



# DAME Project

## Project Suite 1.0

### Data Mining Plugin Wizard User Manual

Doc. : DMPlugin\_user\_manual\_VONEURAL-MAN-NA-0004-Rel1.3

Issue: 1,3

Date: 14/04/10

<b>What</b>	<b>Who</b>	<b>When</b>
<b>Prepared by</b>	<b>M. Fiore, S. Cavuoti</b>	<b>14/04/10</b>
<b>Approved by</b>	<b>M. Brescia</b>	<b>14/04/10</b>
<b>Released by</b>	<b>G. Longo</b>	<b>14/04/10</b>



## Revision Matrix

Issue	Author	Date	Section/Paragraph Affected	Reason/Initiation/Documents/Remarks
0,1	M. Fiore	05/03/09	All	First draft
1	M. Brescia	14/09/09	All	Accepted & released
1,1	M.Fiore	24/03/10	All	General Revision
1,2	M. Brescia	25/03/10	all	check
1,3	S. Cavuoti	14/04/10	5	Par. 4 deleted Par. 5 revioned
1,3	M. Brescia	14/04/10	all	Changed par. 6 and final check



**INDEX**

1 Reference & Applicable Documents.....4

2 Definitions, Acronyms and Abbreviations.....5

3 Introduction.....6

4 Developing a DMPlugin .....6

    4.1 Developing a constructor.....6

    4.2 Overriding the run methods.....6

5 The DMPluginCreator Quickstart.....7

6 Complete example of the DMPlugin creation procedure.....15

**TABLE INDEX**

Tab. 1: Reference & Applicable Documents.....4

Tab. 2: Definitions, acronyms and abbreviations.....5

**FIGURE INDEX**

Fig. 1: DMPC 1 .....7

Fig. 2: DMPC 2.....8

Fig. 3: DMPC 3.....10

Fig. 4: DMPC 4.....11

Fig. 5: DMPC 5.....11

Fig. 6: DMPC 6.....12

Fig. 7: DMPC 7.....13

Fig. 8: DMPC 8.....14



## 1 Reference & Applicable Documents

REF. ID	TITLE	DATE	AUTHOR
REF. 1	PDD_VONeural2.0_VONEURAL-NA-PPD-0001-Rel2.0	15/10/08	Project Team
REF. 2	minute_meeting_10_04_12_VONEURAL-MIN-NA-0064	13/04/10	M. Brescia
REF. 3	IEEE Recommended Practice for Software Design Description	23/09/98	IEEE Board
REF. 4	FW_VONEURAL-SRS-NA-0002-Rel1.5	25/03/10	M. Fiore, O. Laurino
REF. 5	FW_VONEURAL-SDD-NA-0005-Rel2.0	14/04/10	M. Fiore, O. Laurino
REF. 6	DMPugins_DAME-TRE-NA-0016-Rel0.3	13/04/10	Brescia, Di Guido

Tab. 1: Reference & Applicable Documents



## 2 Definitions, Acronyms and Abbreviations

A & A	Meaning
FW	Framework
DAME	Data Mining Exploration
REST	Representational State Transfer
WN	Worker Node
SA	Stand Alone
DMPlugin	Data Mining Plugin
DMM	Data Mining Models
SAX	Simple API for XML
DOM	Document Object Model
DR	Driver
DBMS	Database Management System
PDD	Project Description Document
SDD	Software Design Description
SRS	Software Requirement Specification
VO-Neural	Virtual Observatory Neural

Tab. 2: Definitions, acronyms and abbreviations



### 3 Introduction

The Dame data mining suite offers a list of functionalities which can be chosen by a user to launch its experiments. These functionalities are implemented as plug-in. In this way the team can add functionalities to the suite in a simple way. After that it is sufficient to contact DAME working group to ask for the registration of the new functionality inside the DAME Suite.

Every DMPlugin needs to communicate with one or more DMM during its life cycle. To standardize the use of a DMPlugin we have created a set of constraints which a developer needs to respect. This allows a simpler developing of a DMPlugin.

**More information on the DMPlugin Wizard and the software package can be accessed at the web page:**

<http://voneural.na.infn.it/dmplugin.html>

### 4 Developing a DMPlugin

The development of a DMPlugin comes in two different steps: developing a constructor and overriding the run method.

#### 4.1 Developing a constructor

The signature of every DMPlugin constructor is the following:

- `public AbstractDMPlugin(IDrStorage dr_, String ip_, Integer experimentId_, String seid_, String userName_, Integer sessionId_) throws IOException` : the constructor used for the experiments

The first line of every constructor must be a `super(dr_,ip_,experimentId_,seid_,userName_, sessionId_)` statement, to call the `AbstractDMPlugin` constructor.

After that the developer must:

- create a new `Functionality` object, filling its params
- call the method `setInformations` with the previously created object
- Create an `HashMap` of running modes. For every running mode the developer needs to fill every info about input files, output files, partial output files, fields, tags and constraints. Refer to the previous section for informations about these structures.
- Call the method `setRunningModes` with the previously created hash map.

Additionally, the developer must create another constructor with the following signature: `public AbstractDMPlugin() throws IOException`:  
the body of this constructor is simply a call to `this(null,null,0,null,null,0)`

#### 4.2 Overriding the run methods

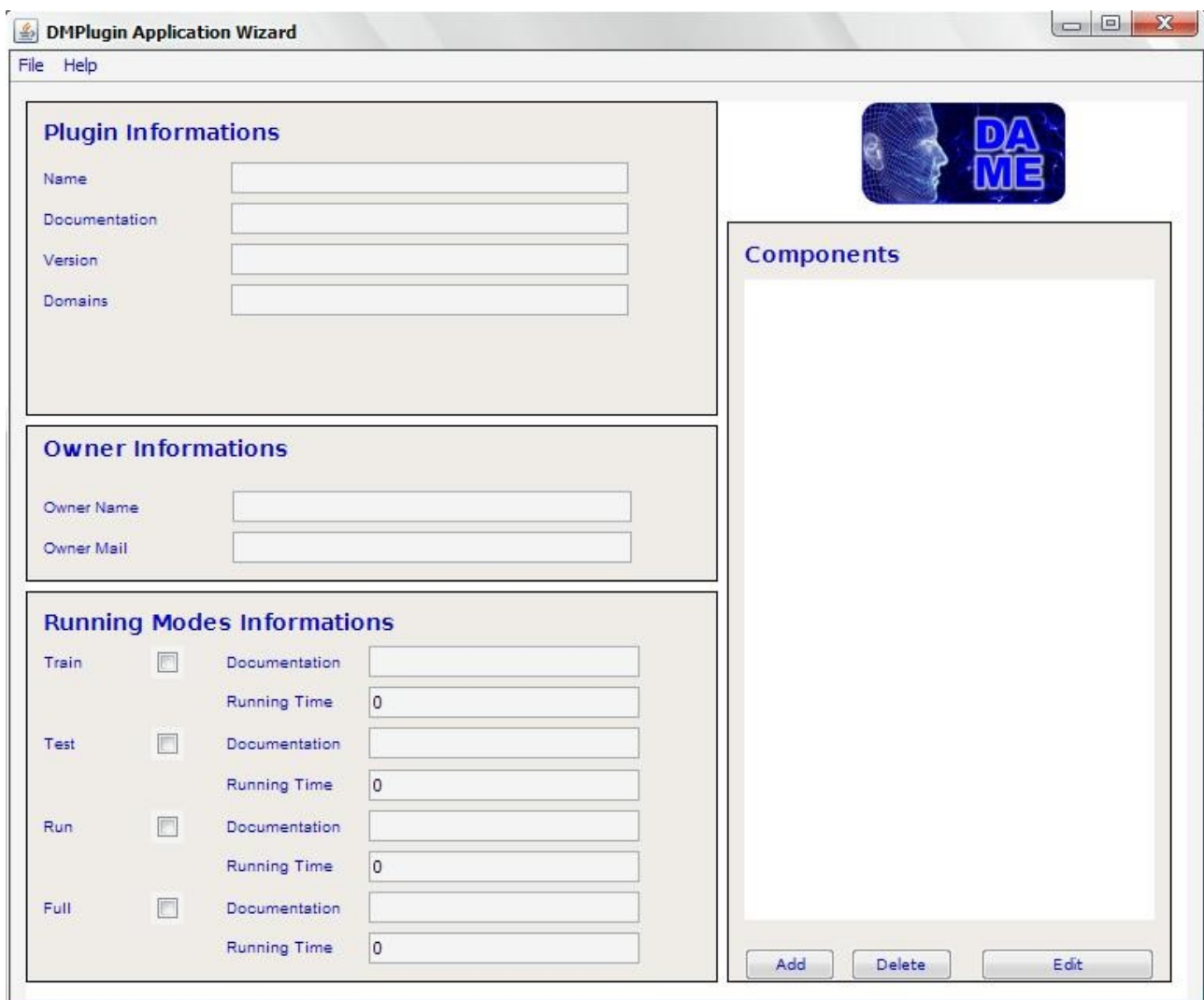
These methods represent the specific execution of a DMPlugin.

## 5 The DMPluginCreator Quickstart

The DMPluginCreator is a GUI which allows a developer to generate the constructor for its DMPlugin in a simple way.

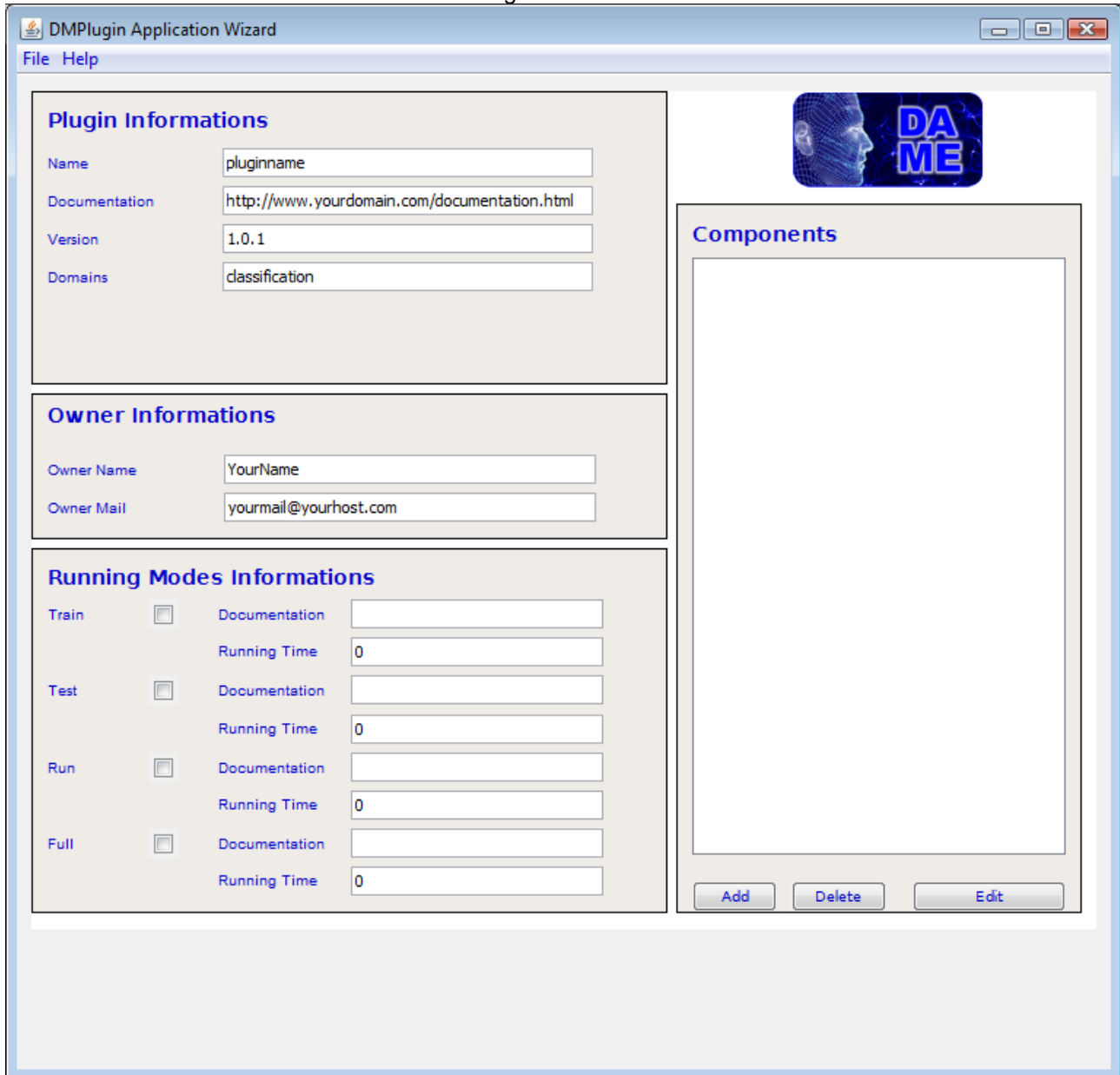
The following procedure shows how to create a simple DMPlugin. For additional informations refer to the DMPluginCreator Manual.

Fig. 1: DMPC 1



This is the starting window of the DMPluginCreator. Every field is blank at the moment.

Fig. 2: DMPC 2



Plugin Informations		
Name	:	pluginname
Documentation	:	http://www.yourdomain.com/documentation.html
Version	:	1.0.1
Domains	:	classification

Owner Informations	
Owner Name	YourName
Owner Mail	yourmail@yourhost.com

Running Modes Informations		
Train	<input type="checkbox"/>	Documentation: [ ] Running Time: 0
Test	<input type="checkbox"/>	Documentation: [ ] Running Time: 0
Run	<input type="checkbox"/>	Documentation: [ ] Running Time: 0
Full	<input type="checkbox"/>	Documentation: [ ] Running Time: 0

First of all the following are the basic information to be specified about the Plugin under configuration:

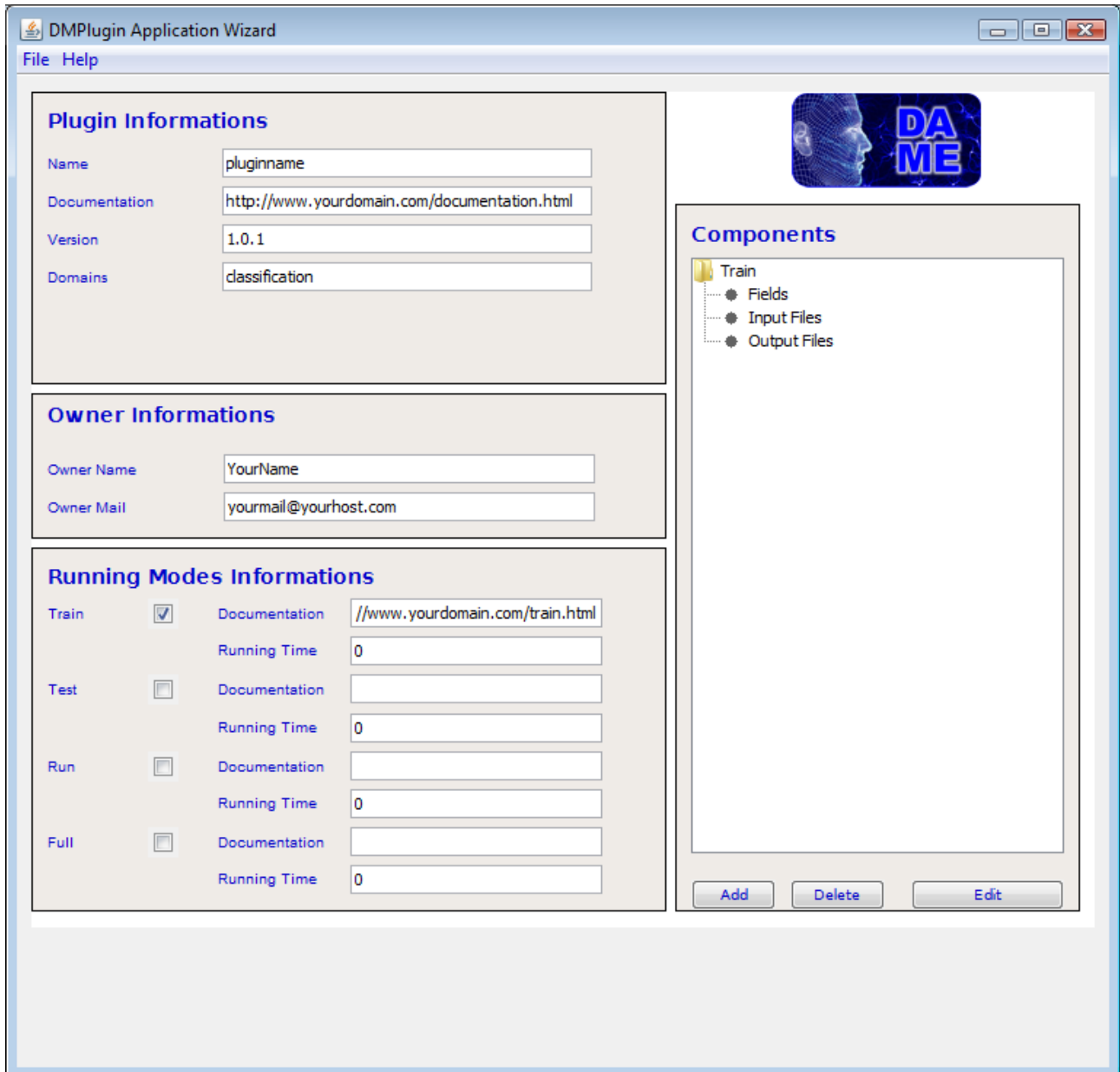
1. The name of the plugin (please note that this name is the same of the final java class produced)
2. The link to the documentation of your plugin
3. The version of the plugin
4. The domain of the plugin (classification, regression, clustering etc etc)
5. Your name and your mail in order to be contacted for further information.

When you have correctly filled in this information, you're ready to select which are the running modes (they sound like "use cases") you want to run with your plugin.

Running modes are in fact the use cases that the user want to be executed by your plugin, (in principle, to obtain a complete experiment, the plugin of a *supervised* neural network will have all the running modes

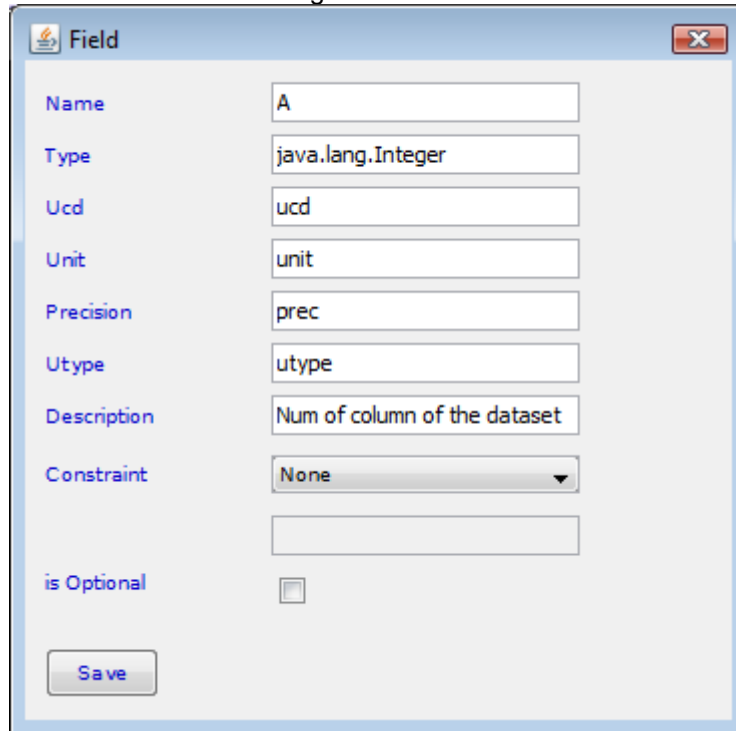


selected (train a new net, test a previously trained net, run your net on a new dataset or a full use case, where full automatically imply the sequenced execution of train, test and run cases), while an *unsupervised* model can't have a test use case, or a *statistic* plugin usually has just a run use case.



When you click on the checkbox of the running mode, you choose in the right panel appears a tree with three branches: Fields, Input Files, Output Files, that at this point are still empty. You must also add the link to the documentation of the specific use case (if any).

Fig. 3: DMPC 3



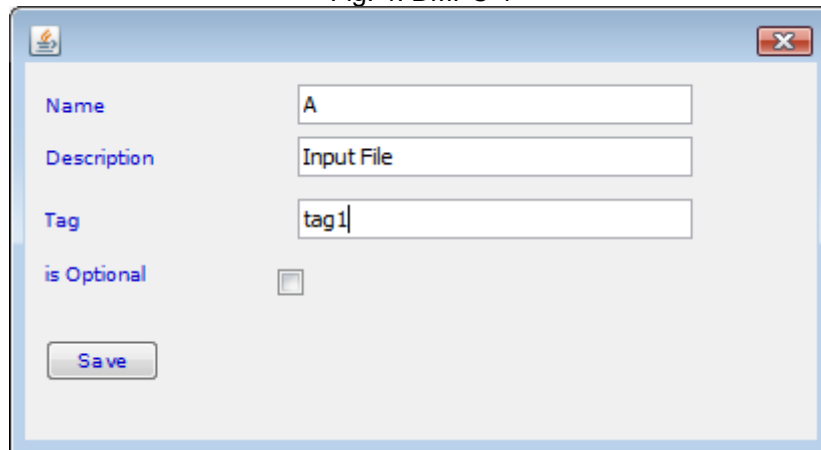
Name	A
Type	java.lang.Integer
Ucd	ucd
Unit	unit
Precision	prec
Utype	utype
Description	Num of column of the dataset
Constraint	None
is Optional	<input type="checkbox"/>

Save

In the Field dialog we can put information about the field. Pressing “Save” will create the new DMMFileParam. Now we are going to create a DMMFileParam and a DMMOutputFileParam by repeating the same steps we've done with the DMMFieldParam. We select the words “input file” or “output file” in the Train tree and we click on the add button. After that we can insert information about these new items using the new dialog and we can create the item by clicking on the save button.

We can use constraint if we need that a field can be just in a range, or can assume just same exact values, you can also declare that this field is optional and the user can leave it blank.

Fig. 4: DMPC 4

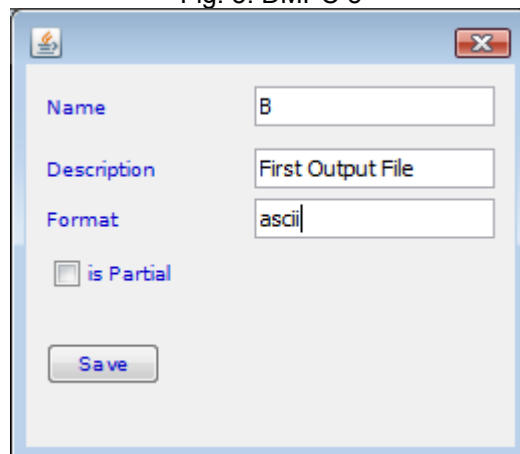


The screenshot shows a dialog box titled 'DMPC 4'. It contains the following fields and controls:

- Name:** A text input field containing the letter 'A'.
- Description:** A text input field containing the text 'Input File'.
- Tag:** A text input field containing the text 'tag1'.
- is Optional:** A checkbox that is currently unchecked.
- Save:** A button located at the bottom left of the dialog.

By adding an input file you can put the information about this file and declare if this file is optional or not.

Fig. 5: DMPC 5

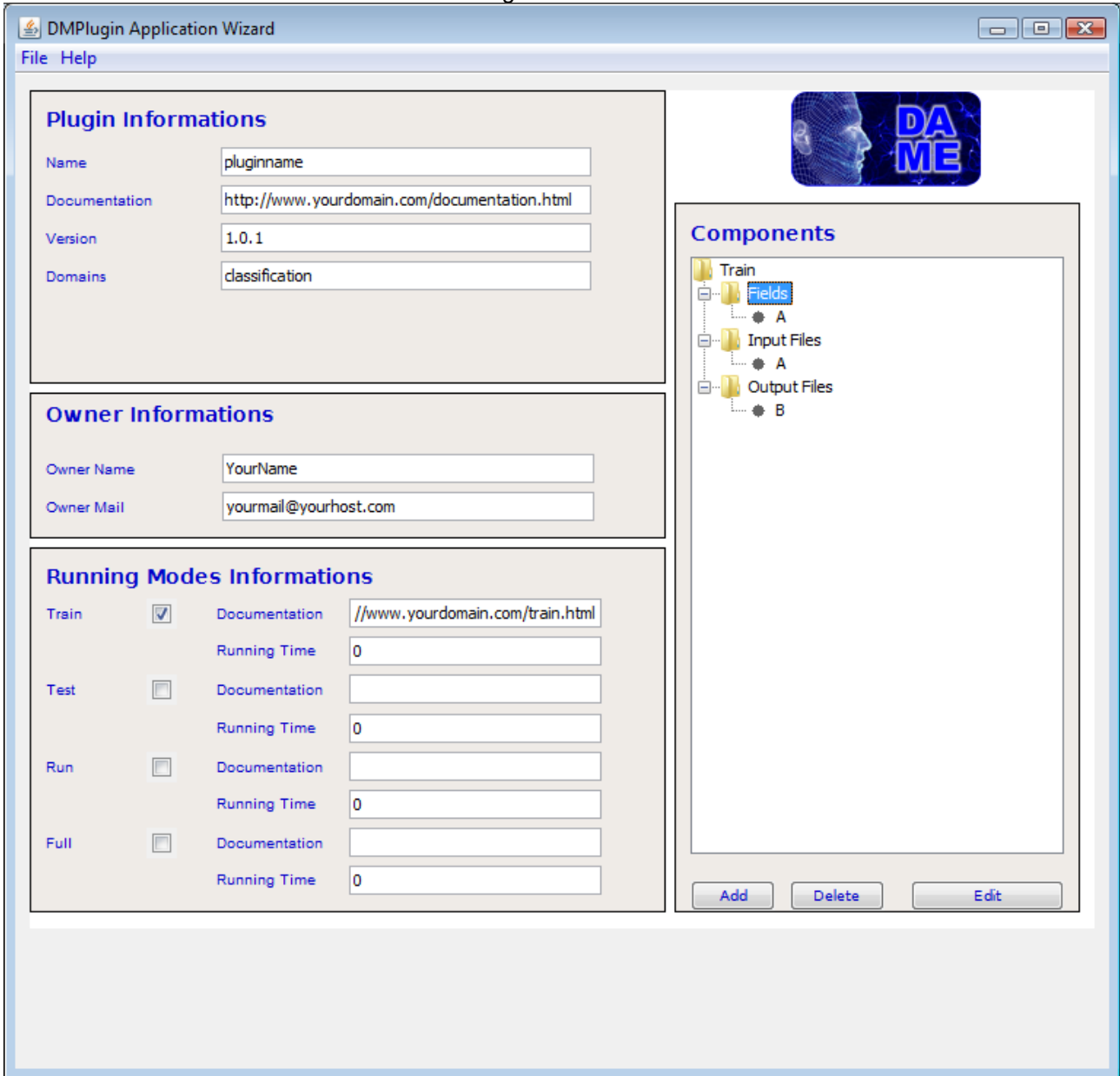


The screenshot shows a dialog box titled 'DMPC 5'. It contains the following fields and controls:

- Name:** A text input field containing the letter 'B'.
- Description:** A text input field containing the text 'First Output File'.
- Format:** A text input field containing the text 'ascii'.
- is Partial:** A checkbox that is currently unchecked.
- Save:** A button located at the bottom left of the dialog.

By adding an output file you must declare its desired format (ascii, CSV, fits, image etc...), depending also on the specific use case and data mining model selected for the current experiment (subject of the current plugin).

Fig. 6: DMPC 6



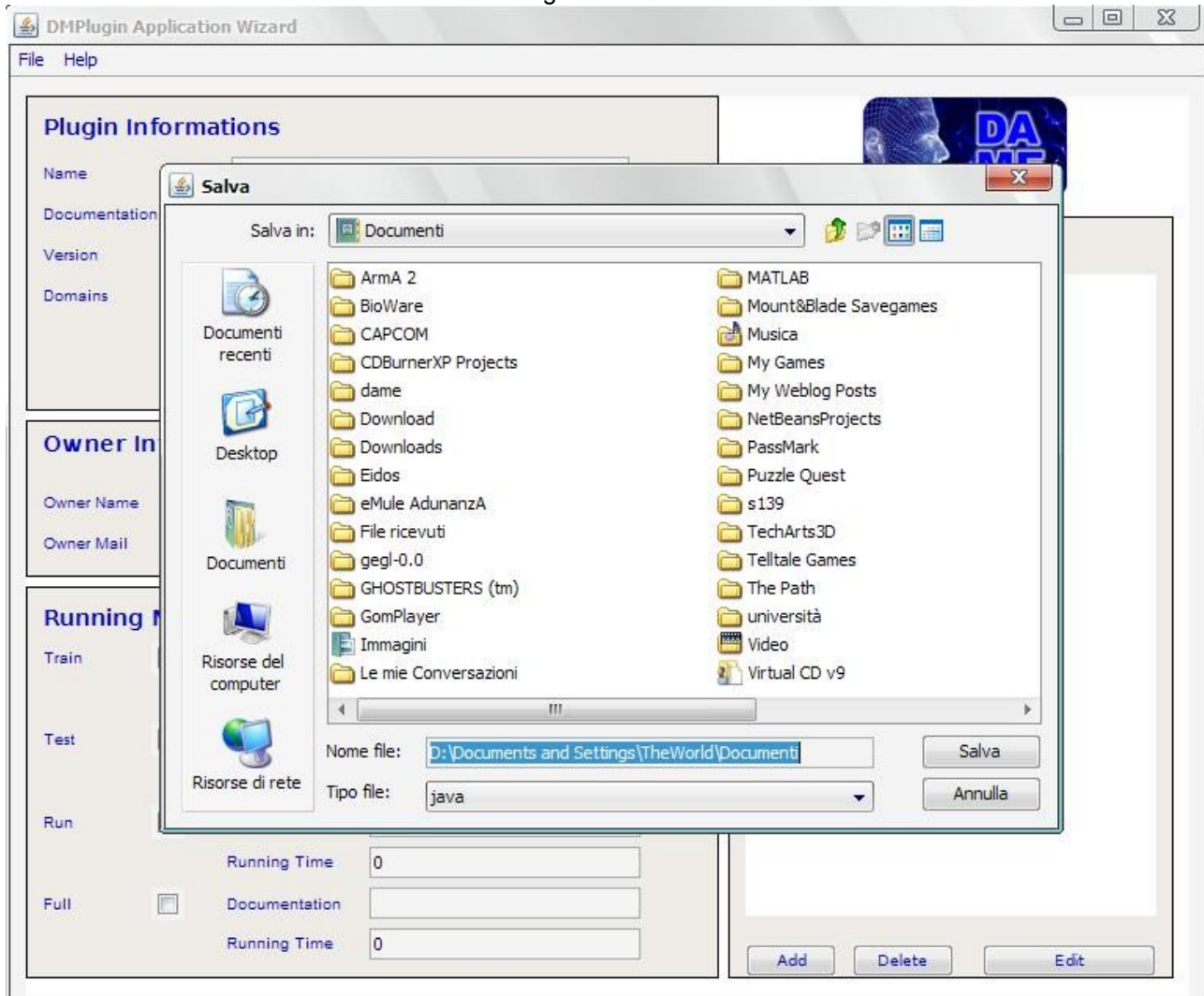
As you can see, the “Components” panel now shows the newly created components in the Train tree (train because this is the unique running mode previously selected).

Fig. 7: DMPC 7



Now we are gonna save this plugin by clicking on "File" in the menu bar and then on "Save". After that you can load your plugin by clicking on "File" in the menu bar and then on "Load". In this way you can keep working on your plugin.

Fig. 8: DMPC 8



After your plugin is completed you can generate the code of its constructor by clicking on “File” in the menu bar and then on “Generate Code”. **The file name must be the same of the plugin name**. This operations generates a java file. After that you can edit this file and insert your running modes code in the function trainRun, testRun, runRun and fullRun.



## **6 Complete example of the DMPlugin creation procedure**

What follows is a practical example of how the previous procedure should correctly be applied to create and build a DMPlugin from scratch.

TO BE DEFINED



\_\_oOo\_\_