SHORT COURSE: Handling of spatial data and connection of external algorithms to the DELFT-FEWS platform

Prepared by Susana Ochoa-Rodriguez

August 2013

1. FILES PROVIDED

You will be provided with a zip folder containing the following files/folders:

- **bin (folder):** contains the actual Delft-FEWS software
- **FEWS_SA (folder):** contains files corresponding to a specific application (or customisation) of the FEWS platform. In this case you have been provided with the files corresponding to the customisation we did in the tutorial on 15th February 2013.
- **jre-6u33-windows-i586.exe:** executable file to install the Java Runtime Environment
- **Files_Shortcourse (folder):** contains XML configuration files and datasets to be used during the tutorial.

2. SOFTWARE INSTALLATION AND FOLDER STRUCTURE (for standalone version)

Delft-FEWS is based on the Java software platform. Because of this, **Delft-FEWS is operating system independent and can be deployed without installer and registering code libraries**. The user only has to keep in mind some simple rules. One of them is the **folder structure**. Normally, the folder structure of Delft-FEWS looks like this:

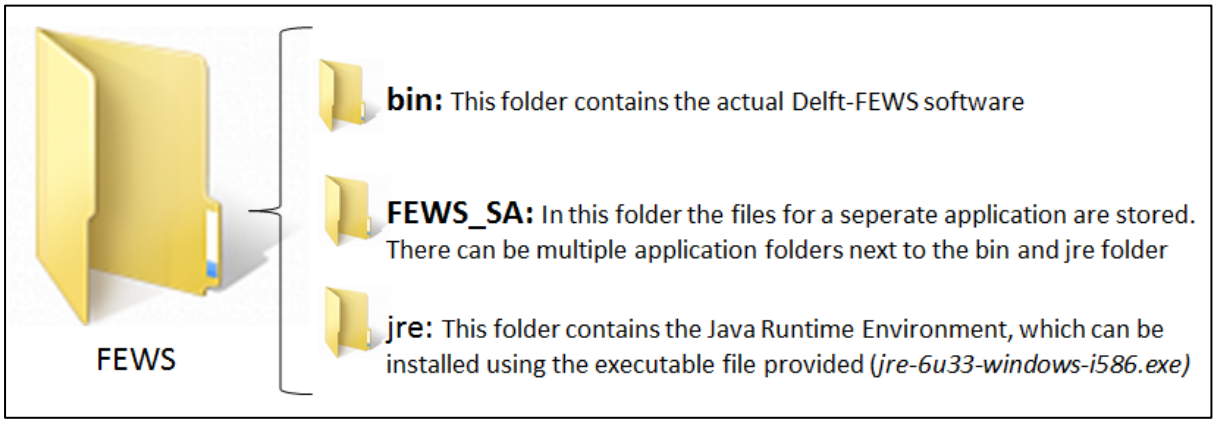

Figure 1: Required folder structure

The FEWS folder can be stored on any drive (except for a network drive).

To achieve the described folder structure and install the Java Runtime Environment, follow these steps:

a) Create a folder called 'FEWS' on any drive of your computer (you can give a different name to this folder if you wish)

b) Within the 'FEWS' folder, copy the following files/folders (which were provided): 'bin' folder 'FEWS_SA' folder, 'jre-6u33-windows-i586.exe' executable file

c) Within the 'FEWS' folder, create a new folder called 'jre'

d) Your 'FEWS' folder should look like this (but your 'jre' folder is empty at this point):

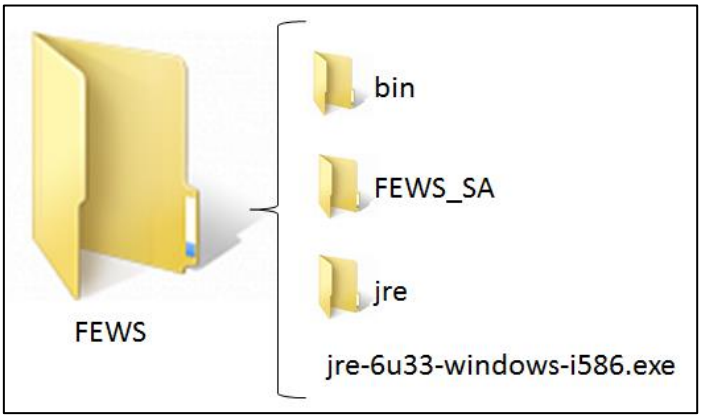

Figure 2: Installation files and folders

e) Run the executable file 'jre-6u33-windows-i586.exe' in order to install the Java Runtime Environment (JRE). MAC users must download and install JRE for MAC and everything else from here onwards should work fine and in the same way as for Windows users.

f) When asked (during the installation of the JRE), tick the 'Change destination folder' option and then click on 'Install':

Figure 3: Installation of Java Runtime Environment

g) Afterwards, select the folder where you wish to install the JRE. This must correspond to the 'jre' folder previously created

h) After installing the JRE and achieving a folder structure as that illustrated in Figure 1, you are ready to start customising your Delft-FEWS platform for your own pilot location!

i) ***Note:*** *besides the 'bin', 'FEWS_SA' and 'jre' folders, you can have any other folders and files you wish within the main 'FEWS' folder (as long as these have names different from the 3 required folders).*

3. RUNNING DELFT-FEWS

To run Delft-FEWS, go to the folder 'FEWS\bin' and double click on the executable file 'Fews_sa.exe', which has the following icon:

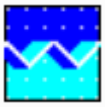

Figure 4: Fews_sa.exe icon

Afterwards the following screen should be displayed, which corresponds to the Delft-FEWS explorer (**already customised during the previous tutorial for the Cranbrook catchment (UK)**):

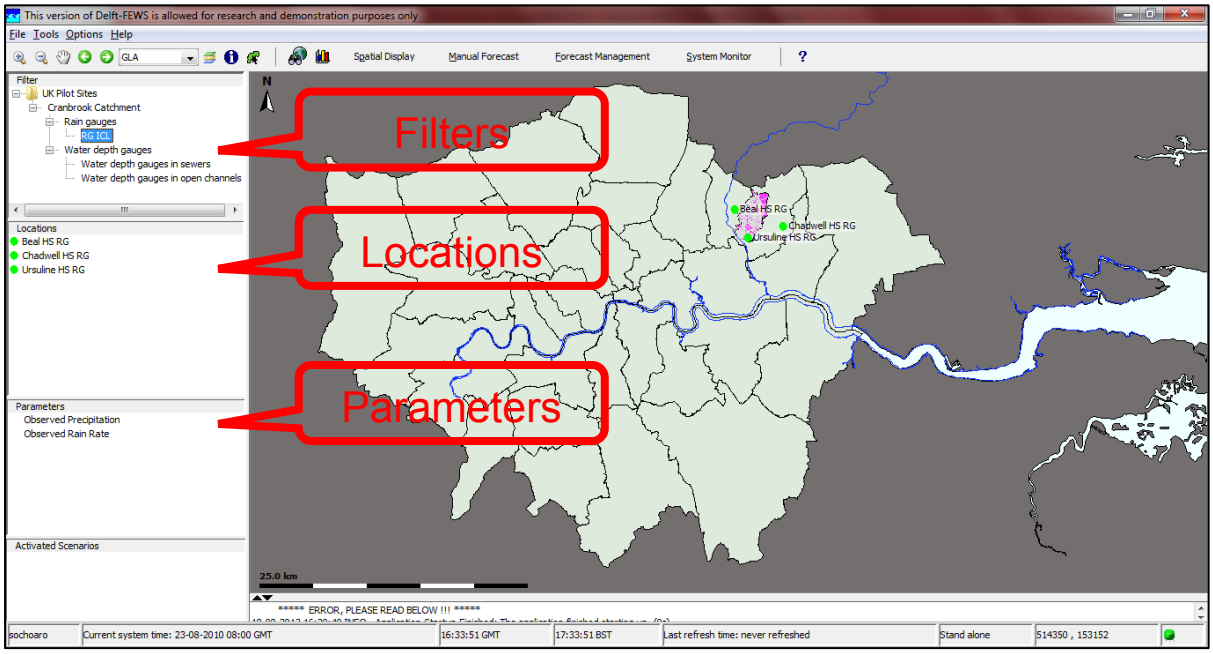

Figure 5: Delft-FEWS explorer window

4. GENERAL NOTES / RECOMMENDATIONS FOR CUSTOMISATION AND USE OF THE DELFT-FEWS PLATFORM (please read them carefully!)

- The Delft-FEWS system was developed using Java (thus the need of installing the JRE), but is **fully configurable through open XML** (Extensible Markup Language) formatted configuration files.
- The Schemas (.xsd) corresponding to each XML file can be found in:
  FEWS\Fews_customised\bin\ Delft_FEWS_schemas.jar
  This folder can be opened using a zip-software
- In XML files, **tags** are used to mark the start and end of **elements**, which are the logical units of information in an XML document. The tags begin with the less-than character (<) and end with the greater-than character (>) and the name of the element is contained within these two characters. An element consists of a **start tag**, followed by the **actual content of the element**, followed by an **end tag**. The end tags include a solidus (**/**) before the element's name. For example:

$$\underbrace{<\text{geoDatum}>}_{start\ tag}\ \underbrace{\text{WGS 1984}}_{\substack{actual\\content}}\ \underbrace{</\text{geoDatum}>}_{end\ tag}$$

- Comments in XML files can be included as follows: <!--COMMENTS-->
- Users are advised to install **Notepad++** and use it **for modifying the XML** configuration files (http://notepad-plus-plus.org/)
- The **files you will normally work on** when customising the Delft-FEWS platform are located in the following folder: FEWS\FEWS_SA\config
- Anytime you modify an XML file, it is advisable to **save it with a higher version** (in case you do something wrong, you can still keep the previous working version). By default, FEWS will read and use the latest version of a given file.
- Comprehensive documentation of DELFT-FEWS is available in Deltares' **public wiki**: http://publicwiki.deltares.nl/display/FEWSDOC/Home
  You can easily find answers to your questions here.

5. EXERCISE: IMPORT AND DISPLAY OF RADAR (GRID) DATA IN NIMROD FORMAT *(45 min)*

- **General description of activity:**

For this exercise we will use sample Nimrod radar data for the UK with 1 km spatial resolution and 5 min temporal resolution.

In order to import these data into Delft-FEWS, it is necessary to **create a new instance** of an existing module called *TimeSeriesImportRun*. Afterwards, the new instance must be included in a **workflow**, so that it can be executed, and it is necessary to design a **display** for the new data that has been imported.

- **Datasets and configuration files provided for this activity:**
  - The radar Nimrod data that need to be imported into Delft-FEWS can be found in: FEWS\Files_Shortcourse\01_Import_Display_Nimrod\Nimrod_Data
  - The configuration files (*.xml) required for creating the new instance, workflow and display can be found in: FEWS\Files_Shortcourse\01_Import_Display_Nimrod\XML_files

- **Instructions:**

The general steps for creating and using a new instance of an existing module, including creation of workflows and visualisation, are summarised in Figure 6. Following these steps and using the data and configuration files provided, you are invited to:

- Implement a new instance for importing Nimrod data
- Create a workflow for executing the new instance
- Create a display for the data that has been imported
- Test all of the above

Your main task is therefore to understand the connections between the different configuration files, to **place them in the right folders** and make sure that everything runs smoothly. Slides 50-57 of the PowerPoint presentation may help you in this process.

Note that:

- The configuration XML files that you have been given have higher versions than the ones currently used by the Delft-FEWS platform (i.e. the ones in the FEWS_SA folder). Therefore, once you copy the new XML file into the configuration folders, Delft-FEWS will by default use the new files (as these have a higher version than the original ones).

Remember that:

- Before running the newly created module instance, you must place the Nimrod data into the folder created for that purpose.
- The workflows that you create can be run from the 'Manual Forecast' window, which can be accessed from the buttons located in the toolbar of the Explorer window (see Figure 5).
- The Spatial Display window can be accessed from the buttons located in the toolbar of the Explorer window (see Figure 5). The Spatial Display window is used to display spatial (2D) data (e.g. radar data, flood extents), whereas the Explorer window can only be used to display time series at point locations.
- In order for data to be displayed in the Explorer or Spatial Display windows, it is necessary to change the display time to a time for which there are data available. For doing this, click F12 – this will display a menu from which you can choose the option called: "Set display time to last available for selection".

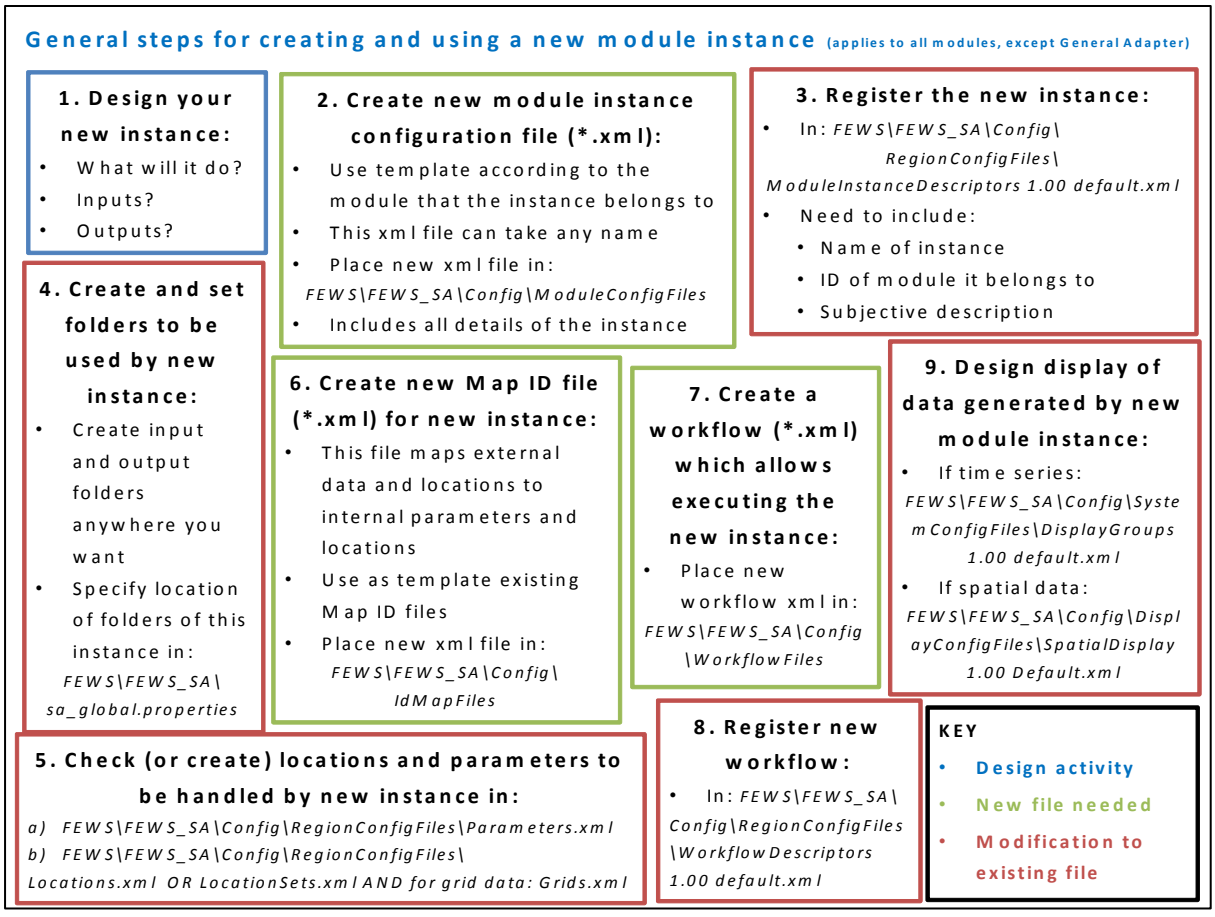

Figure 6: General steps for creating and using new instances of Delft-FEWS existing modules

6. EXERCISE: LINKAGE OF EXTERNAL ALGORITHMS TO DELFT-FEWS

- **General description of activity:**

For this exercise we will work with a simple algorithm which multiplies time series of rainfall depth by a value provided by the user.

This external algorithm will be linked to Delft-FEWS using a **new instance of the General Adapter module**. The algorithm takes the following 3 arguments, whose values can be set in the instance configuration (*.xml) file:

- Location of folder from which input data will be read
- Location of folder to which outputs will be written
- Multiplier to be applied to rainfall depth

After creating the new instance, it must be included in a workflow, so that it can be executed and tested.

- **Datasets and configuration files provided for this activity:**
  - The executable file corresponding to the external rainfall processing algorithm mentioned above can be found in: FEWS\Files_Shortcourse\02_External_Algorithm\Executable file\CSV_processing.exe
  - A sample input file to be used in the testing of this algorithm can be found in: FEWS\Files_Shortcourse\02_External_Algorithm\Input_test.csv
  - The configuration files (*.xml) required for this activity can be found in: FEWS\Files_Shortcourse\01_Import_Display_Nimrod\XML_files

- **Instructions:**

The steps required for creating and using a new instance of the General Adapter module are very similar to those illustrated in Figure 6. The differences between them are the following:

- In the case of the General Adapter **an executable file (*.exe) must be provided, which corresponds to the external algorithm to be linked to Delft-FEWS**. The executable file must be placed in a new folder, located in: FEWS\FEWS_SA\Modules
- The location of this new folder must be specified in: FEWS\FEWS_SA\sa_global.properties
- The external algorithm reads input data and writes output data from/to specific folders. There are three possibilities for specifying the location of the input and output folders:
  i. A permanent folder location is included in the algorithm (by the developer). In this case, the user cannot modify such location and must stick to it.
  ii. The algorithm reads a 'profile text file' containing the parameters/information required to run. This file may include the location of the input and output folders.
  iii. The algorithm takes as arguments the location of the input and output folders. In this case, the user simply needs to specify these locations in the module instance configuration file (*.xml). **We strongly recommend this option for future algorithms to be developed.**

- It is possible to generate the input data required by the external algorithm using an 'Export' instance of Delft-FEWS. Similarly, the outputs generated by the external algorithm can be imported into Delft-FEWS using an 'Import' instance. If this is the case, then a workflow including **'export –> external algorithm –> import'** routines must be created (see Figure 8).
- Since the external algorithm does not directly handle data from the Delft-FEWS database, it is not necessary to map internal and external parameters and locations (Steps 5 and 6 in Figure 6) at this stage. This is only necessary when data are being exported out of Delft-FEWS to be used by the external algorithm, or when the outputs of the algorithm are imported back into the Delft-FEWS database.
- It is not possible to visualise the output data of the external algorithm before these are imported into Delft-FEWS (Step 9 in Figure 6).

The general steps for creating and using a new instance of the General Adapter module, as well as the 'export –> external algorithm –> import' workflow concept described above are illustrated in Figure 7 and Figure 8.

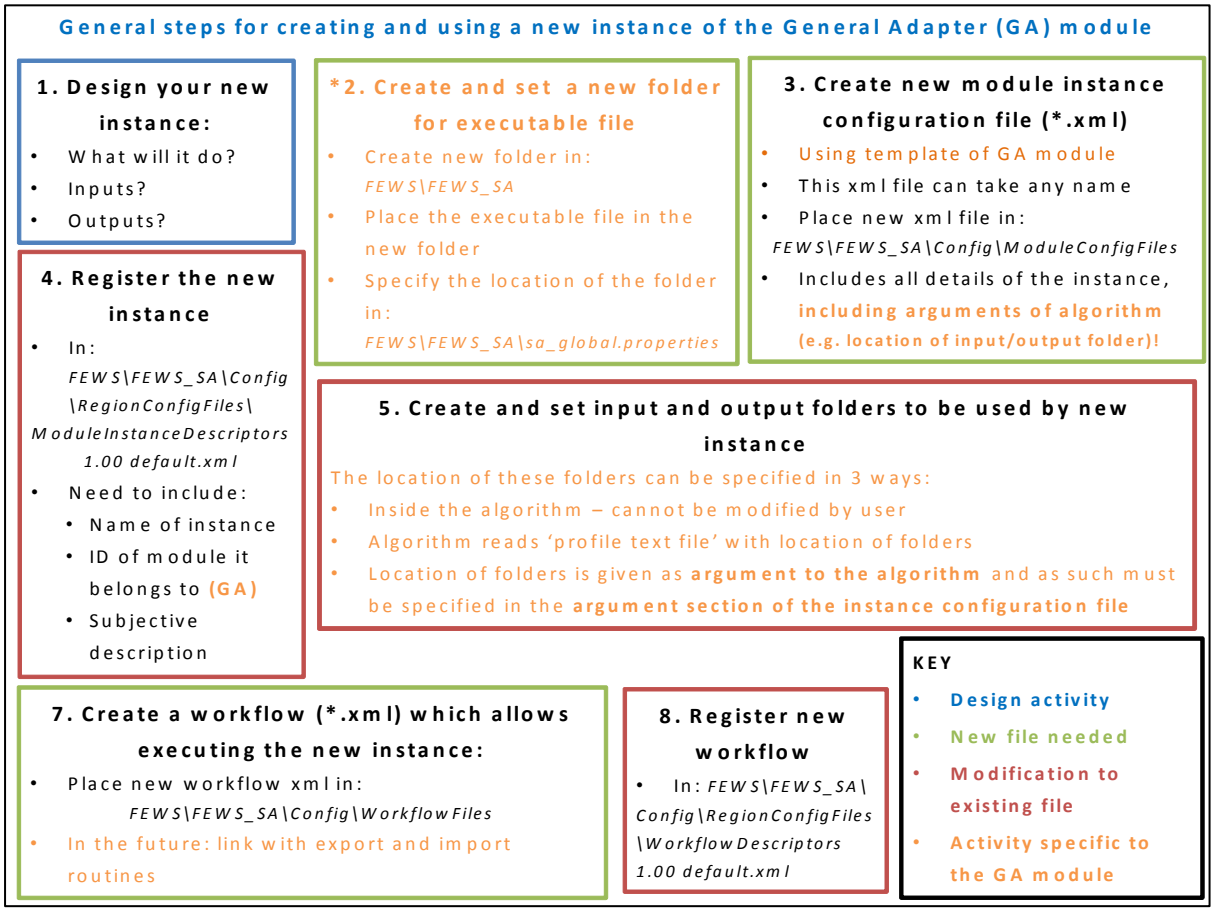

Figure 7: General steps for creating and using new instances of Delft-FEWS General Adapter module

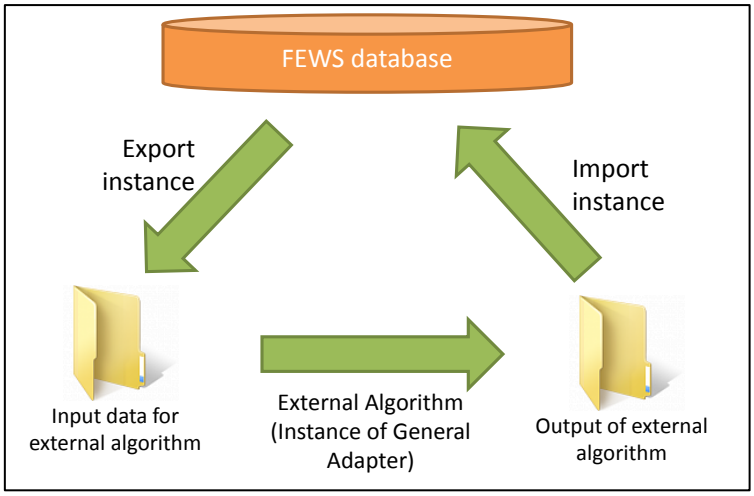

Figure 8: 'Export –> External algorithm –> Import' workflow for running external algorithms fully integrated to Delft-FEWS

Following the steps described above (Figure 7) and using the data and files provided, you are invited to:

- Implement a new instance of the General Adapter module in order to link the test rainfall processing algorithm (executable file) to Delft-FEWS. For Steps 2 and 5 you must follow the folder structure illustrated in Figure 9.
- Create a workflow for executing the new instance.
- Test all of the above using the test data provided.

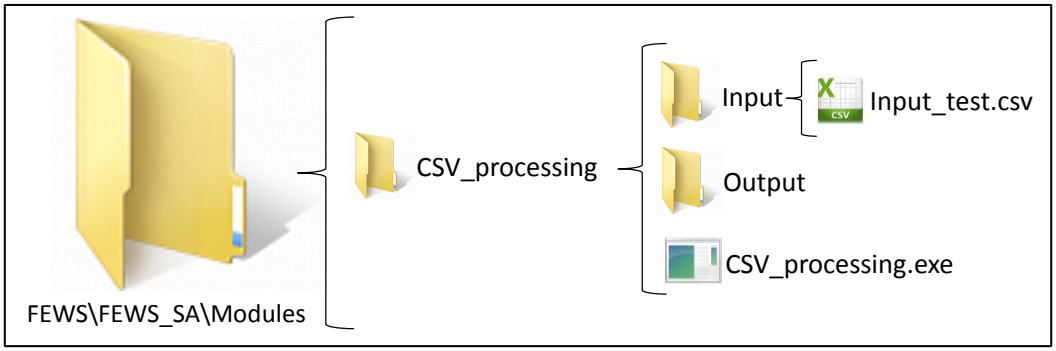

Figure 9: Folder structure to be used in exercise – linkage of external algorithms to Delft-FEWS

Note that:

- In this exercise the location of the input/output folders of the external algorithm constitute arguments of the algorithm; as such, their location must be defined in the 'arguments' section of the new instance configuration file (CSV_processing.xml). This has been already specified in the files provided, but you are suggested to check this and make sure you understand it.
- Due to time constraints, during this tutorial we will only test the external algorithm alone, but we will not implement export and import routines to complete the 'cycle' (as illustrated in Figure 8). However, in the previous tutorial you already learned how to export CSV data and in future tutorials we will carry out complete exercises, which include all elements of the cycle.

- Since we will not import the outputs of the algorithm into Delft-FEWS, we cannot display the data yet.

Remember that:

- The new workflow that you have created can be run from the 'Manual Forecast' window, which can be accessed from the buttons located in the toolbar of the Explorer window (see Figure 5).
- You can test if the external algorithm has run successfully by looking at the 'output' folder you created and defined before (see Figure 9). If the algorithm was successfully implemented and run, a new output file ('Output.csv') must have been generated, which contains the updated rainfall values, after multiplying the original values (in 'Input_test.csv') by a factor specified in the instance configuration file ('CSV_processing.xml').