_B2BRequestFForGreenhouseAdvice: OK</pre>
---	--

2.15 Flspace-Sketches Plugin

We have the possibility, like using a blackboard or note, to draw figures and lines, write text and others as a useful tool to define or create sketches about anything that we have in mind, and we need to write this in a note or sketch. Moreover, we can save this sketch or open any already created.

From the Flspace menu bar, select the option My Sketches.

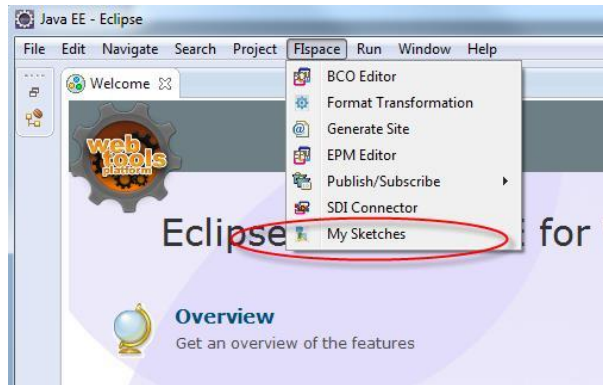


Figure 97: Menu item to launch MySketches plugin

Using the plugin

This tool provides the possibility to draw in a panel different kind of figures, write text, draw lines, change the colour of the figures and text, fill figures, save the content of the panel or open an image file to modify. We can subdivide these tools in groups of functionalities and describe how works each group.

The first group that contains the basic options to draw are shown in the next figure:

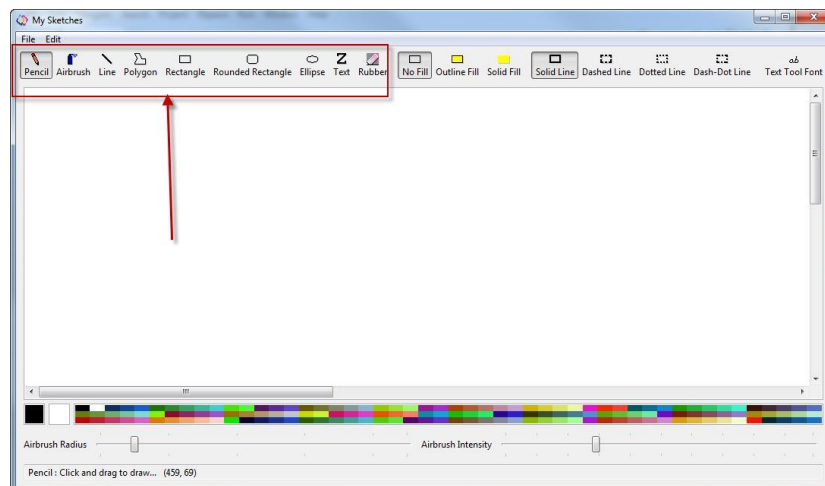


Figure 98: Basic options in MySketches

Special options are *Text* and *Rubber* options.

To use *Text* we need, firstly, click on the option and, with the right button of the mouse we can change the literal of the text that will appear in the screen.

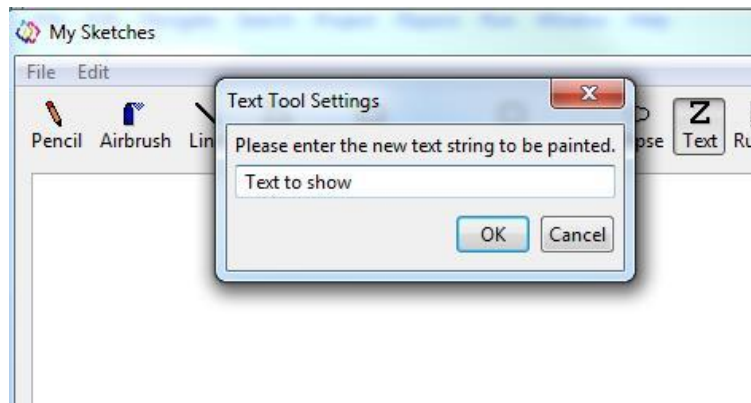


Figure 99: Text option in MySketches (I)

After this, we can press *OK* button and move the cursor in the desired position where we can insert the text and then, clicking left button of the mouse, the text is inserted.

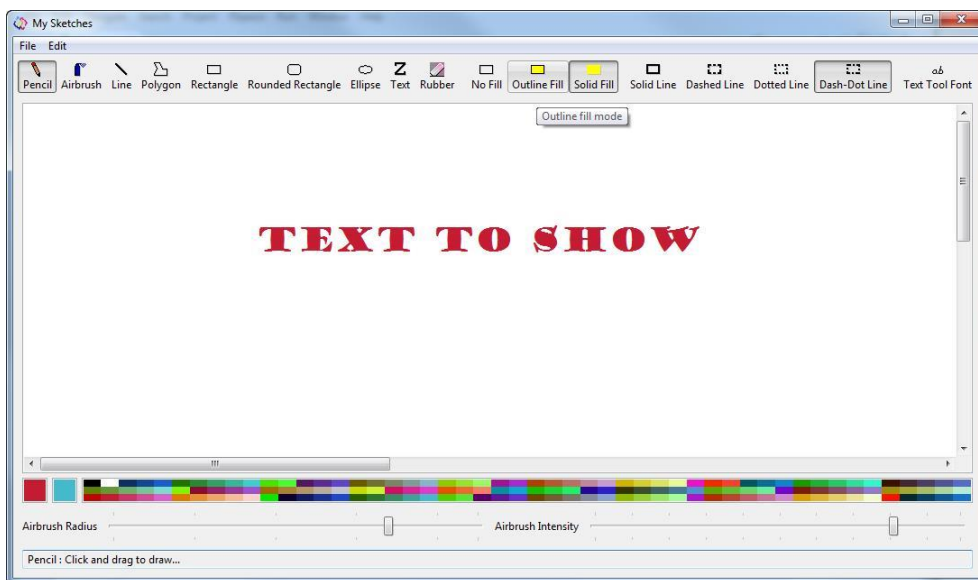


Figure 100: Text option in MySketches (II)

Rubber erases a specific area of the screen. We must move the cursor to the initial position to erase and, moving the cursor from left to right, it begins to erase the content selected.

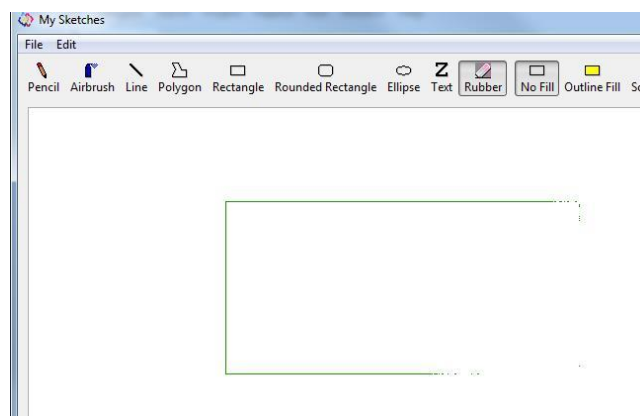


Figure 101: Rubber option in MySketches

The next group of functionalities to consider is related to *filling* forms:

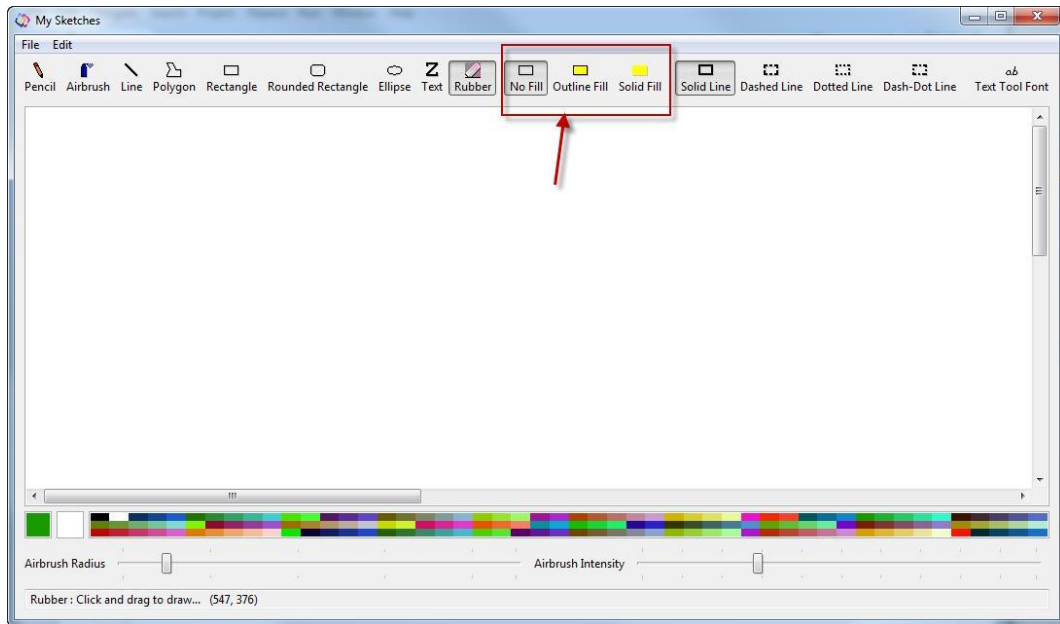


Figure 102: Filling options in MySketches (I)

Example of the result of apply Fill Options

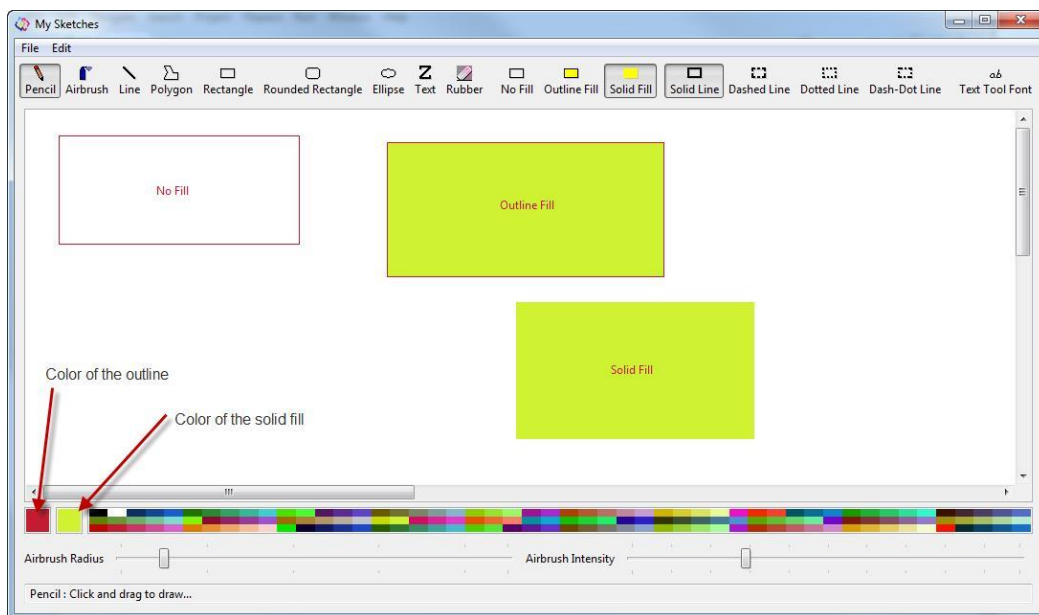


Figure 103: Filling options in MySketches (II)

The next group is related with the type of *line*, only applicable to lines, polygons, rectangles and ellipses.

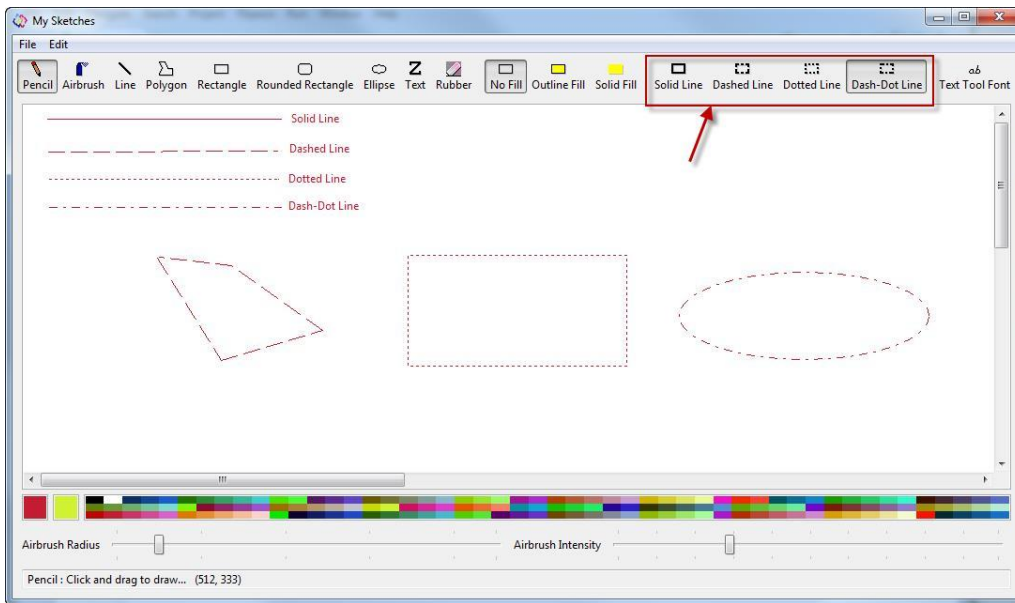


Figure 104: Lines in MySketches

The last group within in the main bar of options is *Text Tool Font*, where we can choose among different kinds of fonts and change the appearance, like bold, italic, etc.

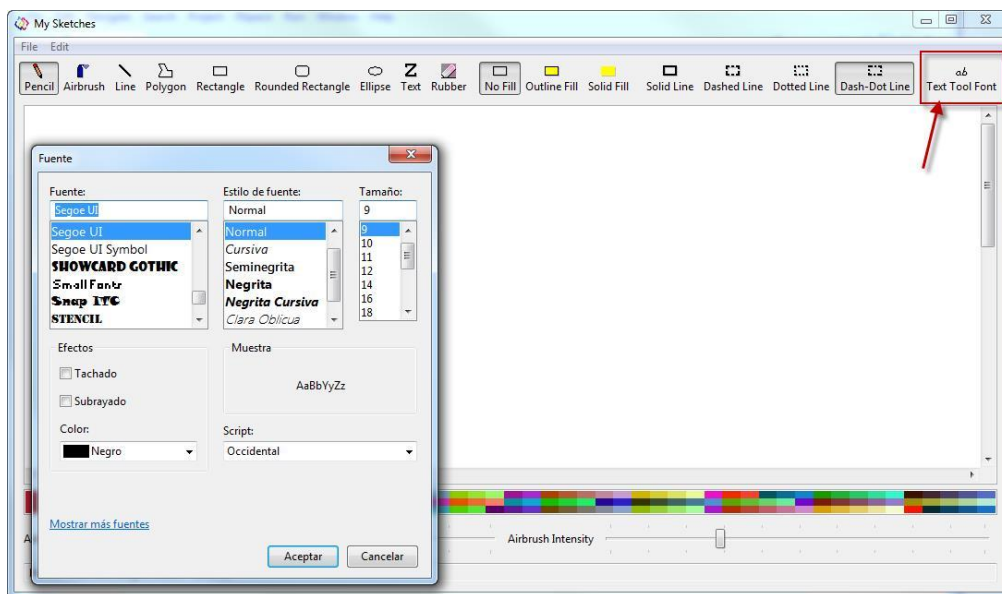
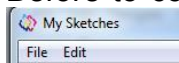


Figure 105: Text Tool Font option in MySketches

Before to continue, it's important to pay a mention about the main menu in the toolbar



Into the **File menu**, we can find the next options:

Load opens a File dialog and allows to load an image file (only jpeg, jpg, gif or png files) from the local machine into the drawing panel.

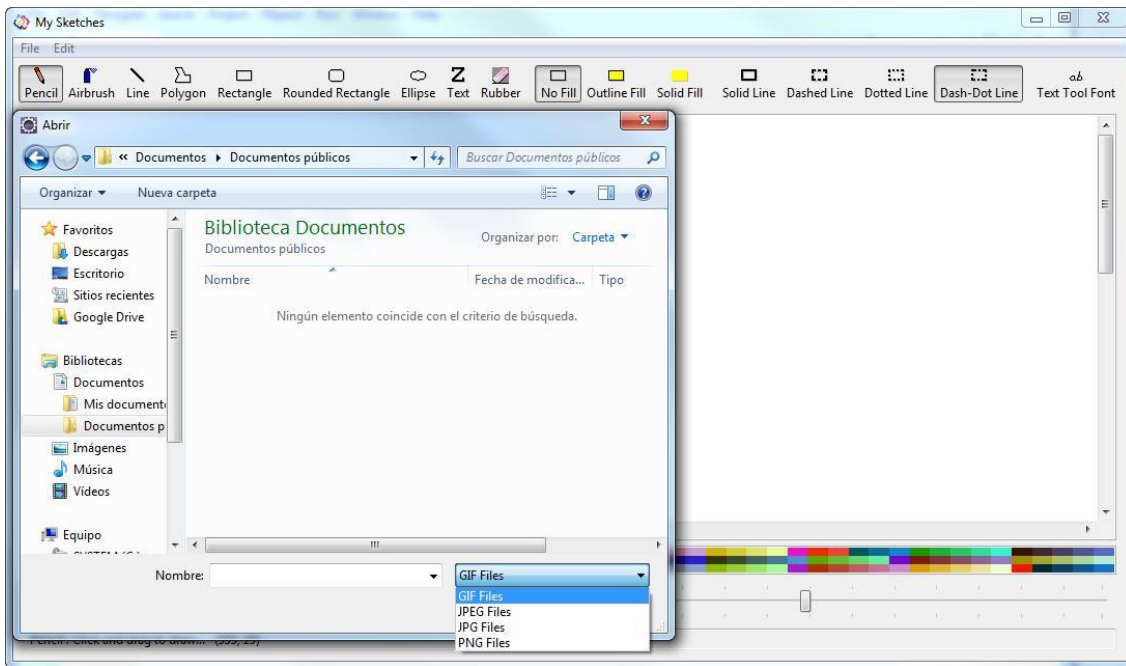


Figure 106: Load option in MySketches

Save allows the user to save into the local machine the content of the panel in form of image file (only jpeg, jpg, gif or png files).

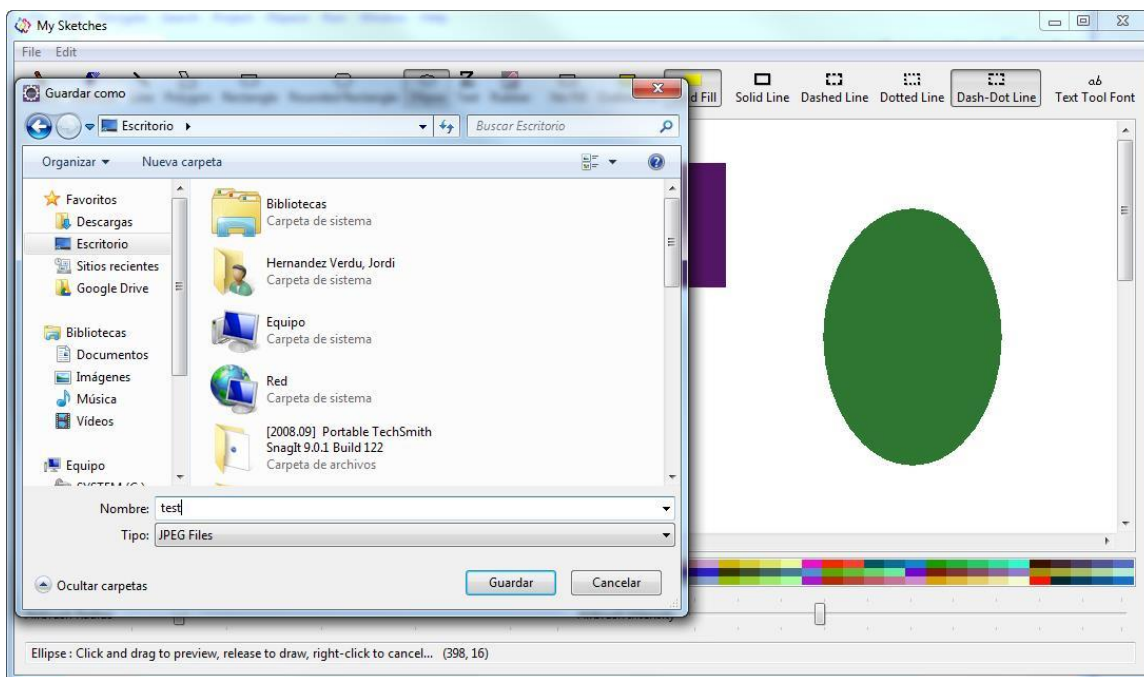
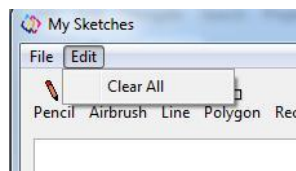


Figure 107: Save option in MySketches

Above figure shows the image saved as jpeg file.

Into the **Edit menu**, we can find the next option:



Clear All option allows the user to clear the entire drawing panel

The rest of the options located at the bottom of the screen are related with the *colours* of the lines and the filling of the figures.

We can see two squares indicating the current chosen colour. Left square indicates the *foreground colour* applied to the lines and figures. Right square indicates the *background colour* or fill colour applied to the figures.

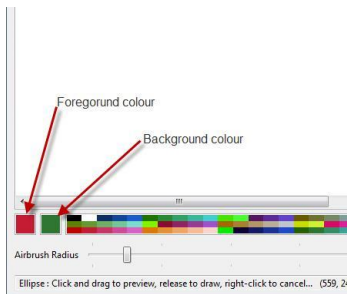


Figure 108: Colour option in MySketches

To select a colour is necessary to move the cursor with the mouse to the colours area.

To select the Foreground colour, with the left button of the mouse, click over the chosen colour of the colour area and, immediately, the Foreground square shows the chosen colour.

To select the Background colour, with the right button of the mouse, click over the chosen colour of the colour area and, immediately, the Background square shows the chosen colour.

The last options are *Airbrush Radio* and *Airbrush Intensity*. Airbrush Radio determines the thickness or diameter of the brush and Airbrush Intensity determines the opacity of the brush. As you can know, these options only work with Airbrush Option from the dashboard.

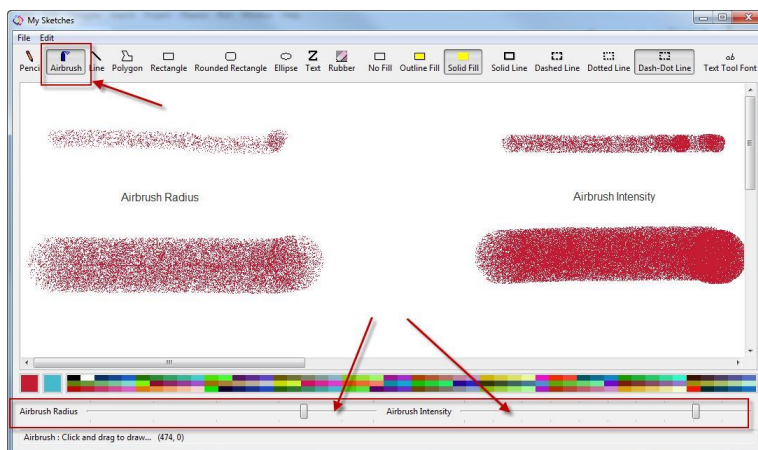


Figure 109: Airbrush options in MySketches

2.16 Flspace-SOAP Plugin

From the Bar menu select File -> New Project -> SOAP Client Generator, you can open this wizard.

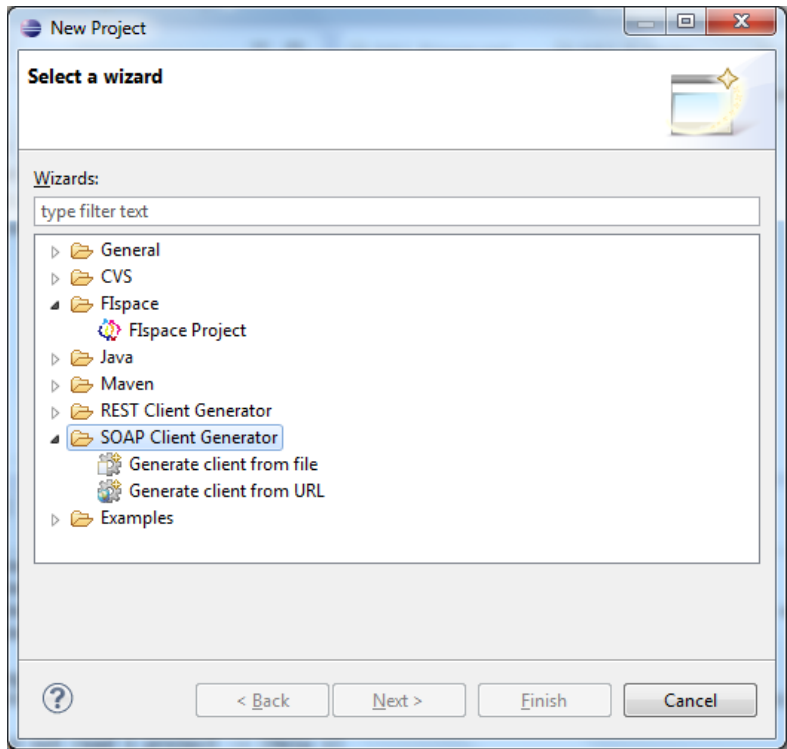


Figure 110: Flspace SOAP tool

If you want to generate a SOAP client from a file, choose the corresponding item.

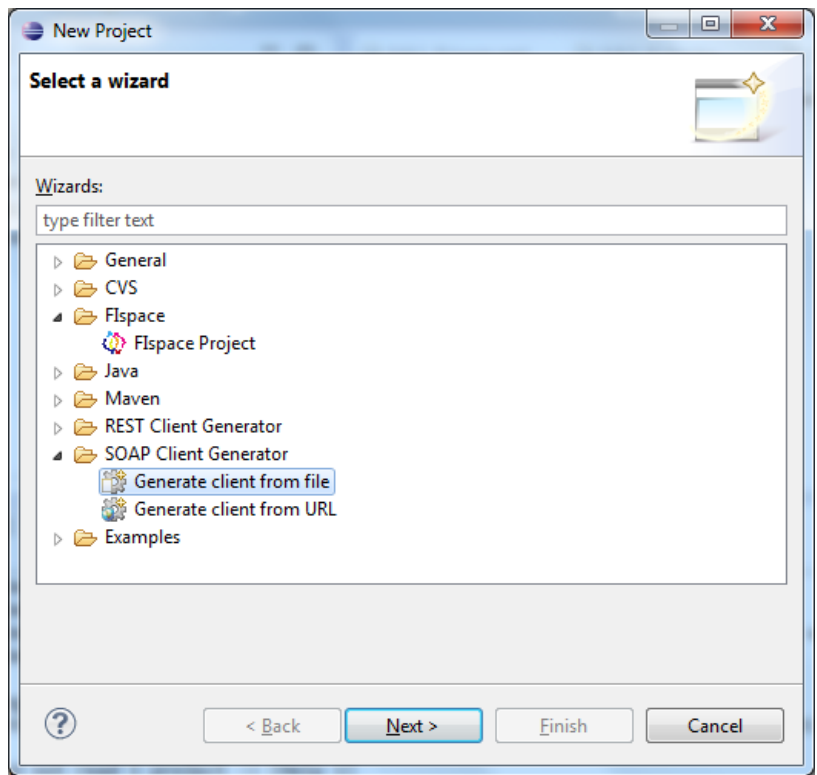


Figure 111: Flspace SOAP tool from file (I)

Click on Next button:

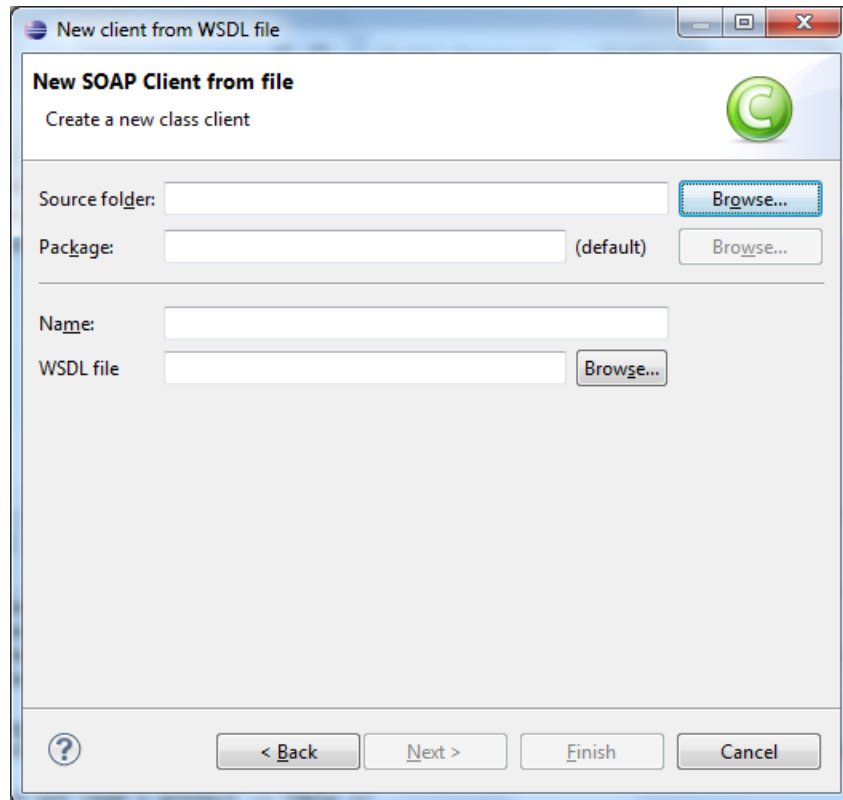


Figure 112: FIspace SOAP tool from file (II)

If you want to generate a SOAP client from an URL, choose the corresponding item.

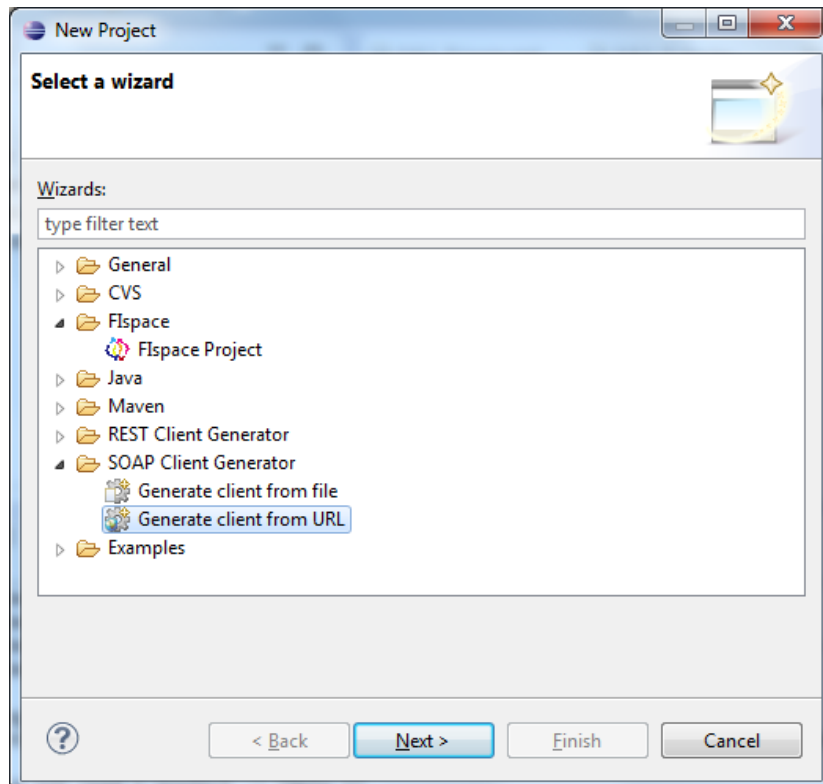


Figure 113: FIspace SOAP tool from URL (I)

Clicking on Next button:

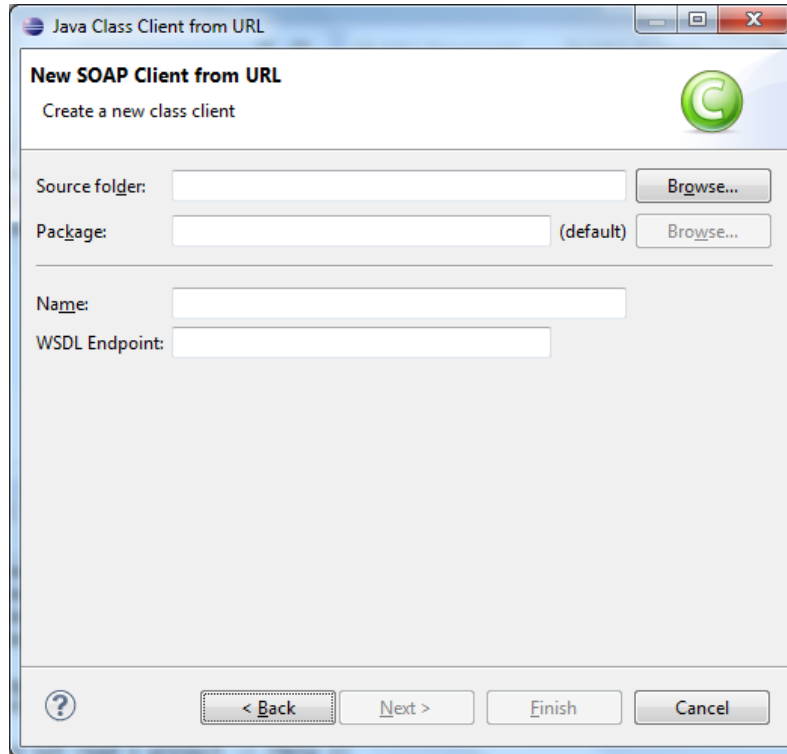


Figure 114: Flspace SOAP tool from URL (II)

2.17 Flspace-uploader Plugin

This plugin provides the connection SDK to Flspace Store. Flspace projects with a **widget** nature (widget and decoupled widget projects) can be uploaded to Flspace Store.

It is only available for widget nature projects, and the FlspaceStudio perspective being active. Right-click on a project containing a widget:

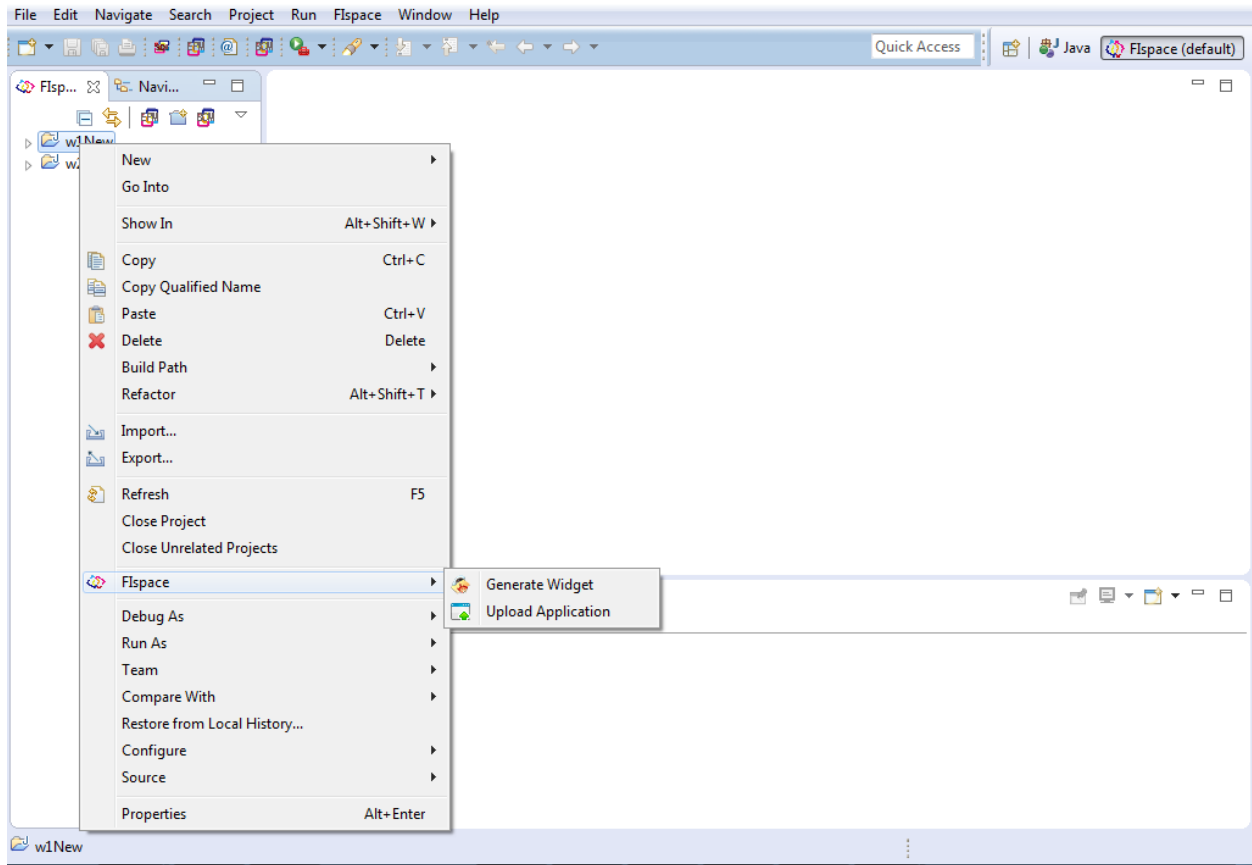


Figure 115: Flspace Widget Uploader

Selecting the Flspace (defaults) perspective, you can right-click on an app to be uploaded. Choose the Flspace entry from the menu and click in this order:

- First, **Generate Widget** to generate the widget and to be validated.
- Second, **Upload Application** to upload it (deploying and uploading).

The results can be seen in Console view. Notice the “*.wgt” file just generated.

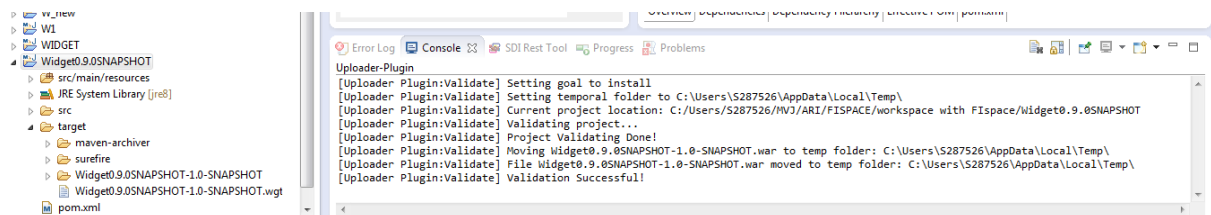


Figure 116: Flspace Widget generation results

Note: The uploading is currently under testing.

2.18 Flspace-wizard Plugin

Select from the main menu File --> New --> Project. It will then appear a "Select Wizard" where it has to be chosen Flspace -> Flspace Project and click on Next.

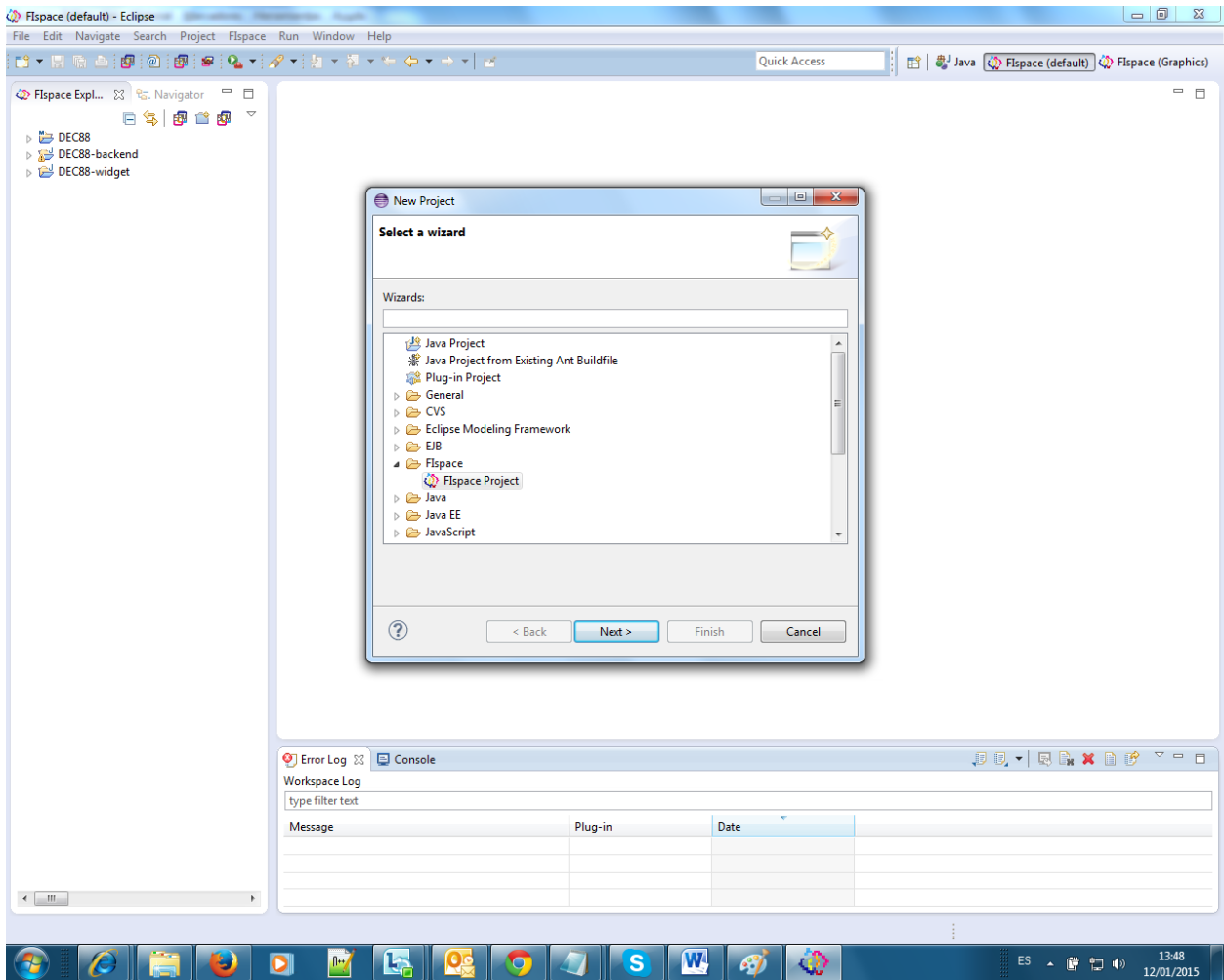


Figure 117: Flspace Project Wizard

The next form will appear.

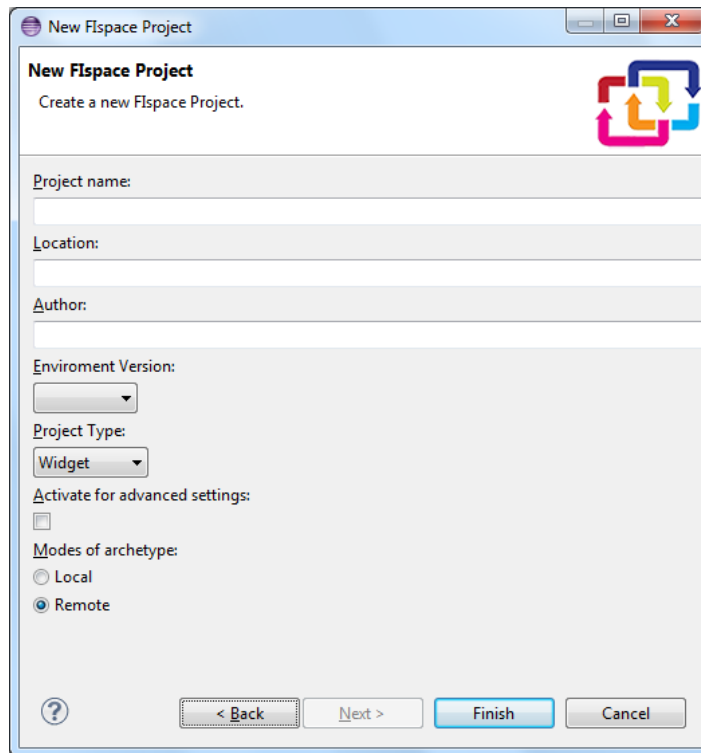


Figure 118: FISpace Project Wizard Form

In the form it is *mandatory* to inform "Project Name" and "Project Type". Once those data are filled, click "Finish". Wait till the process is completed. **Once it is completed, please refresh the new Project in the Package Explorer.**

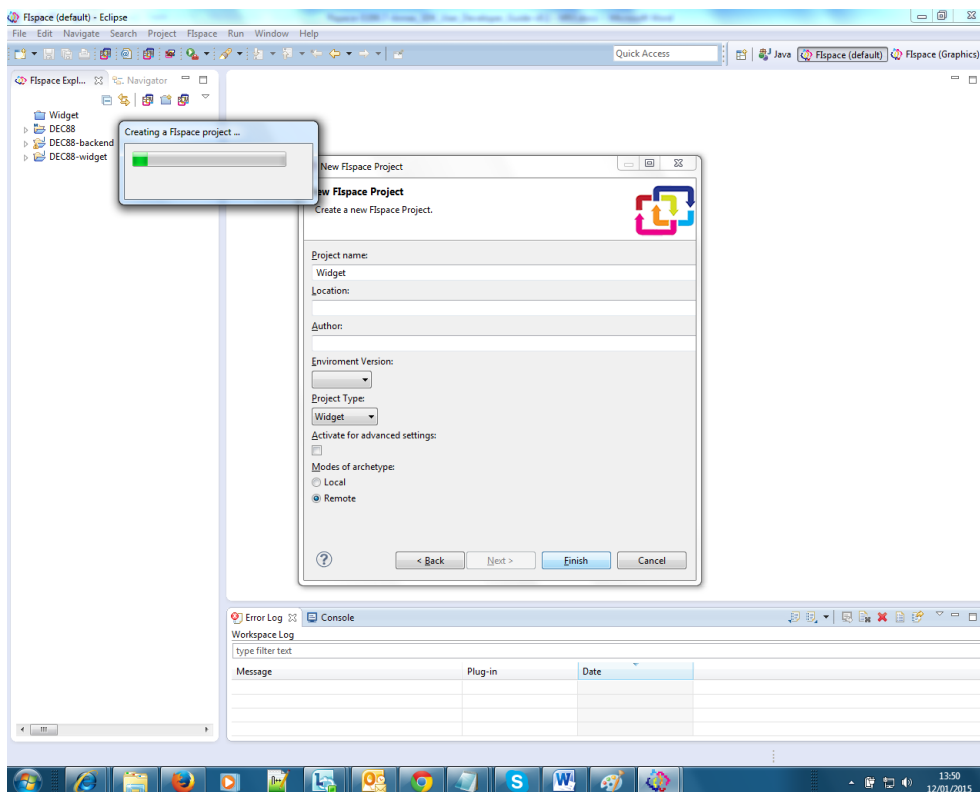


Figure 119: Waiting for FISpace project creation

In case you are under a proxy, you will have to follow the next instructions: Select from the main menu the option "Window" and then select "Preferences". Go to "Flspace Studio --> Settings" and complete the form.

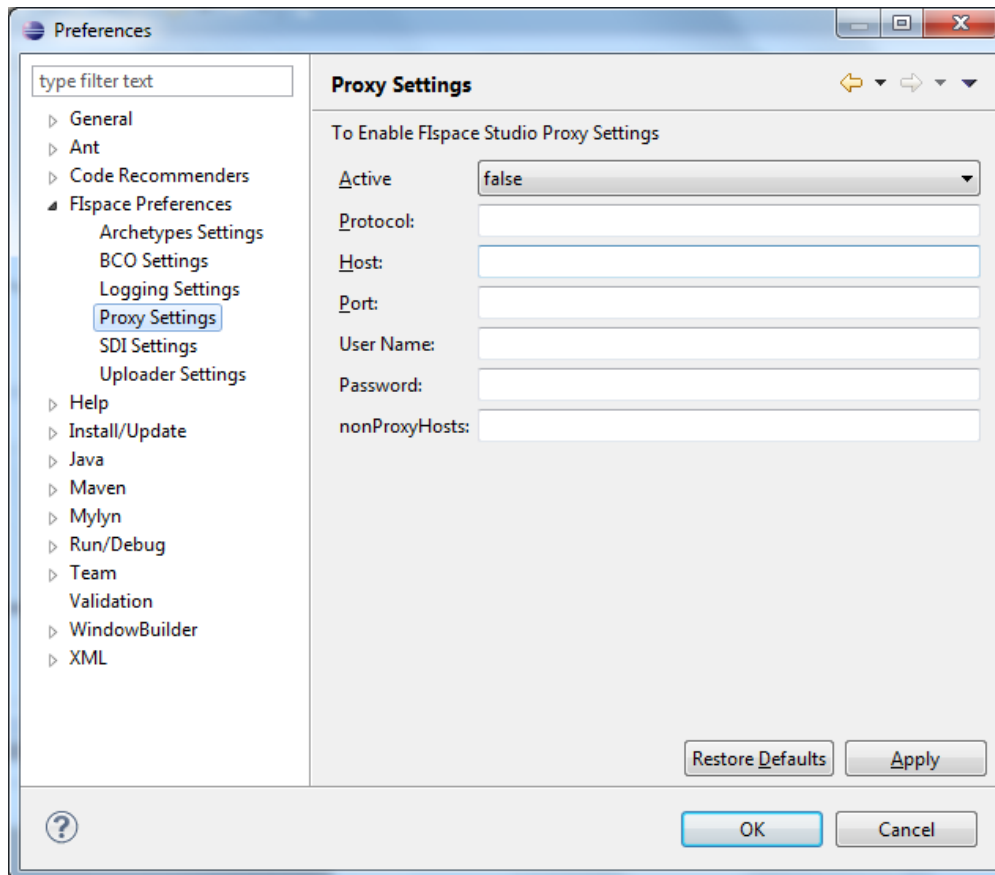


Figure 120: Flspace Proxy Settings

Next sections show several types of projects that can be created.

2.18.1 Widget Project

Widget refers to a small application with limited functionality that can be installed and executed within a web page by an end-user. The Widget project enables the development of *WireCloud* projects [18], which is the platform in which these Widgets can be installed and executed.

This simple project contains a single .html file with a "Hello World". It is also necessary to provide a config.xml file containing instructions of how the WireCloud should install and execute the widget. The widget project was designed to work *out of the box*, that is, once the project is created it can be packaged as a Widget and published into *Wirecloud* directly, without needing to configure the index.html and config.xml files.

A widget project is to be created:

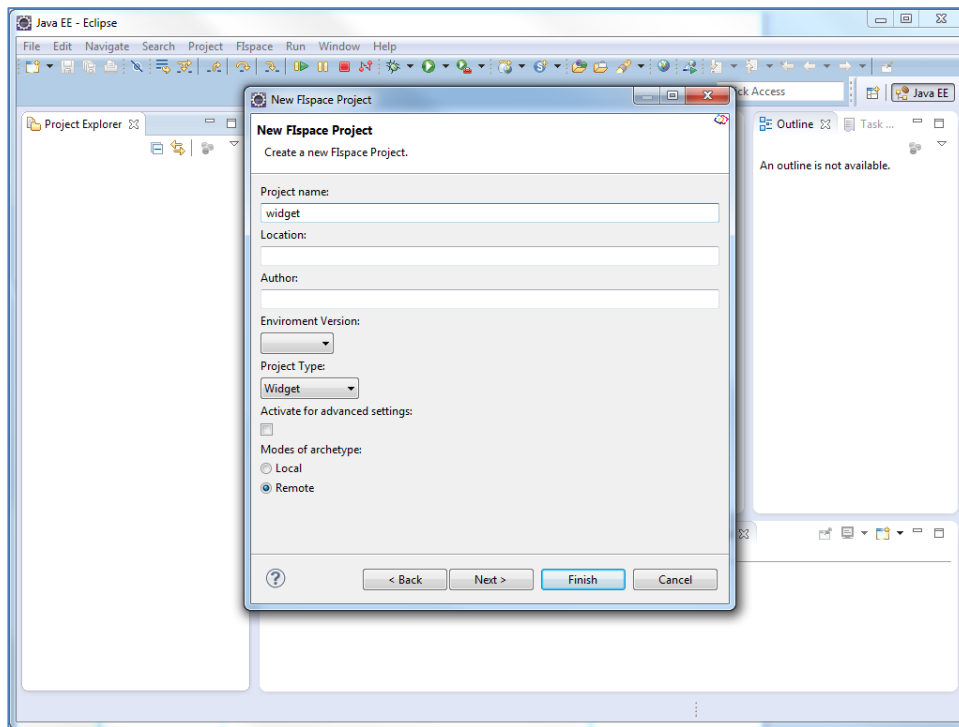


Figure 121: Flspace Widget project

After clicking on the Finish button, you obtain the following widget project structure.

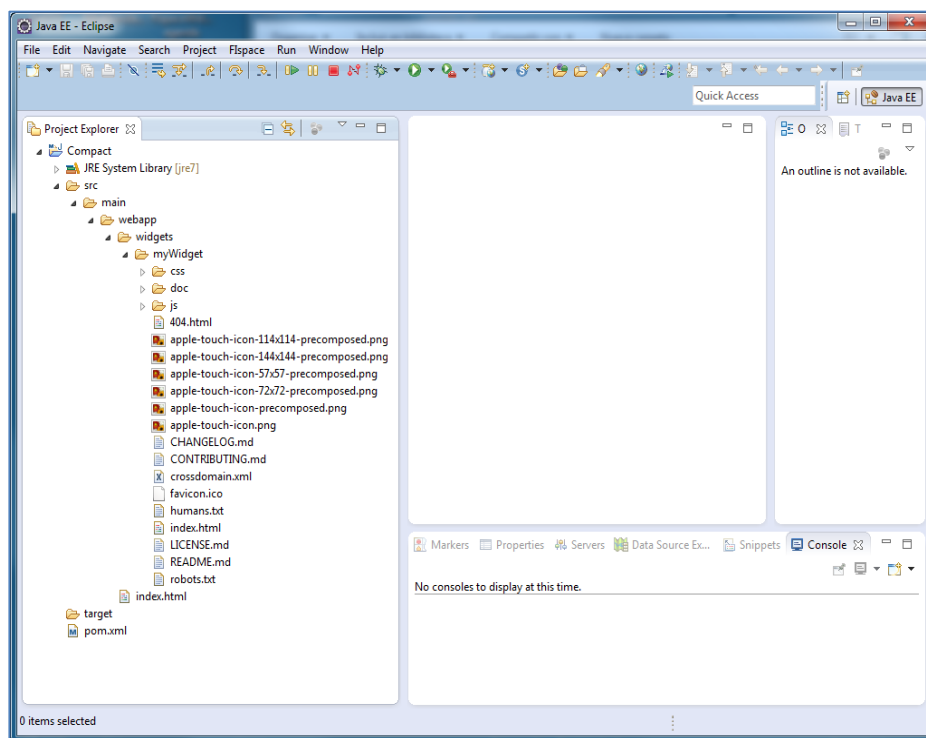


Figure 122: Flspace Widget project structure

2.18.2 Compact Project

The compact archetype also contains a Widget for *WireCloud* [18], but, unlike the Widget project, this one is more complex, with HTML5 features. The HTML template

Boilerplate² was used to implement this Widget. According to the authors, this template helps you build fast, robust and adaptable web apps or sites.

A compact project is to be created:

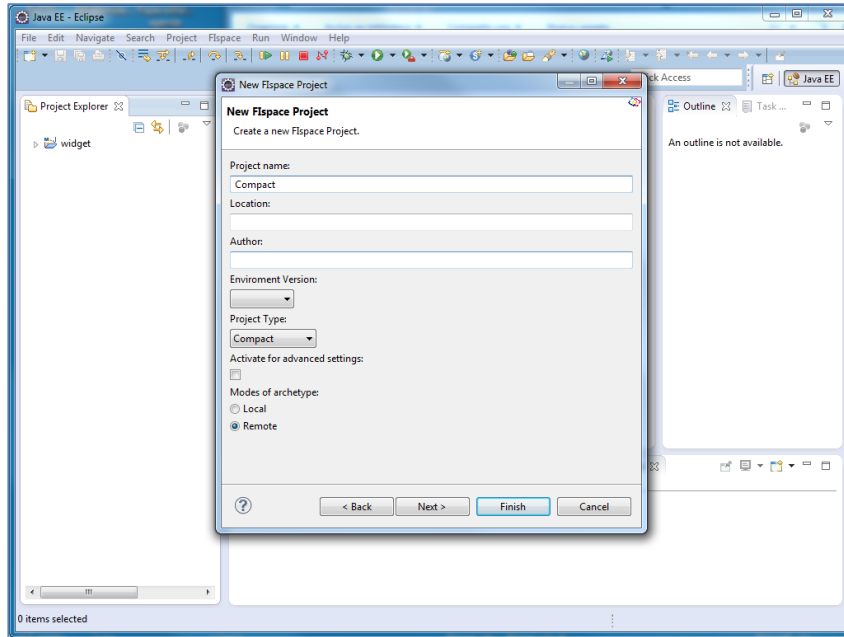


Figure 123: Flspace Compact project

After clicking on Finish button, you obtain the following compact project structure.

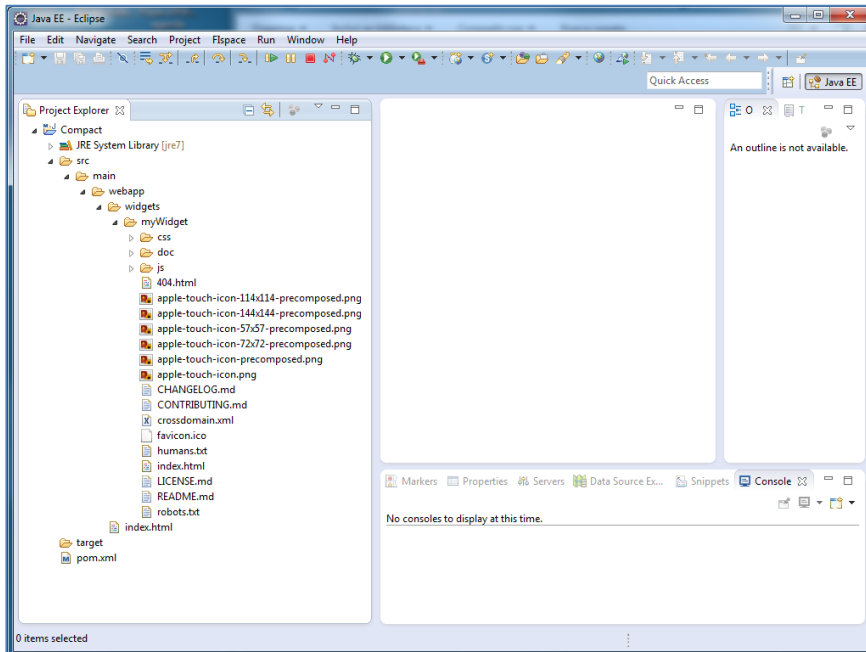


Figure 124: Flspace Compact project structure

² <http://html5boilerplate.com/>

2.18.3 Decoupled Project

The decoupled project consists in two parts, the widget part and the backend part. The widget part is the same as the one provided by the Widget project. The backend part provides support to the widget part giving some java classes that can be used to connect both sides.

So that the backend part can connect to the Widget, the backend must be deployed by using a java-based application container, such as Jetty Web Server (<http://www.eclipse.org/jetty/>).

The backend side contains a class called *FlspaceClient.java*, which prepares the backend to connect to Flspace through the SDI, importing the libraries *eu.fspace.api*.

This class provides some examples of requests and responses to Flspace.

A decoupled project is to be created.

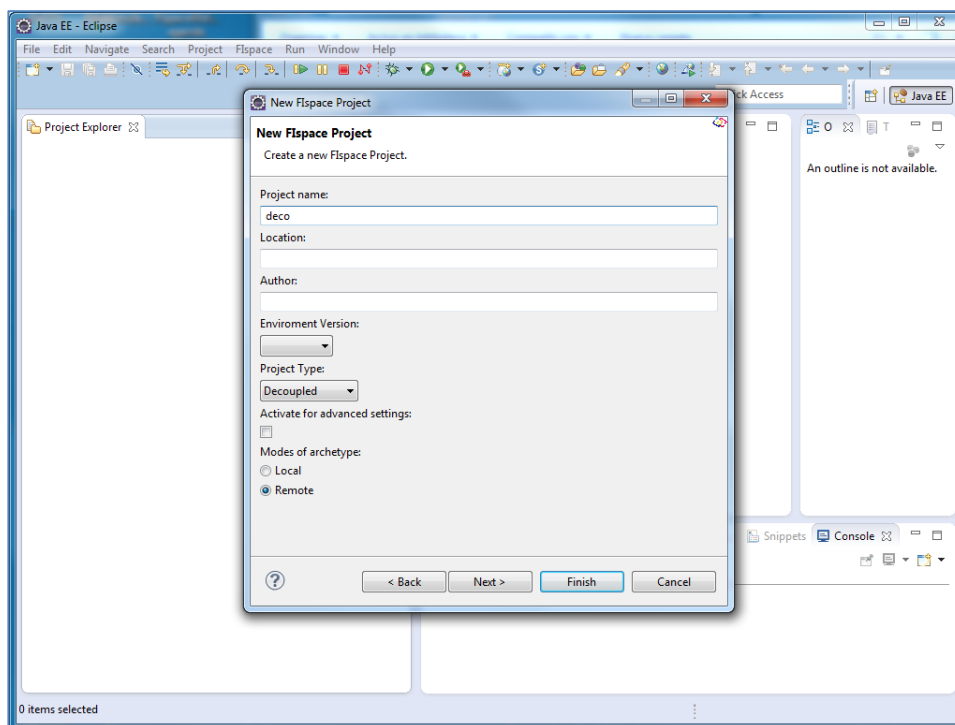


Figure 125: Flspace Decoupled project

After clicking on Finish button, you obtain the following decoupled project structure. Notice the three resulting projects.

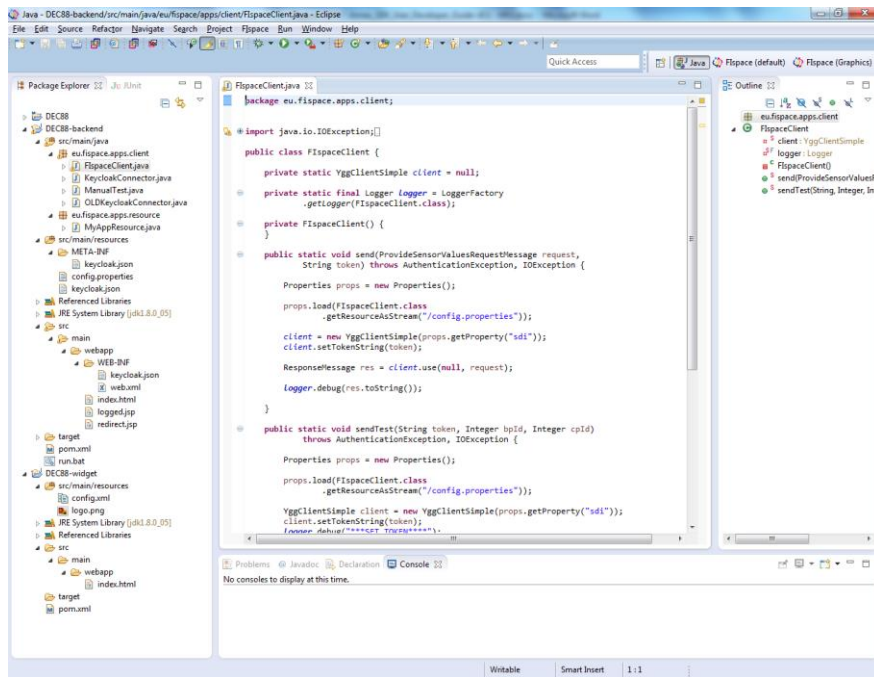


Figure 126: Flspace Decoupled project structure

2.18.4 Node.js Project

The Node.js project represents a basic Node.js project, which helps users to start coding. Node.js is a runtime environment and a library for running applications written in JavaScript outside the browser (for example, on the server). Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

The structure of the project is compatible to Node.js projects, and also compatible to Express [23], a web application framework for node, widely used.

A node.js project is to be created.

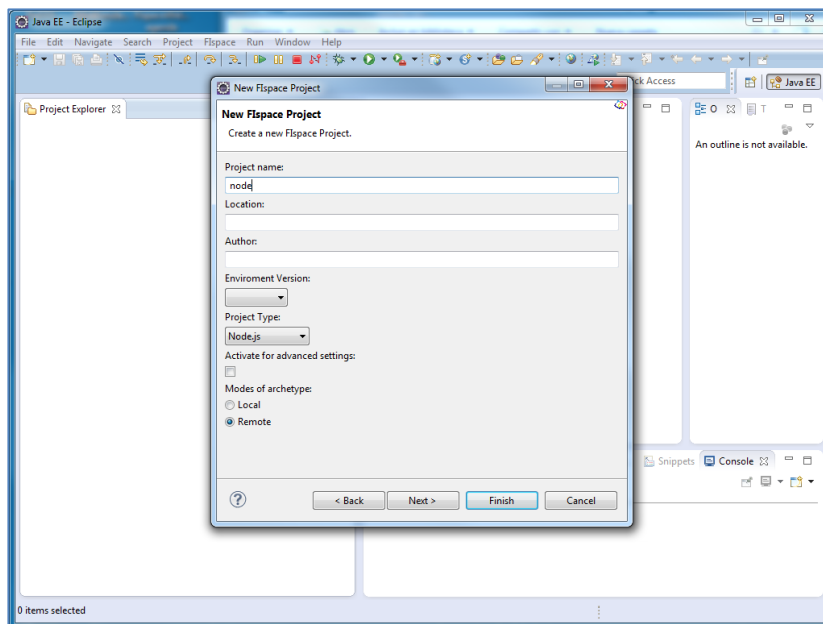


Figure 127: Flspace Node.js project

After clicking on Finish button, you obtain the following node.js project structure.

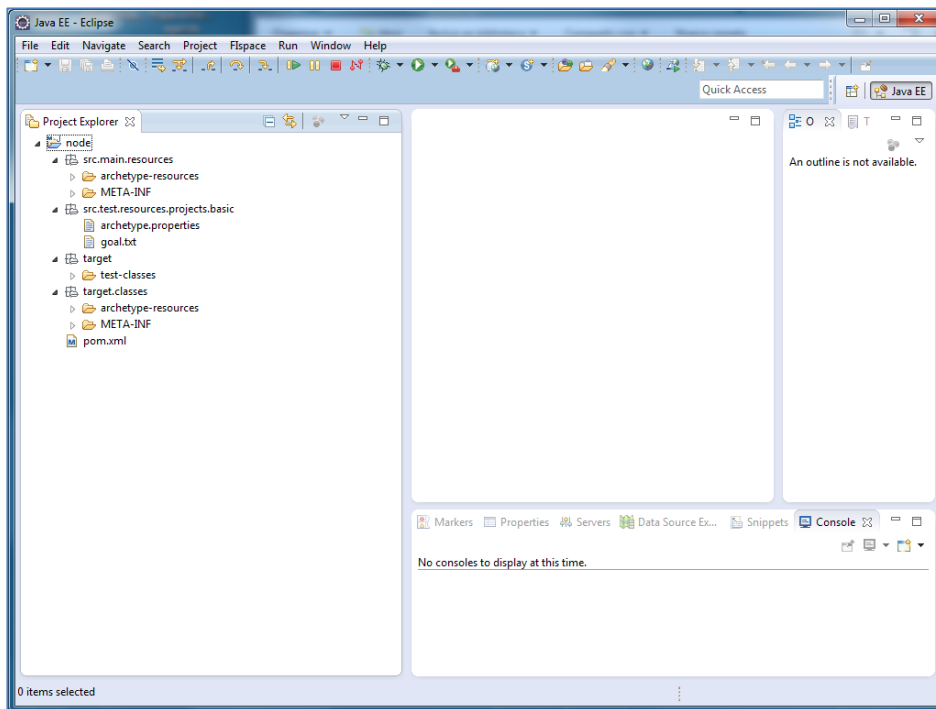


Figure 128: Flspace Node.js project structure

2.18.5 JavaFX Project

JavaFX is a rich client platform for enterprise business applications. It reuses Java libraries and allows developers to access native system capabilities. The JavaFX project provides a basic JavaFX application to interconnect to Flspace capabilities through the SDI. To do that it uses the *NotificationMessage* class to send a basic message when a button is pressed in the Java GUI.

A javaFX project is to be created.

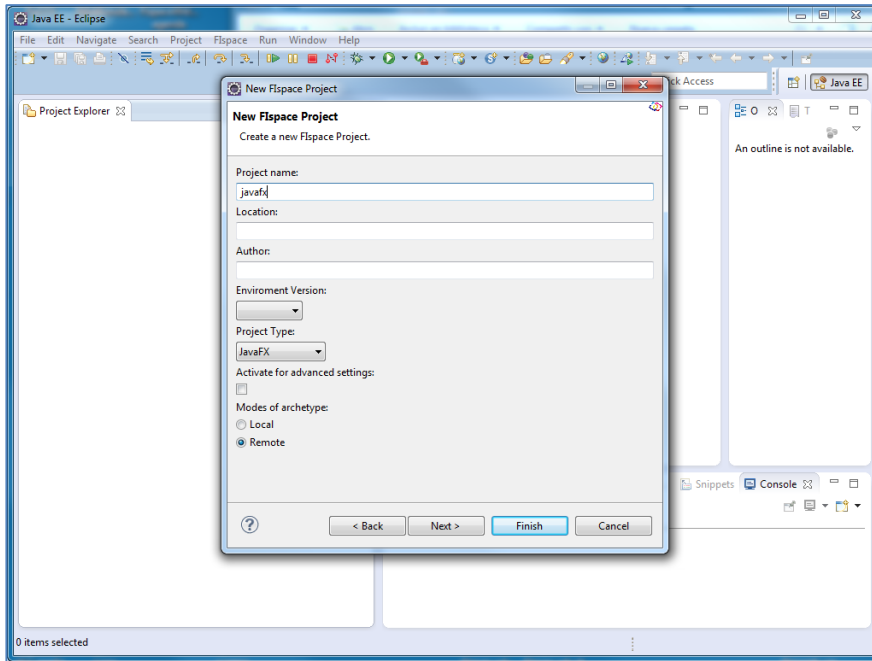


Figure 129: Flspace JavaFX project

After clicking on Finish button, you obtain the following javaFX project structure.

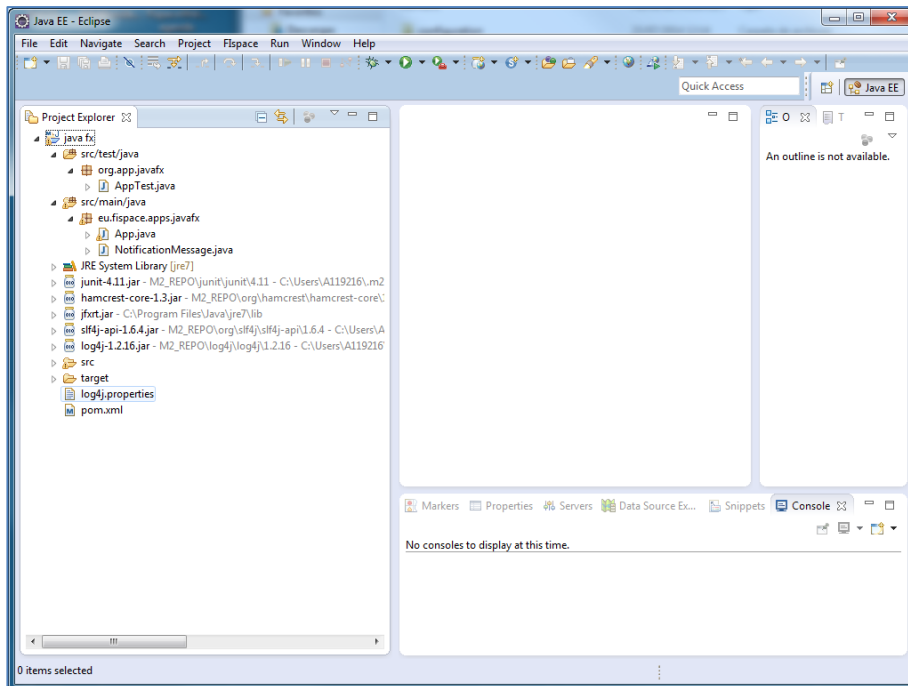


Figure 130: Flspace JavaFX project structure

2.18.6 RCP app archetype

RCP (*Rich Client Platform*) [27] constitutes the minimal set of plug-ins needed to build a rich client application (platform application with a UI). The RCP project enables Eclipse [27] to be used in a wide range of end-user applications that are not Integrated Development Environments (IDEs).

A RCP project is to be created:

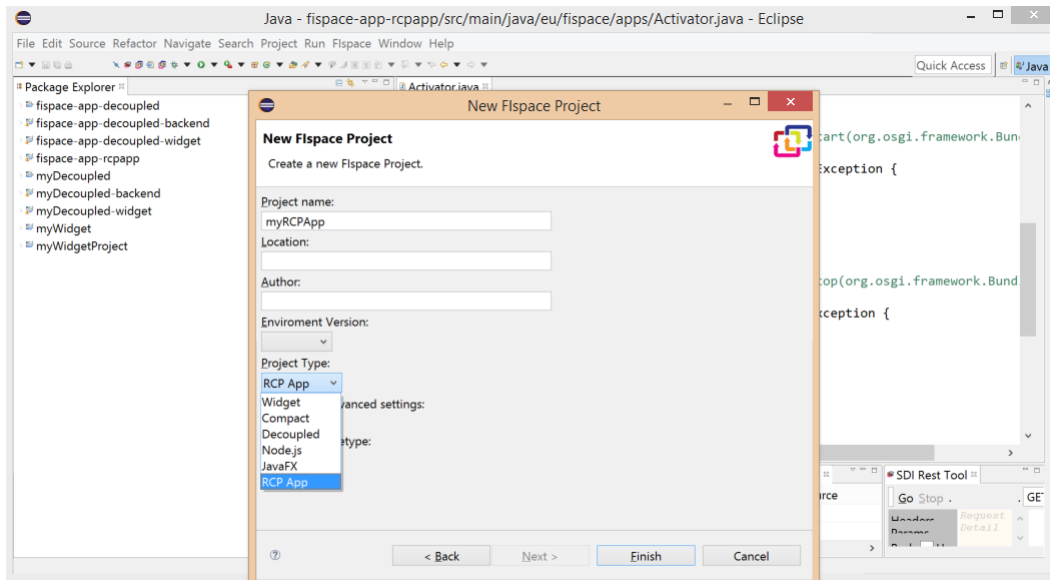


Figure 131: Fispace RCP project

After clicking on Finish button, you obtain the following RCP project structure.

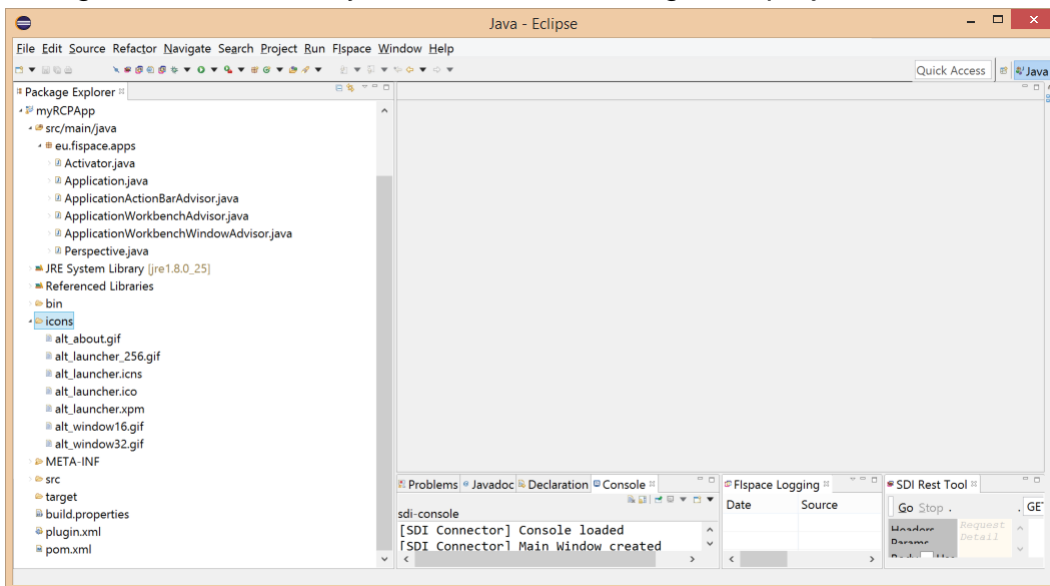


Figure 132: Fispace RCP project structure

After a successful new project generation with the wizard plugin, the next steps are required to launch the RCP App:

1. Open plugin.xml file
2. Click on “Launch an eclipse application”:

Testing

Test this plug-in by launching a separate Eclipse application:

[Launch an Eclipse application](#)

[Launch an Eclipse application in Debug mode](#)

Figure 133: Launching Flspace RCP project

3. Once the project is launched as an “Eclipse Application”, the application “Hello RCP” is displayed:

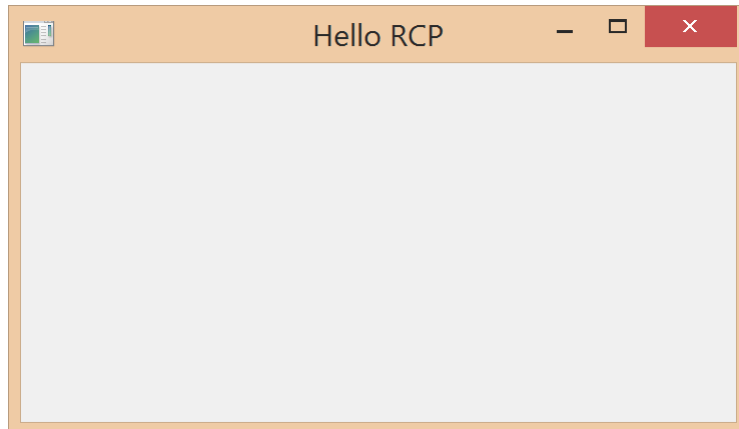


Figure 134: Default RCP project GUI

2.19 FIspace-xmleditor Plugin

Double click over config.xml file should open the config.xml file in the FIspace editor.

FIspace XML editor provides structured form-based UI for viewing and editing Wire-Cloud config.xml.

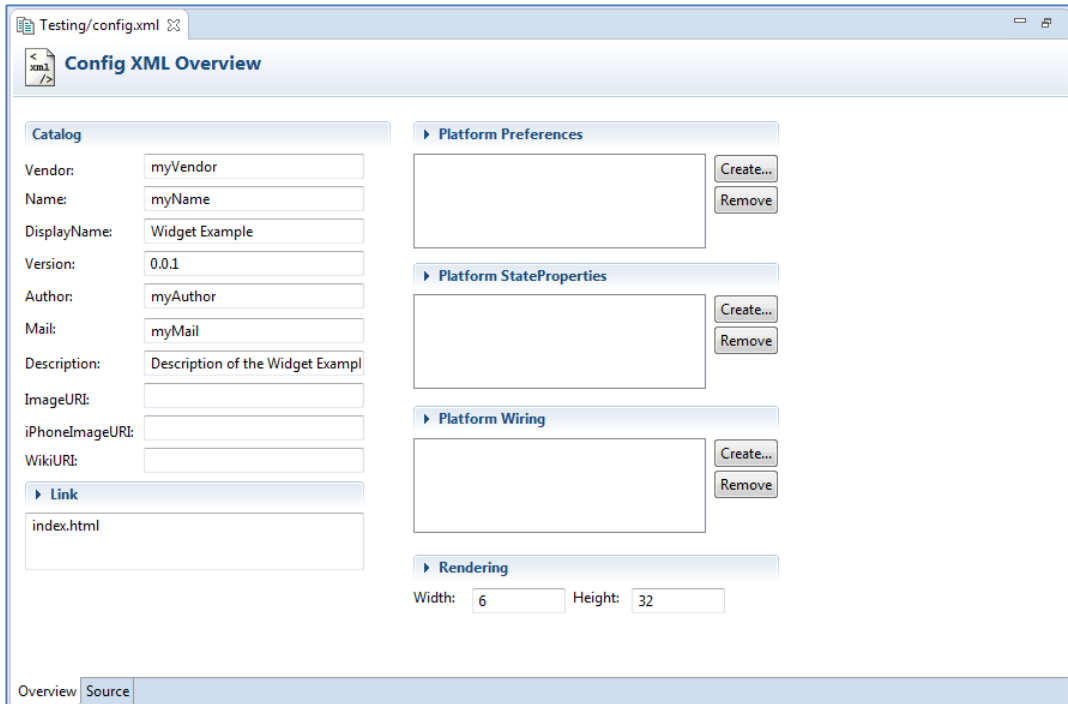


Figure 135: FIspace XML-editor

Form-based UI

The form-based POM editor UI provides rich editing and navigation features. See figure above.

Platform preferences

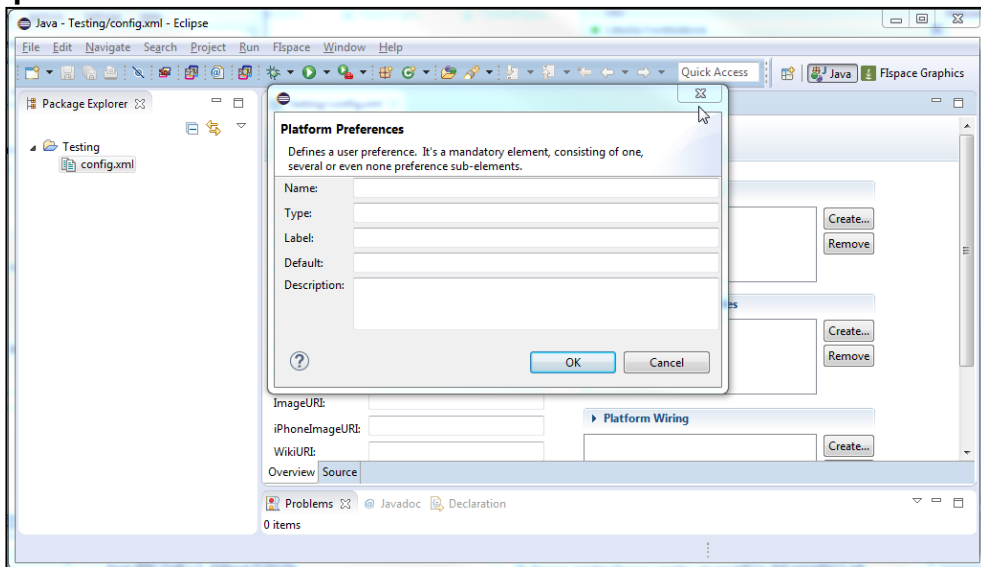


Figure 136: FIspace XML Editor Form Platform preferences form

Platform StateProperties

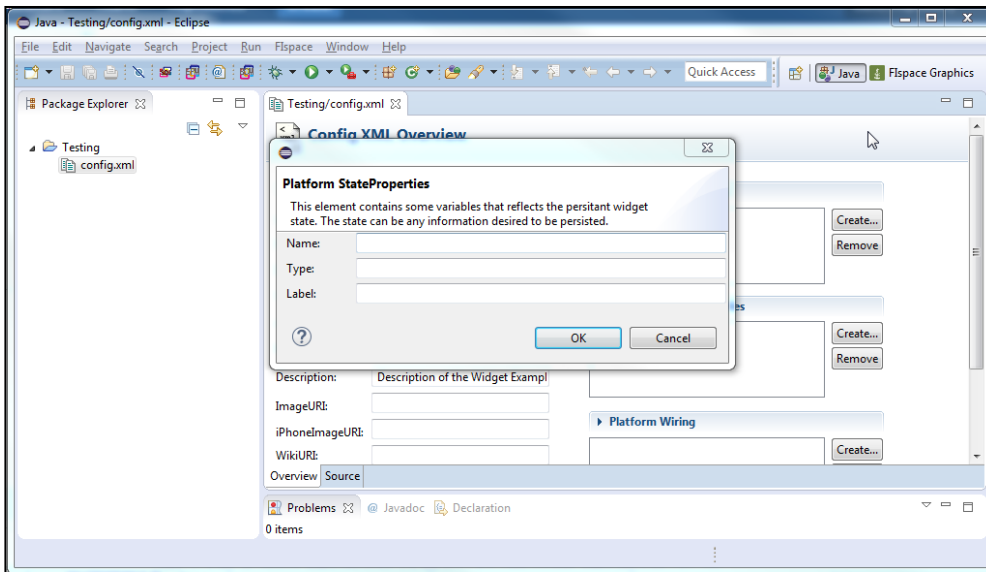


Figure 137: Flspace XML Editor Form Platform State Properties form

Platform Wiring

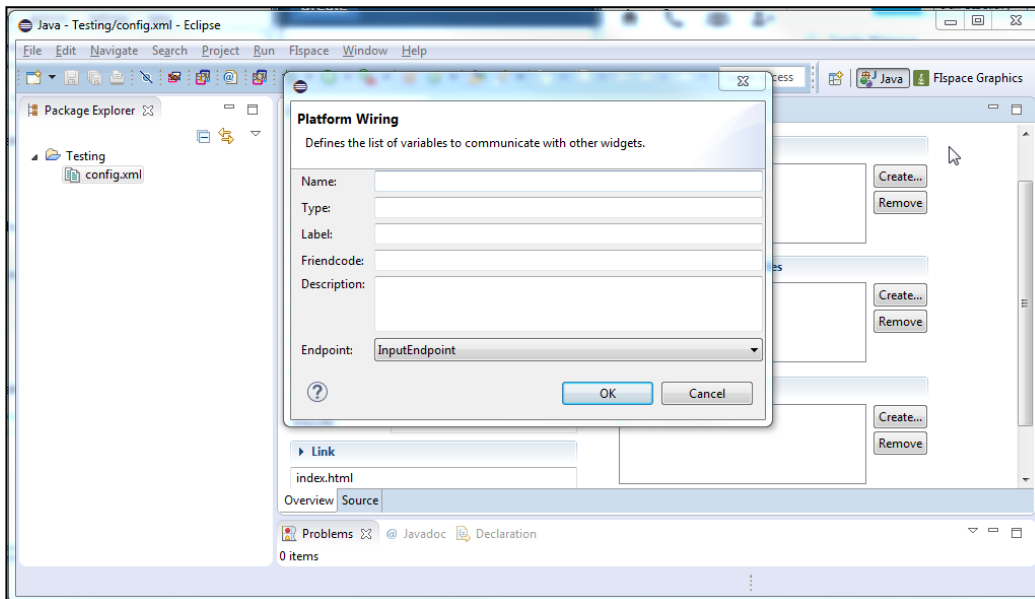
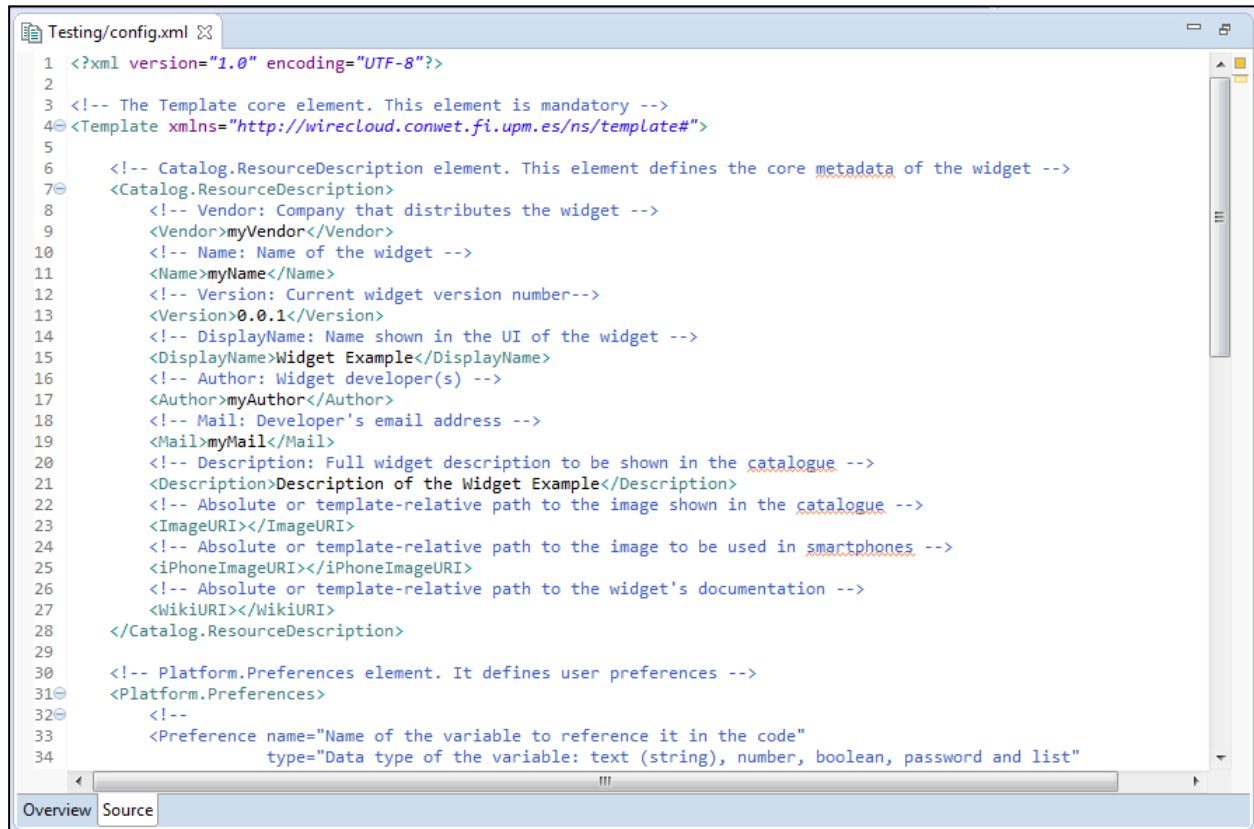


Figure 138: Flspace XML Editor Form Platform Wiring form

XML Editor Text View

Text View remains the most popular XML editing view of Flspace, and we continue to add new features to facilitate XML development. Intelligent XML editing is provided with syntax colouring, code completion, context-sensitive entry helpers, and Smart Fix validation for automatically correcting validation errors (See Figure below).



```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <!-- The Template core element. This element is mandatory -->
4 <Template xmlns="http://wirecloud.conwet.fi.upm.es/ns/template#"
5
6 <!-- Catalog.ResourceDescription element. This element defines the core metadata of the widget -->
7 <Catalog.ResourceDescription>
8 <!-- Vendor: Company that distributes the widget -->
9 <Vendor>myVendor</Vendor>
10 <!-- Name: Name of the widget -->
11 <Name>myName</Name>
12 <!-- Version: Current widget version number-->
13 <Version>0.0.1</Version>
14 <!-- DisplayName: Name shown in the UI of the widget -->
15 <DisplayName>Widget Example</DisplayName>
16 <!-- Author: Widget developer(s) -->
17 <Author>myAuthor</Author>
18 <!-- Mail: Developer's email address -->
19 <Mail>myMail</Mail>
20 <!-- Description: Full widget description to be shown in the catalogue -->
21 <Description>Description of the Widget Example</Description>
22 <!-- Absolute or template-relative path to the image shown in the catalogue -->
23 <ImageURI></ImageURI>
24 <!-- Absolute or template-relative path to the image to be used in smartphones -->
25 <iPhoneImageURI></iPhoneImageURI>
26 <!-- Absolute or template-relative path to the widget's documentation -->
27 <WikiURI></WikiURI>
28 </Catalog.ResourceDescription>
29
30 <!-- Platform.Preferences element. It defines user preferences -->
31 <Platform.Preferences>
32 <!--
33 <Preference name="Name of the variable to reference it in the code"
34 <!-- type="Data type of the variable: text (string), number, boolean, password and list"
```

Figure 139: Flspace XML Editor Text

2.20 Flspace-logging Plugin

In this plugin we have the **following extensions** (defined in plugin.xml):

1. PreferencePages:

- Entry in settings menu: Eclipse / Preferences
- Let the user choose:
 - Logs file directory
 - Server Route
 - Username and password
- Desirable (Filter levels, schemas of colours, ...)

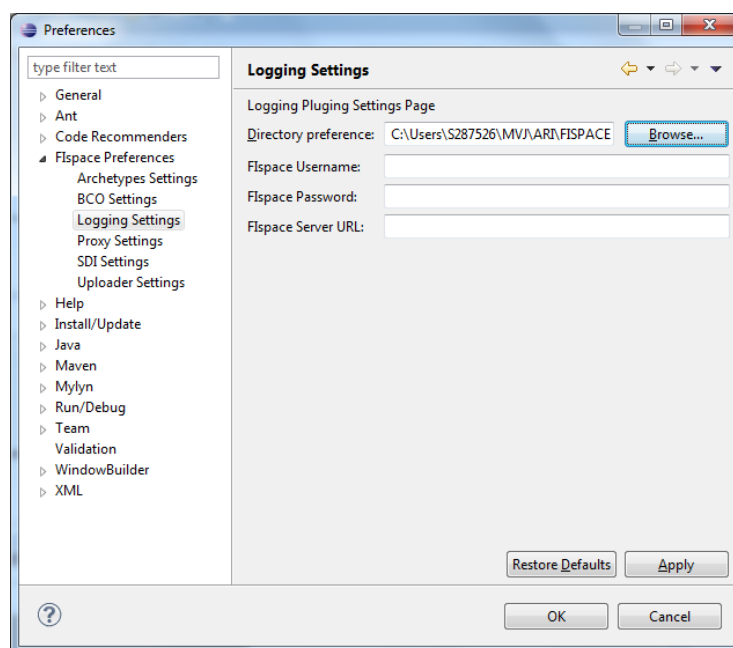


Figure 140: Logging Settings

2. Views:

Flspace Logging Plugin let users, among others, monitor logs from different Flspace projects in a single view (Java Perspective).

Date	Source	Message Level	Message
Fri May 30 08:53:25 CES...	com.example.helloworld.HelloWorld: (HelloWorl...	ERROR	Log4j ERROR socket appender test run successfully!!
Fri May 30 08:53:25 CES...	com.example.helloworld.HelloWorld: (HelloWorl...	DEBUG	Log4j DEBUG socket appender test run successfully!!
Fri May 30 08:53:25 CES...	com.example.helloworld.HelloWorld: (HelloWorl...	INFO	Log4j INFO socket appender test run successfully!!
Fri May 30 08:53:25 CES...	com.example.helloworld.HelloWorld: (HelloWorl...	WARN	Log4j WARN socket appender test run successfully!!
Fri May 30 08:53:31 CES...	com.example.helloworld.HelloWorld: (HelloWorl...	ERROR	Log4j ERROR socket appender test run successfully!!
Fri May 30 08:53:31 CES...	com.example.helloworld.HelloWorld: (HelloWorl...	DEBUG	Log4j DEBUG socket appender test run successfully!!
Fri May 30 08:53:31 CES...	com.example.helloworld.HelloWorld: (HelloWorl...	INFO	Log4j INFO socket appender test run successfully!!
Fri May 30 08:53:31 CES...	com.example.helloworld.HelloWorld: (HelloWorl...	WARN	Log4j WARN socket appender test run successfully!!

Figure 141: Flspace Logging View

3. Configure Flspace projects to use Logging Plugin:

Prerequisites:

1. Use SLF4J APIs for logging. Tested version (slf4j-api-1.7.7.jar).
2. Use Log4j implementation of SLF4J. Tested version (slf4j-log4j12-1.7.7.jar).
3. log4j.jar is required to use Property Configurator.
4. All log messages will be stored in a local log file. Also it will be sent to console.

Example of how to use Log4j implementation:

```
package com.example.helloworld;

import org.apache.log4j.PropertyConfigurator;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class HelloWorld {
    /**
     * Logger for this class
     */
    final static Logger logger = LoggerFactory.getLogger(HelloWorld.class);

    public static void main(String[] args) {

        PropertyConfigurator.configure("log4j.properties");

        logger.error("Log4j ERROR run successfully!!");

        logger.debug("Log4j DEBUG run successfully!!");

        logger.info("Log4j INFO run successfully!!");

        logger.warn("Log4j WARN run successfully!!");

    }
}*
```

```
#Define the log4j configuration for local application
log4j.rootLogger=DEBUG, CA, LOGFILE

#Console Appender
log4j.appender.CA=org.apache.log4j.ConsoleAppender
log4j.appender.CA.layout=org.apache.log4j.PatternLayout
log4j.appender.CA.layout.ConversionPattern=%d--%p--%c: (%13F:%L)--%m%n

# LOGFILE is set to be a DailyRollingFileAppender appender using a PatternLayout.

log4j.appender.LOGFILE = org.apache.log4j.DailyRollingFileAppender
log4j.appender.LOGFILE.File=${user.home}/fispaces-logs/logFile.log
log4j.appender.LOGFILE.DatePattern='.'yyyy-MM-dd
log4j.appender.LOGFILE.layout=org.apache.log4j.PatternLayout
log4j.appender.LOGFILE.layout.conversionPattern=%d--%p--%c: (%13F:%L)--%m%n
log4j.appender.FA.Threshold=DEBUG
```

Figure 142: Configuring projects to use Logging Plugin

5. Configure a logging server

In “Eclipse preferences”, select the Flspace Logging Preferences:

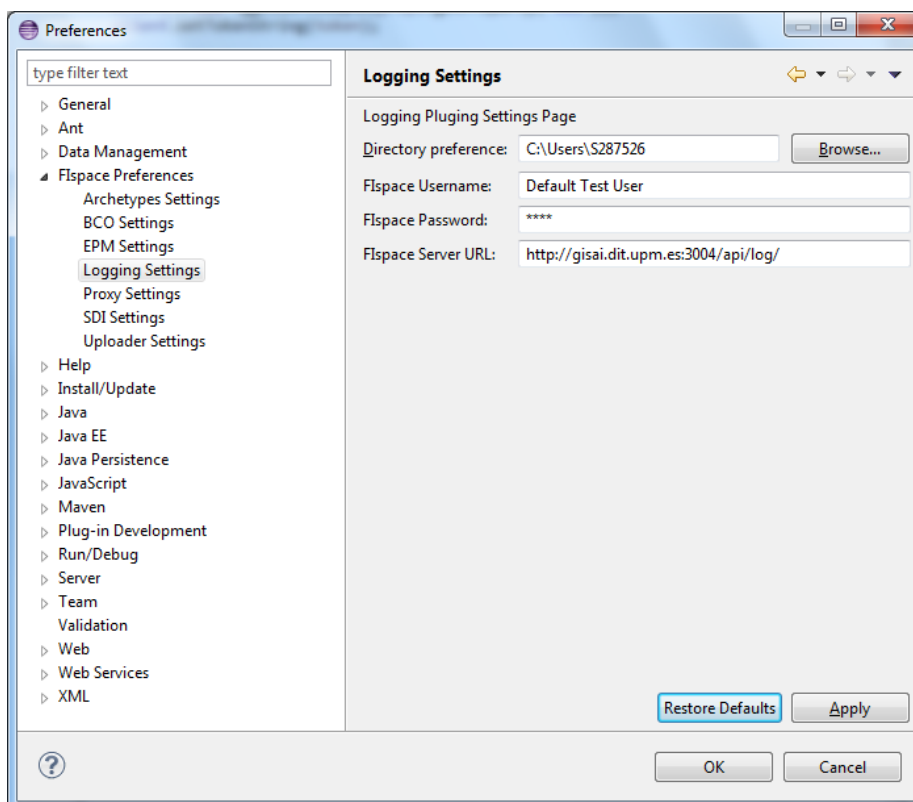


Figure 143: Flspace Logging preferences (defaults)

Press the “Restore defaults” button to fill the fields with default values or write the corresponding parameters:

Default URL for the Logging server API is: <http://gisai.dit.upm.es:3004/api/log>

New logs will be sent to the server automatically.

6. Review logged logs

For reviewing purposes, a web interface can be used to read saved logs in server. It allows reading sent logs and filtering them by upload user.

Current URL is: <http://gisai.dit.upm.es:3004>.

Choosing Logs option in the header will show the logs interface:

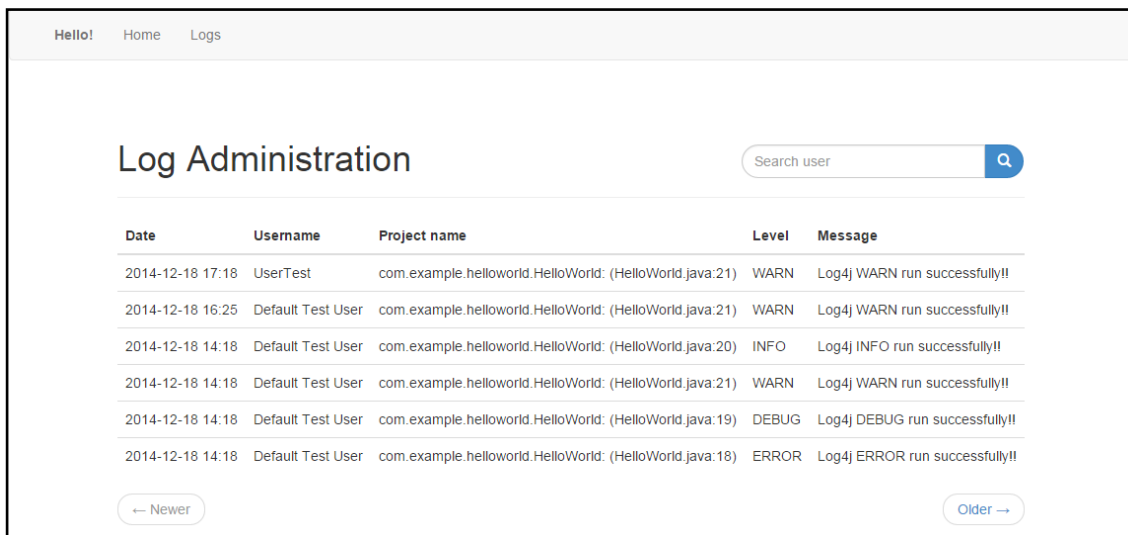


Figure 144: Flspace Logging log server

Filtering logs by user can be done searching in the box for a knowing user. Start typing in the box will display the found users which contain the typed string in their username. Clicking on the desired user and only logs of that user will show.

An example is included in the next figure.



Figure 145: Flspace Logging filtering by Username

2.21 Flspace Studio Tool

The users can download the latest version of Flspace Studio for the operating system of their preference from this web page: <http://37.131.251.110:8080/flspace-studio/>

Directory Listing For /0.13.0/ - Up To /		
Filename	Size	Last Modified
linux.gtk.x86.zip	166410.8 kb	Mon, 15 Dec 2014 11:29:46 GMT
linux.gtk.x86_64.zip	166560.8 kb	Mon, 15 Dec 2014 11:25:08 GMT
macosx.cocoa.x86.zip	166101.9 kb	Mon, 15 Dec 2014 11:32:05 GMT
macosx.cocoa.x86_64.zip	199456.0 kb	Mon, 15 Dec 2014 11:32:54 GMT
win32.win32.x86.zip	166453.7 kb	Mon, 15 Dec 2014 12:08:36 GMT
win32.win32.x86_64.zip	167129.9 kb	Mon, 15 Dec 2014 12:08:02 GMT

Apache Tomcat/7.0.53

Figure 146: Flspace Studio available versions

The next “splash screen” appears once the Flspace Studio is launched for first time:



Figure 147: Flspace Studio splash screen

Flspace Studio has its own customised About dialog:

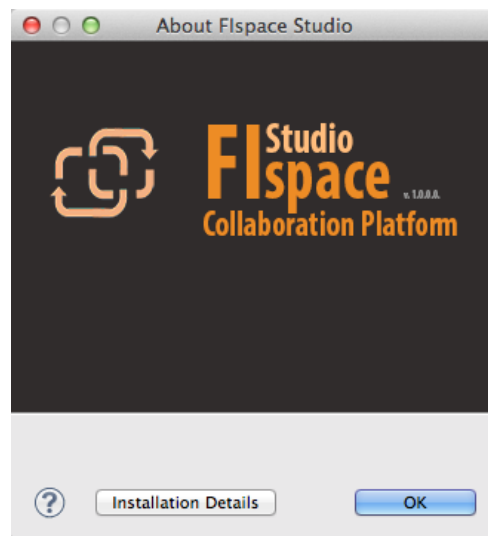


Figure 148: Flspace About Dialog

A custom intro or welcome page is provided with some web links pointing to Flspace documentation and tutorials:

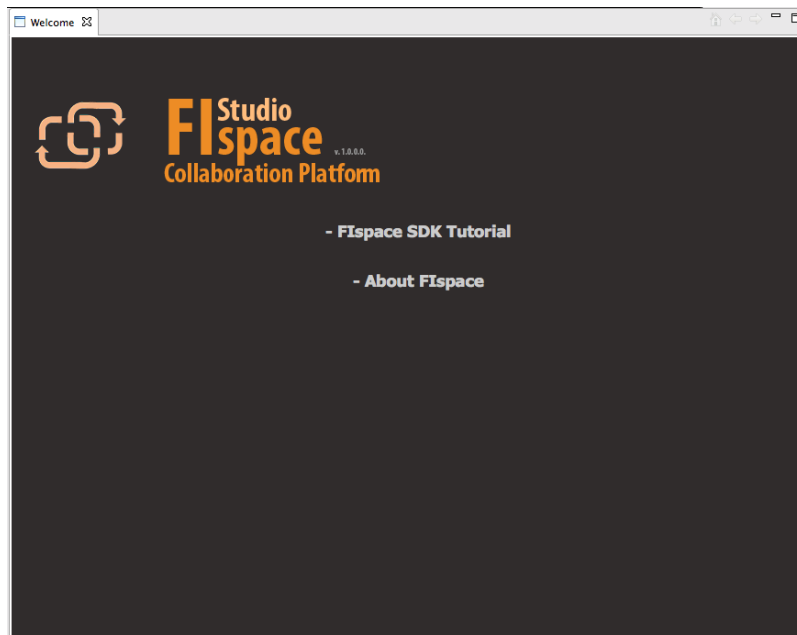


Figure 149: Flspace Intro page

Once the Flspace Studio is started the user experience includes all of the Flspace SDK functionalities under the same IDE platform.

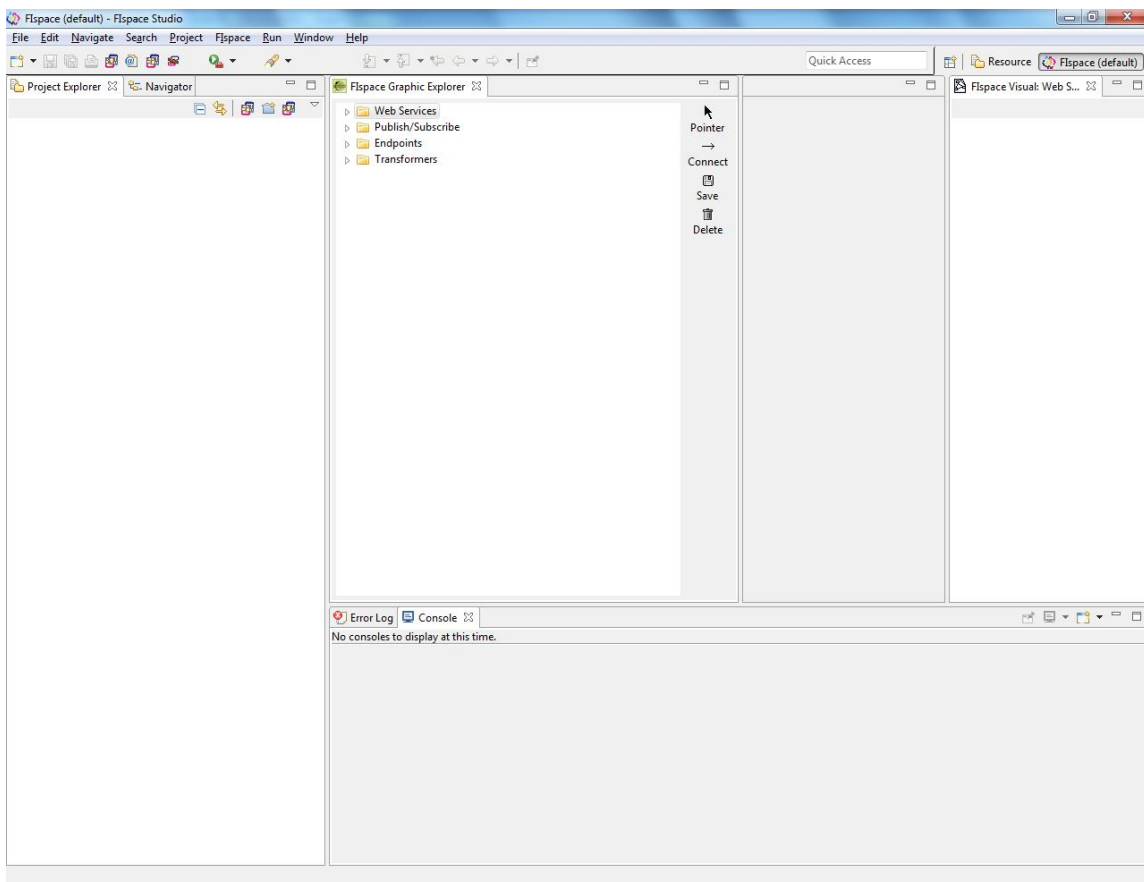


Figure 150: FlspaceStudio Tool

3 Glossary

The glossary provides the coherent terminological framework used in this document.

3.1 Terms and definitions

This section provides definitions of any terms that may be needed in order for the reader to understand the terminology used in the document. The author should define any definition/acronym or technical term used in the document that may be unfamiliar to the reader, and it is best to err on the side of too many rather than too few definitions. This also allows the author to frame a word within a specific context, which provides the reader with a common understanding of the author's definition.

Access control

Authorisation (or denegation) for performing a certain action (based on privileges management). The access control is carried out once the Identification and Authentication procedures have been performed.

Accounting

Process of gathering information about the usage of resources by subjects.

Acceptance and trust

Acceptability indicates the degree of approval of a technology by the users. It depends on whether the technology can satisfy the needs and expectations of its users and potential stakeholders. Within the framework of introducing new technologies, acceptability relates to social and individual aspects as well.

Application

Use of capabilities, including hardware, software and data, provided by an information system specific to the satisfaction of a set of user requirements in a given application domain.

Application Domain

Integrated set of problems, terms, information and tasks of a specific thematic domain that an application (e.g. an information system or a set of information systems) has to cope with.

Application Schema [ISO/FDIS 19109:2003]

Conceptual schema for data required by one or more applications.

Architecture (of a system) [ISO/IEC 10746-2:1996]

Set of rules to define the structure of a system and the interrelationships between its parts.

Architecture (of a system) [ISO/IEC 10746-2:1996]

Set of rules to define the structure of a system and the interrelationships between its parts.

Authentication

Process of verifying the identity of a certain subject. In other words authentication indicates whether a subject is who/what it seems to be.

Generally speaking, this proof can depend on a secret that can be, e.g. what somebody has (key, smart card, ...), what somebody knows (password, ...), what somebody is (biometrical data, ...)

Authorisation

Process of determining whether a subject is allowed to have the specified types of access to a particular resource. This is done by evaluating applicable access control information contained in a so called authorisation context. Usually, authorisation is carried out after the identification and authentication. Once a subject is identified and authenticated, it may be authorized (or not) to perform different types of access.

Availability

Availability refers to the degree to which a system, subsystem, or equipment is in a specified operable and committable state at the start of a mission, when the mission is called for at an unknown, i.e., a random time. So, availability is the proportion of time that a system is in operating condition.

Capability

Capabilities are a set of functionalities, through a combination of software and hardware, used to provide services and data. They can reside in a system or for example in a terminal itself as embedded capabilities or they can be available through the network services and infrastructure and others communication technologies as external capabilities.

Catalogue [derived from <http://www.opengeospatial.org/resources/?page=glossary>]

Collection of entries, each of which describes and points to a feature collection. Catalogues include indexed listings of feature collections, their contents, their coverages, and of meta-information. A catalogue registers the existence, location, and description of feature collections held by an Information Community. Catalogues provide the capability to add and delete entries. A minimum Catalogue will include the name for the feature collection and the locational handle that specifies where these data may be found. Each catalogue is unique to its Information Community.

Certificate Authority

A Trusted Third Party, responsible for ensuring the binding between the public keys and the personal data of their respective owners.

Component

Hardware component (device) or Software Component.

Conceptual model [ISO/FDIS 19109:2003(E); ISO 19101]

Model that defines concepts of a universe of discourse.

Conceptual schema [ISO/FDIS 19109:2003(E); ISO 19101]

Formal description of a conceptual model.

Coverage [ISO 19123]

Function from a spatial, temporal or spatiotemporal domain to an attribute range. A coverage associates a position within its domain to a record of values of defined data types. Thus, a coverage is a feature with multiple values for each attribute type, where each direct position within the geometric representation of the feature has a single value for each attribute type.

Data acquisition

Methods of data acquisition include methods to collect background data, digitally acquire data from sensors, and subjective data (such as data acquired from questionnaires). In addition, data in the form of manually or automatically transcribed data and reductions of collected data is also considered sensor acquired data (but with a manual sensor – the analyst).

Description Logics

Family of logic based knowledge representation languages that are a decidable subset of first order logic with well-defined semantics and inferencing (problem decision procedures). In Description Logics, a distinction is made between the terminological knowledge and the assertional knowledge. This distinction is useful for knowledge base modelling and engineering: for modelling it is just natural to distinguish between concepts and individuals; for engineering it helps by separating key inference problems.

Digital Certificate

A kind of digital document that contains structured information about the identity of its owner along with her/his public key, signed all together with a Certificate Authority's private key.

Digital Signature

The encrypted form of a message with the private key of the owner, indicating in a secure way the creator of the message, as well as the identity of a signed data.

Encryption

The act of modifying the contents of a message in an algorithmic and secure way, so that it can not be observed or altered in while in transit.

End-User

All users that are involved in an application domain and that use the applications, the services built by the system users according to the system and service Architecture.

Feature [derived from ISO 19101]

Abstraction of a real world phenomenon [ISO 19101] perceived in the context of an Application. In this general sense, a feature corresponds to an "object" in analysis and design models.

Framework [<http://www.opengeospatial.org/resources/?page=glossary>]

An information architecture that comprises, in terms of software design, a reusable software template, or skeleton, from which key enabling and supporting services can be selected, configured and integrated with application code.

Generic

A service is generic, if it is independent of the application domain. A service infrastructure is generic, if it is independent of the application domain and if it can adapt to different organisational structures at different sites, without programming (ideally).

Identification

The identification process allows relating a person/device with the service environment. The “electronic identity” is something like a credential or a “business card”, suitable to be verified throughout the authentication process.

Implementation [<http://www.opengeospatial.org/resources/?page=glossary>]

Software package that conforms to a standard or specification. A specific instance of a more generally defined system.

Info-structure Service

Service that is required to operate a system oriented service in the sense that it plays an indispensable role in the operation of an architecture or system oriented service.

Interface [ISO 19119:2005; <http://www.opengis.org/docs/02-112.pdf>]

Named set of operations that characterize the behaviour of an entity.

The aggregation of operations in an interface, and the definition of interface, shall be for the purpose of software reusability. The specification of an interface shall include a static portion that includes definition of the operations. The specification of an interface shall include a dynamic portion that includes any restrictions on the order of invoking the operations.

Interoperability [ISO 19119:2005 or OGC; <http://www.opengeospatial.org/resources/?page=glossary>]

Capability to communicate, execute programs, or transfer data among various functional units in a manner that require the user to have little or no knowledge of the unique characteristics of those units [ISO 2382-1]. (<http://www.opengeospatial.org/ogc/glossary/i>)

Loose coupling [W3C; <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#loosecoupling>]

Coupling is the dependency between interacting systems. This dependency can be decomposed into real dependency and artificial dependency: Real dependency is the set of features or services that a system consumes from other systems. The real dependency always exists and cannot be reduced. Artificial dependency is the set of factors that a system has to comply with in order to consume the features or services provided by other systems. Typical artificial dependency factors are language dependency, platform dependency, API dependency, etc. Artificial dependency always exists, but it or its cost can be reduced. Loose coupling describes the configuration in which artificial dependency has been reduced to the minimum.

Middleware [<http://www.opengeospatial.org/resources/?page=glossary>]

Software in a distributed computing environment that mediates between clients and servers.

Open Architecture [based on (Powell 1991)] [29]

Architecture whose specifications are published and made freely available to interested vendors and users with a view of widespread adoption of the architecture. An open ar-

chitecture makes use of existing standards where appropriate and possible and otherwise contributes to the evolution of relevant new standards.

Operation [ISO 19119:2005; <http://www.opengis.org/docs/02-112.pdf>]

Specification of a transformation or query that an object may be called to execute. An operation has a name and a list of parameters.

Performance indicators definition (PI)

PIs are quantitative or qualitative measurements, agreed on beforehand, expressed as a percentage, index, rate or other value, which is monitored at regular or irregular intervals and can be compared with one or more criteria.

Platform (Service)

Set of infrastructural means and rules that describe how to specify service interfaces and related information and how to invoke services in a distributed system.

Reference Model [ISO Archiving Standards; <http://ssdoo.gsfc.nasa.gov/nost/isoas/us04/defn.html>]

A reference model is a framework for understanding significant relationships among the entities of some environment, and for the development of consistent standards or specifications supporting that environment. A reference model is based on a small number of unifying concepts and may be used as a basis for education and explaining standards to a non-specialist.

Reliability

Reliability is the ability of a system or component to perform its required functions in routine circumstances, as well as hostile or unexpected circumstances, under stated conditions for a specified period of time.

Resource

Functions (possibly provided through services) or data objects.

Service [ISO 19119:2005; ISO/IEC TR 14252; <http://www.opengis.org/docs/02-112.pdf>]

Distinct part of the functionality that is provided by an entity through interfaces.

REST

Representational state transfer (REST) is an abstraction of the architecture of the [World Wide Web](#); more precisely, REST is an architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements, within a distributed [hypermedia](#) system. REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements.

Service [ISO 19119:2005; ISO/IEC TR 14252; <http://www.opengis.org/docs/02-112.pdf>]

Distinct part of the functionality that is provided by an entity through interfaces.

Session

Temporary association between a subject and a principal as a result of an authentication process initiated by the subject. Information about a session is stored in authentication session information.

SOAP

Simple Object Access protocol is a [protocol](#) specification for exchanging structured information in the implementation of [web services](#) in [computer networks](#). It uses [XML Information Set](#) for its message format, and relies on other [application layer](#) protocols, most notably [Hypertext Transfer Protocol](#) (HTTP) or [Simple Mail Transfer Protocol](#) (SMTP), for message negotiation and transmission.

Software Component [derived from component definition of <http://www.opengeospatial.org/resources/?page=glossary>]

Software program unit that performs one or more functions and that communicates and interoperates with other components through common interfaces.

Source System

Container of unstructured, semi-structured or structured data and/or a provider of functions in terms of services. The source systems are of very heterogeneous nature and contain information in a variety of types and formats.

Support Service

Service that facilitates the operation of an architecture or system oriented service, e.g. providing an added value by combining the usage of Info-Structure Services.

System [ISO/IEC 10746-2:1996]

Something of interest as a whole or as comprised of parts. Therefore a system may be referred to as an entity. A component of a system may itself be a system, in which case it may be called a sub-system.

Note: For modelling purposes, the concept of system is understood in its general, system theoretic sense. The term "system" can refer to an information processing system but can also be applied more generally.

System User

Provider of services that are used for an application domain as well as IT architects, system developers, integrators and administrators that conceive, develop, deploy and run applications for an application domain.

Terminal

Terminals are a mobile device that is capable of running mobile services and/or mobile applications.

Use case

A common definition of use cases is the one described by Jacobson (Jacobson et al (1995) [30]): “*When a user uses the system, she or he will perform a behaviourally related sequence of transactions in a dialogue with the system. We call such a special sequence a use case*”. In Other words, a use case is a textual presentation or a story about the usage of the system told from an end user’s perspective.

The use cases provide some tools for people, with different skills (e.g. software developers and non-technology oriented people), to communicate with each other. The use

cases are general descriptions of needs or situations that often are related to basic scenarios and that are independent of the technologies and implementations of the underlying system.

User

Human acting in the role of a system user or end user of the service and system.

WADL

The Web Application Description Language is a machine-readable [XML](#) description of [HTTP](#)-based [web](#) applications (typically [REST web services](#)) WADL models the resources provided by a service and the relationships between them. WADL is intended to simplify the reuse of web services that are based on the existing HTTP architecture of the Web. It is platform and language independent and aims to promote reuse of applications beyond the basic use in a web browser.

Web Service

Self-contained, self-describing, modular service that can be published, located, and invoked across the Web. A Web service performs functions, which can be anything from simple requests to complicated business processes. Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service.

W3C Web Service [W3C, <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>]

Software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

4 References

The following references are used as background documents for the preparation of this document. References are categorized standards (i.e. standards and specifications from the consortium working groups or alliances and specifications or drafts standardization bodies) and other documents, publications and technical or scientific books.

[1]	Flspace project. Flspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.1 <i>"Flspace Design and Release Plan"</i> , 2014.
[2]	Flspace project. Flspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.2 <i>"Flspace Technical Architecture and Specification"</i> , 2014
[3]	Flspace project. Flspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.3 <i>"Flspace Integrated Release V1"</i> , 2014
[4]	Flspace project. Flspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.4 <i>"Flspace Development Progress Report and V1 Updates"</i> , 2014
[5]	Flspace project. Flspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.5 <i>"Flspace Integrated Release V2"</i> , 2014.
[6]	Flspace project. Flspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.6 <i>"Flspace Development Progress Report and V2 Updates"</i> , 2014
[7]	Flspace project. Flspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.7 <i>"Flspace Integrated Release V3"</i> , 2014.
[8]	Flspace Business collaboration web site. http://www.fispace.eu/
[9]	Flspace Developer Documentation web site. http://dev.fispace.eu/doc/wiki/Home
[10]	Flspace Deliverables web site. http://www.fispace.eu/deliverable.html
[11]	Flspace Tutorial web site. http://www.fispace.eu/tutorials.html
[12]	Flspace App Developer Intro web site. http://dev.fispace.eu/doc/wiki/App%20Developer%20Intro
[13]	Flspace SDK Guide web site. http://dev.fispace.eu/doc/wiki/sdk
[14]	Flspace apps for newbies web site. https://bitbucket.org/fispace/apps/wiki/Flspace%20apps%20for%20new

	bies
[15]	FIWARE web site. http://www.fi-ppp.eu/projects/fi-ware/
[16]	FIWARE Catalogue of the Generic Enablers (GEs). http://catalogue.fi-ware.org/
[17]	FIWARE community web site. http://www.fi-ware.org/community/
[18]	FIWARE - Catalogue - Application Mashup - Wirecloud. http://catalogue.fi-ware.org/enablers/application-mashup-wirecloud
[19]	FIWARE - Catalogue - Store - Wstore. http://catalogue.fi-ware.org/enablers/store-wstore
[20]	Eclipse web site. https://www.eclipse.org/ , https://www.eclipse.org/downloads/
[21]	Maven web site. http://maven.apache.org/ , http://maven.apache.org/download.cgi
[22]	RabbitMQ web site. http://www.rabbitmq.com/
[23]	Express web site. http://expressjs.com/
[24]	ACSI EU-project web site. http://www.acsi-project.eu/
[25]	Proton - CEP recorded webinar and tutorial. http://edu.fi-ware.eu/course/view.php?id=58 Proton - CEP user and programmers guide. http://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/CEP_-_User_and_Programmer_Guide
[26]	OAuth 2.0. Open standard Authentication protocol specification. http://tools.ietf.org/html/rfc6749
[27]	Eclipse 4 RCP. http://www.vogella.com/tutorials/EclipseRCP/article.html
[28]	MongoDB. Open-source document database, NoSQL database. http://www.mongodb.org/
[29]	Powell, D. (Ed.) (1991). Delta-4: A Generic Architecture for Dependable Distributed Computing. Re-search Reports ESPRIT. Project 818/2252 Delta-4 Vol.1. ISBN 3-540-54985-4 Springer-Verlag 1991.
[30]	Jacobson, I., Bylund, S., Jonsson, P., and Ehneboom, S. (1995), "Modeling with Use Cases: Using contracts and use cases to build pluggable architectures". Journal of Object Oriented Programming, Vol. 8, No. 2, pp. 18-24.

