Keeping Emulation Environments Portable

FP7-ICT-231954

System User Guide
Emulation Framework
Release 1.1.0 (December 2011)

Release date	21 December 2011
Release version	1.1.0
Author(s)	Bram Lohman (Tessella) Jeffrey van der Hoeven (National Library of the Netherlands) Edo Noordermeer (Tessella)
Organisation	KEEP project

Executive Summary

This is the System User Guide (SUG) for the KEEP Emulation Framework (EF). The EF is software developed by the international KEEP project, co-funded by the European Unions 7th Framework Programme.

The EF offers a convenient way to render digital files and programmes in their native computer environment. It offers users the potential to view these files in their intended 'look and feel', independent from current state of the art computer systems. The spectrum of potential computer platforms and applications that can be supported is practically unlimited. This release of the EF supports emulation of the x86, Commodore, Amiga and Amstrad CPC computer platform. Emulation is done by using existing emulators which are carefully selected on their capability to mimic the functionality of these platforms. The following illustration shows how this works in three steps.
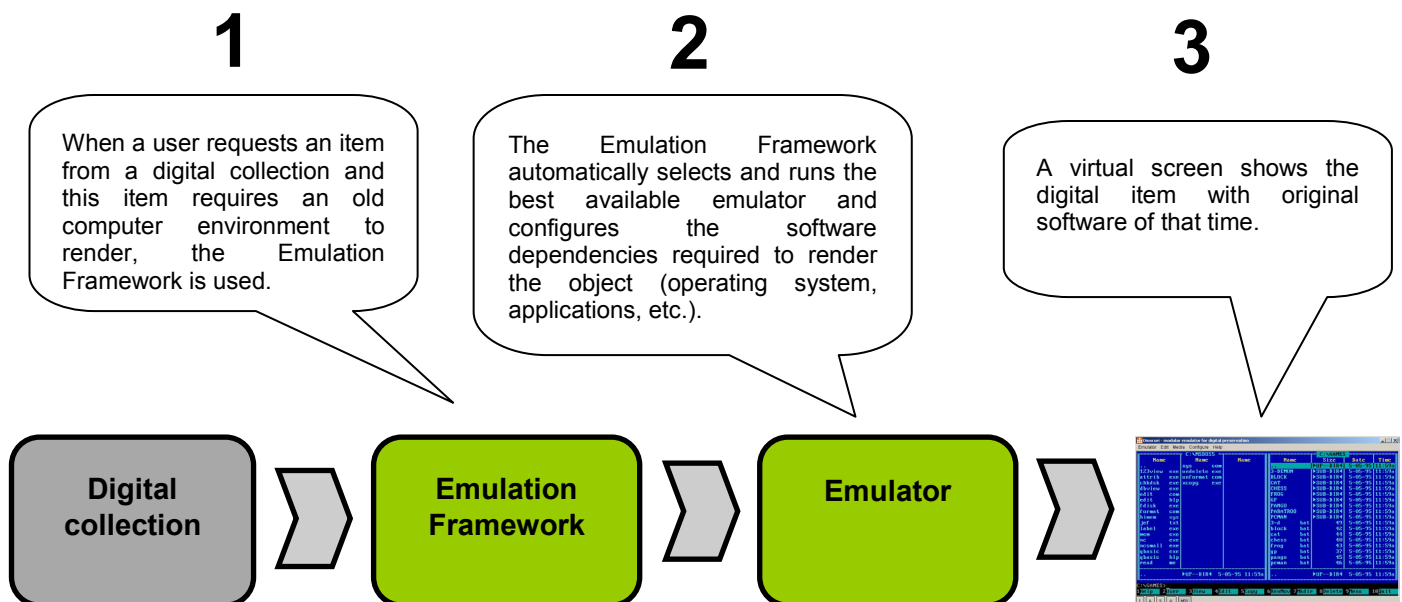
This guide contains information for end-users and administrators.

End-users such as library visitors and librarians are instructed on the use of the Emulation Framework. Detailed examples of how to use the supplied GUI are included which cover how to select a digital object, characterise it, determine the possible ways for rendering the object, as well as setting up, configuring and running the emulation environment containing the digital object.

Administration of the system is also explained. Explicit details on how to install and uninstall the Emulation Framework are given. The configuration of the Emulation Framework using either the Graphical User Interface (GUI) for administrators or the configuration file, as well as management of the embedded database using a database interface are described. The former includes a description of all the configuration settings, while the latter includes details on the database tables and how to access them.

Table of Contents

1 Introduction

This System User Guide (SUG) describes how to install and use the Emulation Framework (EF) developed by the KEEP project[1]. The document offers support to **end-users** and **administrators**.
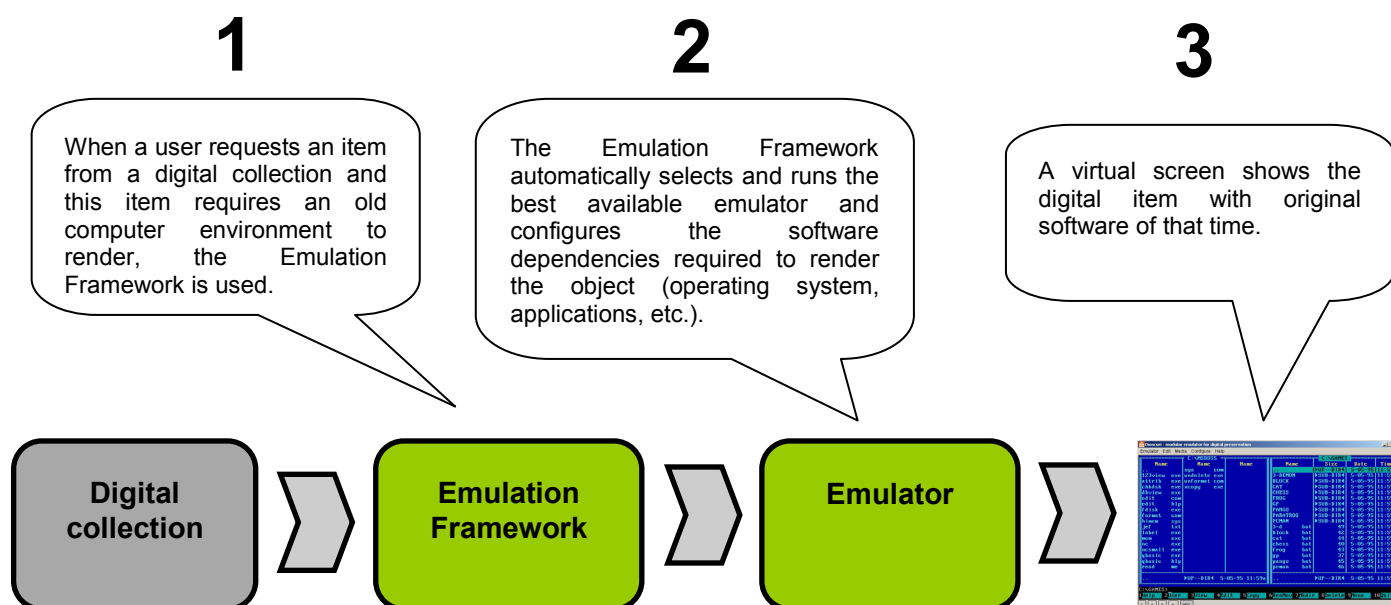
If you are a library visitor, librarian, archivist or other kind of user with interest in accessing old files and programmes with this tool, you are recommended to read the section **End-user guide**. This guide will explain how to use the EF and will give you insight into all features available.

If you are the administrator that would like to install or change the EF in a computer environment, please follow the section **Administrator guide**.

If you are a home user that would like to install and test the EF on a standalone basis, please make sure you read both guides.

1.1 What is the Emulation Framework?

The Emulation Framework allows you to render digital files and computer programmes in their native environment. This offers you the potential to view these files in their intended 'look and feel', independent from current state of the art computer systems. The spectrum of potential computer platforms and applications that can be supported is practically unlimited. This release of the EF supports emulation of the x86, Commodore, Amiga and Amstrad CPC computer platform. Emulation is done by using existing (open source) emulators which are carefully selected on their capability to mimic the functionality of these platforms. The following illustration shows how this works in three steps.

1

When a user requests an item from a digital collection and this item requires an old computer environment to render, the Emulation Framework is used.

2

The Emulation Framework automatically selects and runs the best available emulator and configures the software dependencies required to render the object (operating system, applications, etc.).

3

A virtual screen shows the digital item with original software of that time.

Digital collection ⟩ **Emulation Framework** ⟩ **Emulator** ⟩

---

[1] KEEP project website, available at: http://www.keep-project.eu

1.2 About the KEEP project

KEEP (Keeping Emulation Environments Portable) is an international research project co-funded by the European Union 7th Framework Programme. It does research into an emulation-based preservation strategy and develops several tools to support that. The consortium consists of eight organisations representing a wide range of stakeholders in Europe: cultural heritage institutes, research institutes, commercial ICT partners and the gaming industry. The project has a duration of three years and ends in February 2012.

More information can be found on the KEEP website: http://www.keep-project.eu

1.3 About the software in this release

The EF software consists of three parts:

1. Core Emulation Framework with Graphical User Interface (GUI)
2. Software Archive
3. Emulator Archive

The **Core EF** is the technical heart of the system, performing the automatic identification of file formats, selecting the required software and automatically configuring the emulation environment. It has a simple GUI to interact directly with the user. For selecting the software and emulator, the Core EF interacts with the Software Archive and the Emulator Archive.

The **Software Archive** is a separate web service that contains the software (applications and operating systems) available for the EF. The download package comes with the following open source operating systems:

- FreeDOS – an open source MS DOS look-a-like operating system
- Damn Small Linux – a small Linux kernel with limited functionality

The **Emulator Archive** is a separate web service that contains the emulators available for the EF. The download package comes with the following open source emulators:

1. Dioscuri – x86 Java-based emulator capable of running MS DOS and Linux.
2. QEMU – x86 capable of running MS Windows and Linux.
3. VICE – Commodore 64 emulator
4. UAE – Amiga emulator
5. Java CPC – Amstrad emulator
6. BeebEm – BBC Micro emulator
7. Thomson – Thomson emulator

The Core EF, Software Archive and Emulator Archive are developed by Tessella with support from the National Library of the Netherlands (Koninklijke Bibliotheek, KB). The Core EF GUI is developed by the National Library of the Netherlands.

This document was written for **EF Release version 1.1.0 (December 2011)**. For a previous version of the EF, please read the documentation for that particular release.

2 End-user guide

The Emulation Framework (EF) consists of three separate components:

- the **Core** including the Graphical User Interface (GUI);
- the **Emulator Archive** that provides the EF with emulators;
- the **Software Archive** that provides the EF with software required by the emulators.

As an end-user, you will only deal with the GUI and Core directly; your system administrator has installed and configured the Emulator Archive and Software Archive for you.

2.1 Starting the Emulation Framework

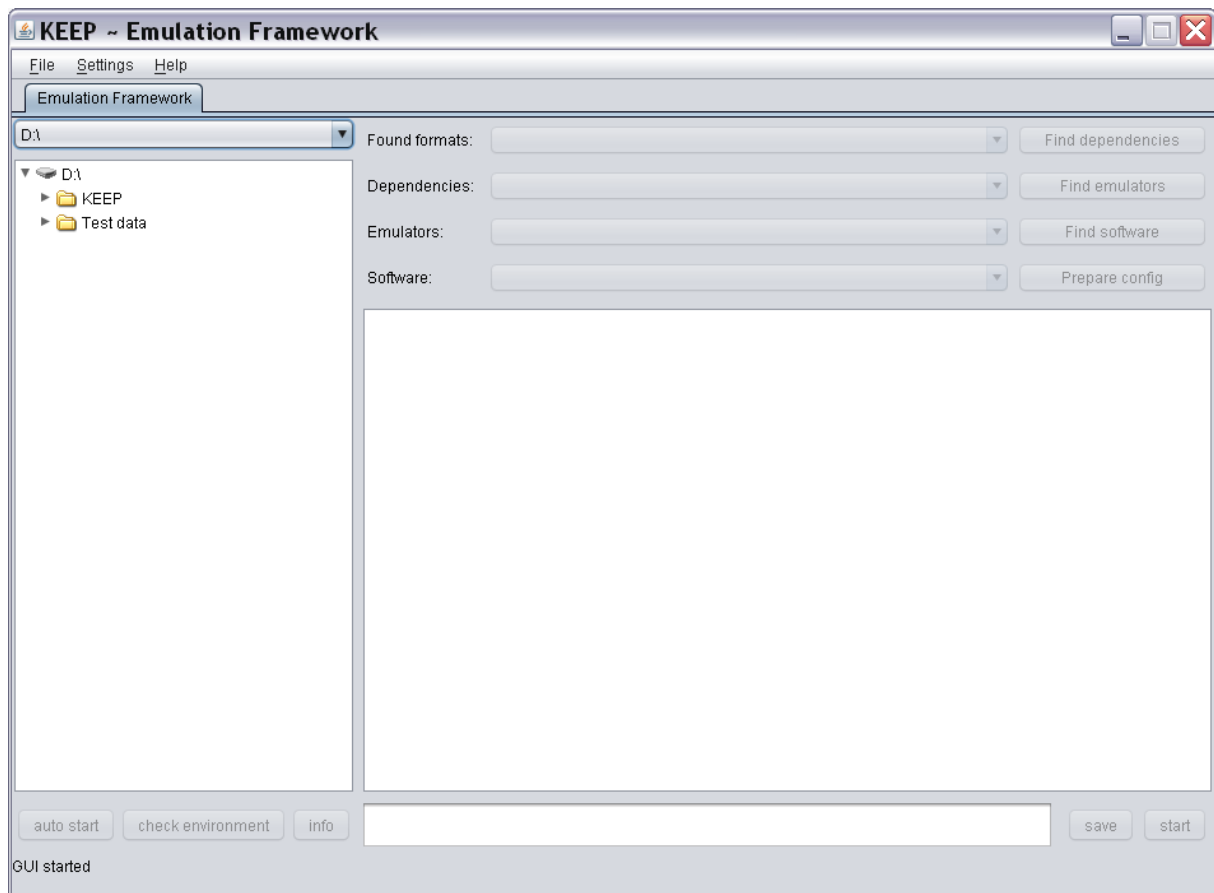To start the Emulation Framework and to launch the GUI, double-click on one of the batch scripts:

`runEF.bat` (for Windows platforms)

or

`runEF.sh` (for Linux platforms)

Note: your administrator should have installed and set up the system.

The following window should appear:

2.2 Functions of the Emulation Framework GUI

The EF window offers you the following functionality:

- Top menu bar:

   a. *File* – exit to close the EF application

   b. *Settings* – manage the language of the EF and location of the Software Archive and Emulator archive web services.

   c. *Help* – reference to online help documentation and credits.

- Tab *Emulation Framework* – select and render a digital file using emulation.

- *File tree* – overview of (local) file system. Here you can select a file to render.

- *Select boxes* – four steps in which you can manually select the appropriate dependencies on hardware and software for the selected file.

- *Configure window (currently blank)* – area in which you can manually configure the selected emulator.

- Buttons:

   a. *auto start* – to automatically start the emulation process without human intervention.

   b. *check environment* – to manually adjust the emulation process.

   c. *info* – show detailed technical information about the selected file.

   d. *save* – save changes to the configuration of the emulator.

   e. *start* – manually start the emulation process after you have selected the hardware and software dependencies for the selected file.

- *Status bar* - all important status information will be shown here.

Note: if you see more than one tab next to the tab *Emulation Framework*, you have administrator rights and are capable of managing the Software and Emulator archives. In this section, only tab *Emulation Framework* is described. The other options are explained in the section for administrators.

2.3 Choosing a file or application

The Emulation Framework (EF) can emulate any of the supported computer environments listed in section **1.3 About the software in this release**). To start an environment, the EF requires a file as input. This file can be an office document such as PDF, excel, Wordperfect, or an audio and video stream. It can also be a disk or tape image containing a computer application that can run on one of the supported computer environments. Examples are: a Commodore 64 disk image and an ISO CD-ROM file.

Click on the **file tree** on the left to select a file you would like to have rendered using emulation. By expanding the nodes of the tree you can browse the directories.

The following screen shows the GUI with a file selected:
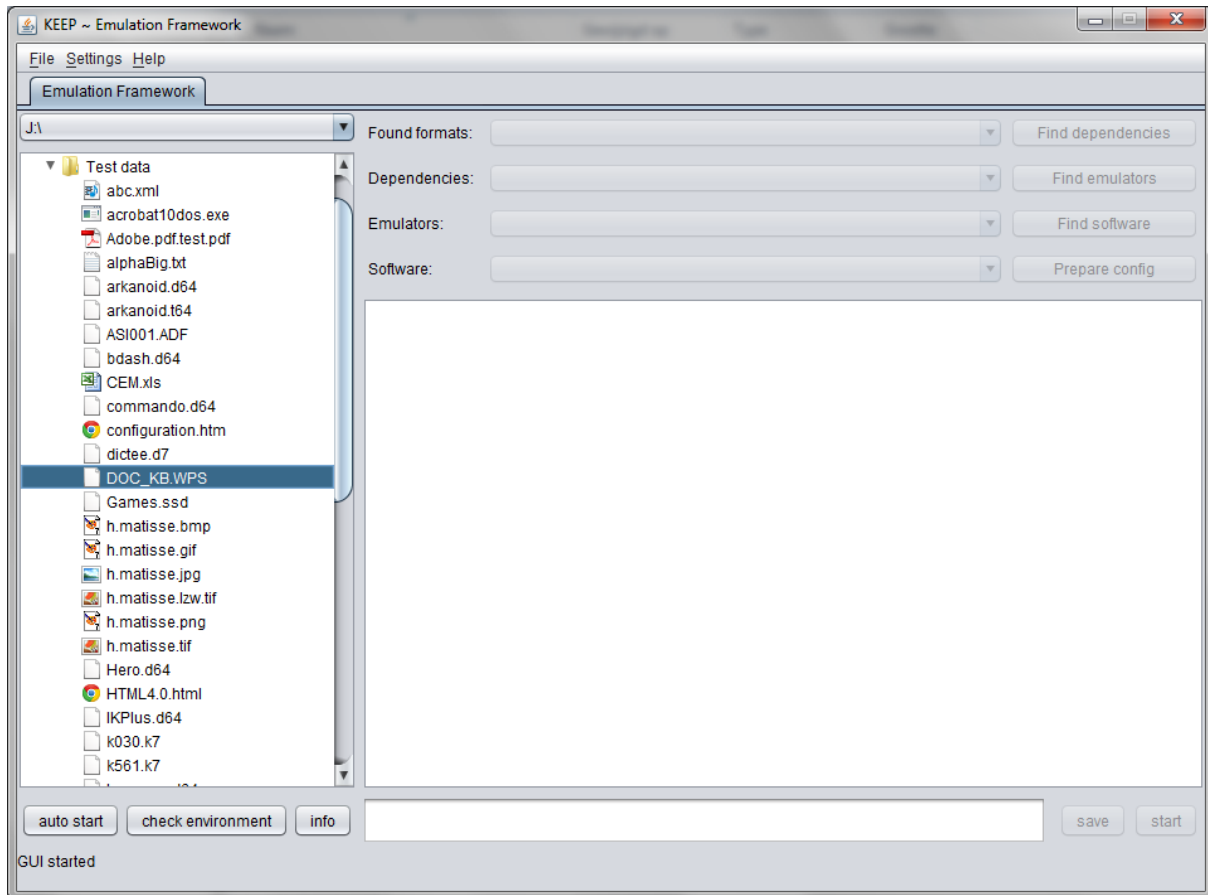
At the bottom of the screen three buttons are now enabled. At this point you can choose between the automatic mode by clicking the **auto start** button, and manual mode by clicking the **check environment** button. These two modes are explained in detail in the following sections.

2.4   Automatic mode – starting an emulator

The EF has an automatic mode which, when given a digital file or application, will automatically characterise it, and select an environment (consisting of an emulator and the required software) that is able to render it. The success of this automatic process depends on the correct identification of the file format and the available emulators and software.

To start the automatic mode for a given file, **double-click the file** with the mouse. Alternatively, once a file has been selected, the **auto start** button can be pressed. With either option, the EF will automatically try to find the required software and emulator for the selected digital file and if successful, automatically starts the emulation process. Depending on the size of the emulation software to download from the Software Archive, this process may take a few minutes. The result is shown in the following screen.
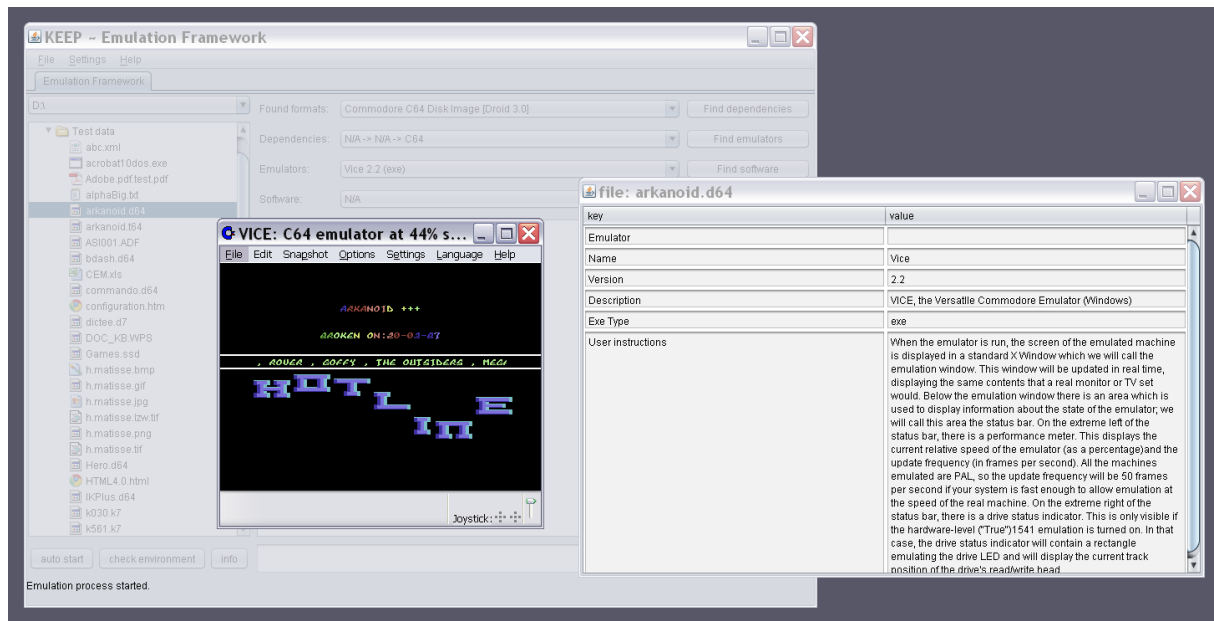
The emulator is running (in this example Commodore 64 emulator VICE) and a popup window is opened showing extra information about the emulator and loaded software.

When finished the emulator can be closed by the user and the emulation process ends automatically.

2.5 Manual mode – starting an emulator

The manual mode allows you to influence the way the emulated computer environment is constructed. To do so, you need to follow the following five steps:

1. Check environment
2. Find dependencies
3. Find emulators
4. Find software
5. Prepare configuration & start

In the following sections, each of these steps are explained in more detail.

2.5.1 Step 1 – Check environment

Once a file is selected, the **check environment** button will be enabled. Pressing this button starts the manual mode. The first step is to identify the file format. By clicking the **check environment** button the EF determines the type of format the digital file has. This forms the basis for creating an emulated computer environment.

After a few seconds, the top slot of the right pane contains the results of the identification process, as shown in the following figure:
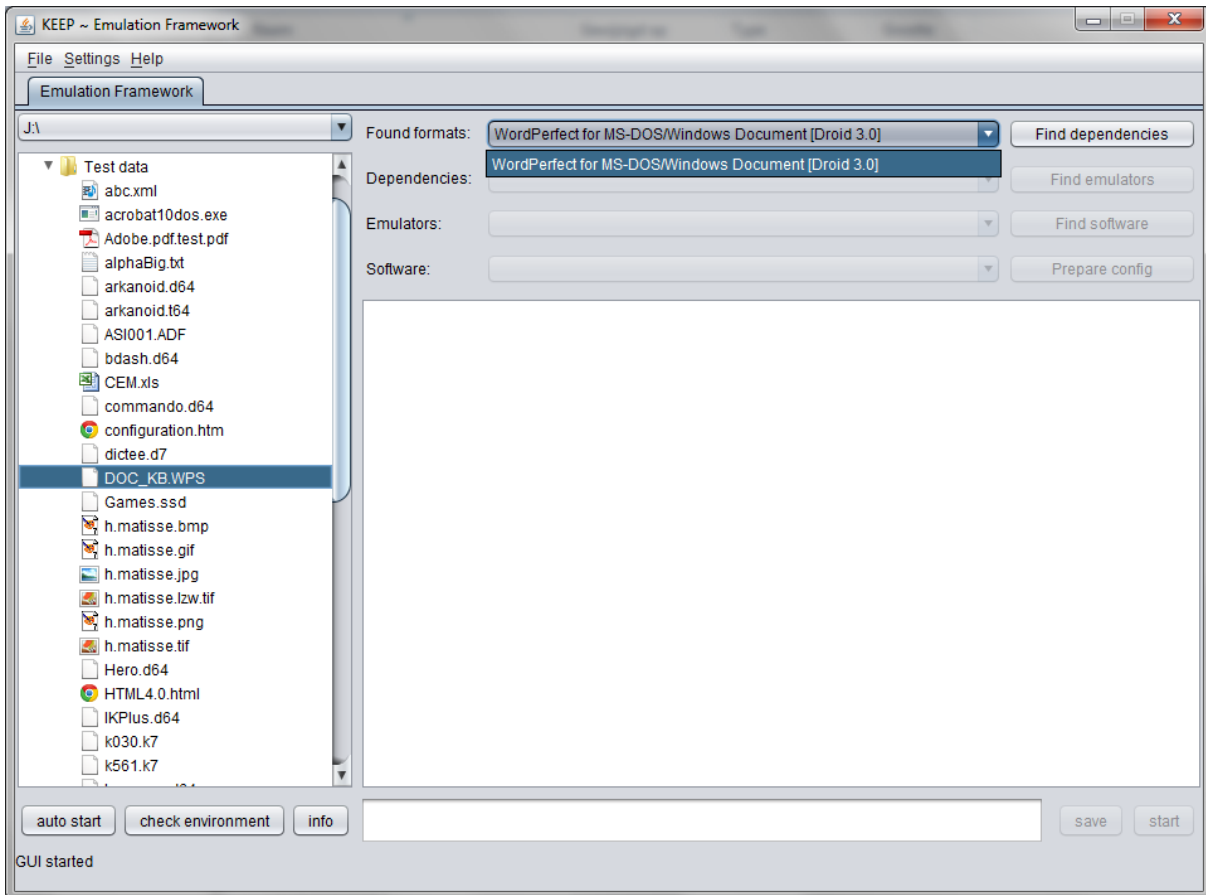
In this example, the EF has found a single result, and it is automatically selected. If no formats are found, the EF has not been able to successfully identify the file, and the process cannot continue from this point.

2.5.2   Step 2 – Find dependencies

Once a file format has been found, the **Find dependencies** button is enabled. A dependency is a computer environment consisting of an application, operating system and hardware platform. By pressing this button the EF contacts the Software Archive and requests a list of all possible combinations to render the file.

An example of the dependencies returned for the above file format is as follows:
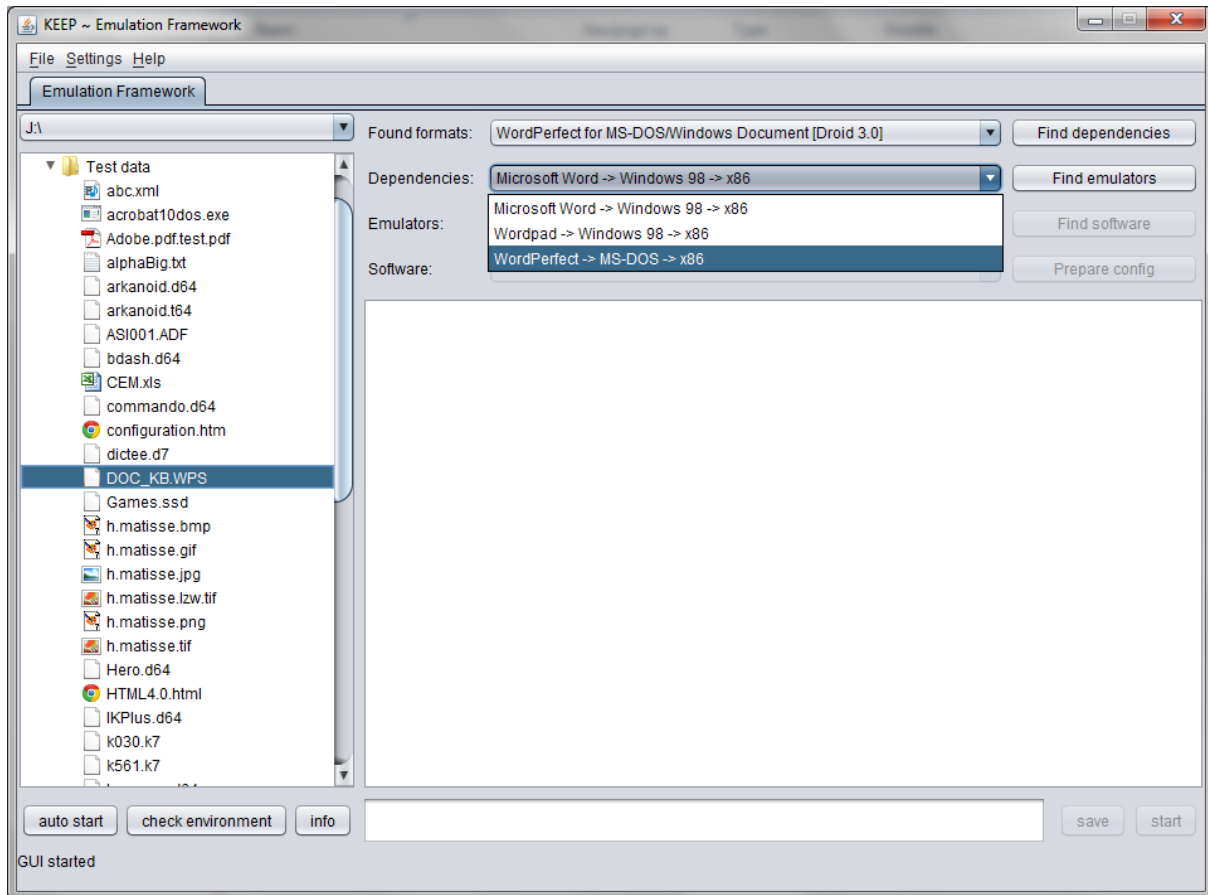
In this case, three pathways have been found, and the following environment is selected:

```
WordPerfect (application) -> MS-DOS (operating system) -> x86 (platform)
```

Note: if no dependencies can be found, a message will be displayed in the status bar at the bottom of the screen. Sometimes, the field for application is defined as N/A. In this case, no application is required to run the file (e.g. the file is a computer game depending only on operating system and hardware).

2.5.3   Step 3 – Find emulators

Once a dependency has been selected, the **Find emulators** button is enabled. By pressing this button the EF requests all available emulators for this dependency from the Emulator Archive.

Once the process is completed, the next drop-down box will be populated with these results. If there are no valid results, this will be displayed in the status bar at the bottom of the window.

A list of valid results is shown in the following view:
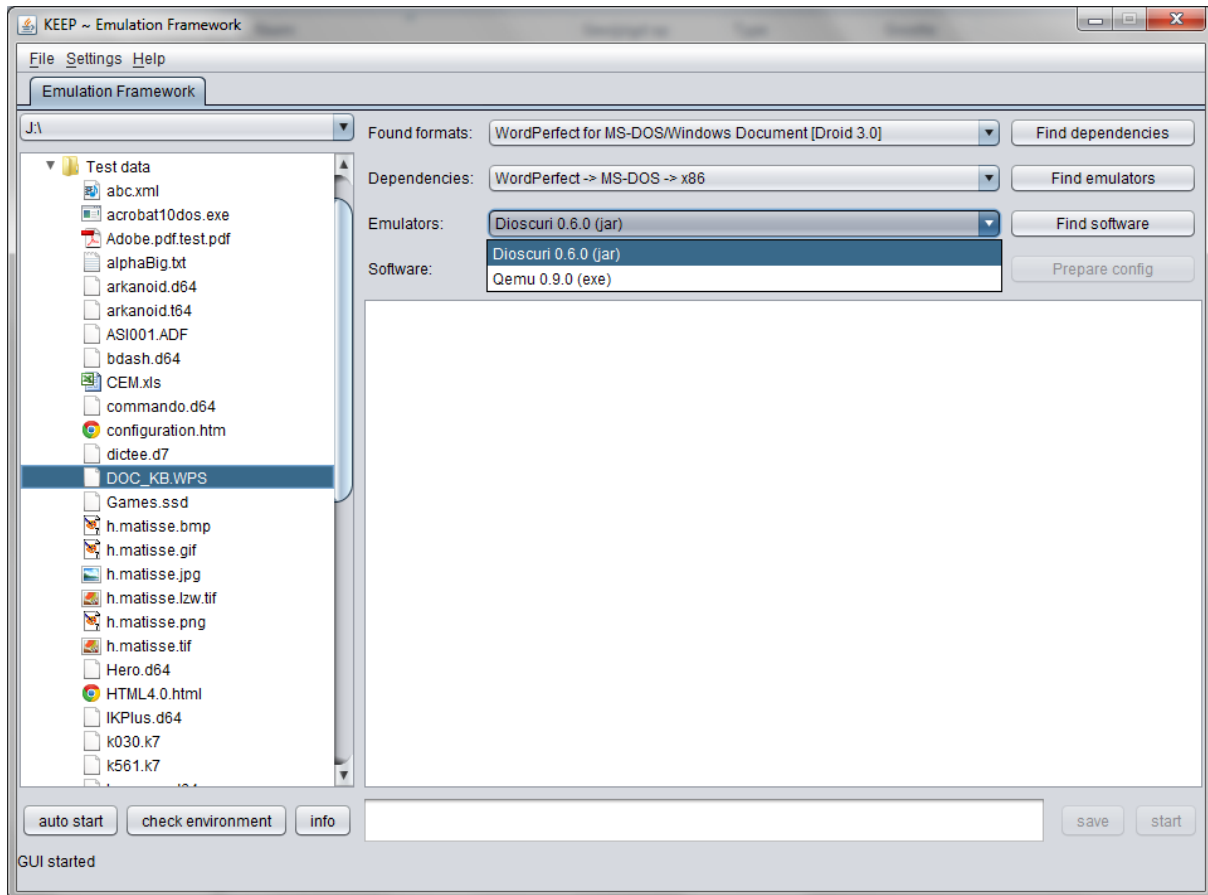
In this case, two emulators have been found and one of these, Dioscuri, is selected.

Note: based on language settings of the EF, it might be that you see less available emulators than you expected. See section **User settings** for more information.

2.5.4 Step 4 – Find software

Once an emulator has been selected, the required software that is part of the dependency needs to be selected and downloaded. Therefore, the EF queries the Software Archive for available software images (pre-installed operating system and applications) that match the required dependency stack. To start this process, the button **Find software** must be pressed.

Similar to querying the Emulator Archive either a warning at the status bar at the bottom of the screen will inform you that no software has been found, or the valid software will be shown in the result box, as can be seen in the following screenshot:
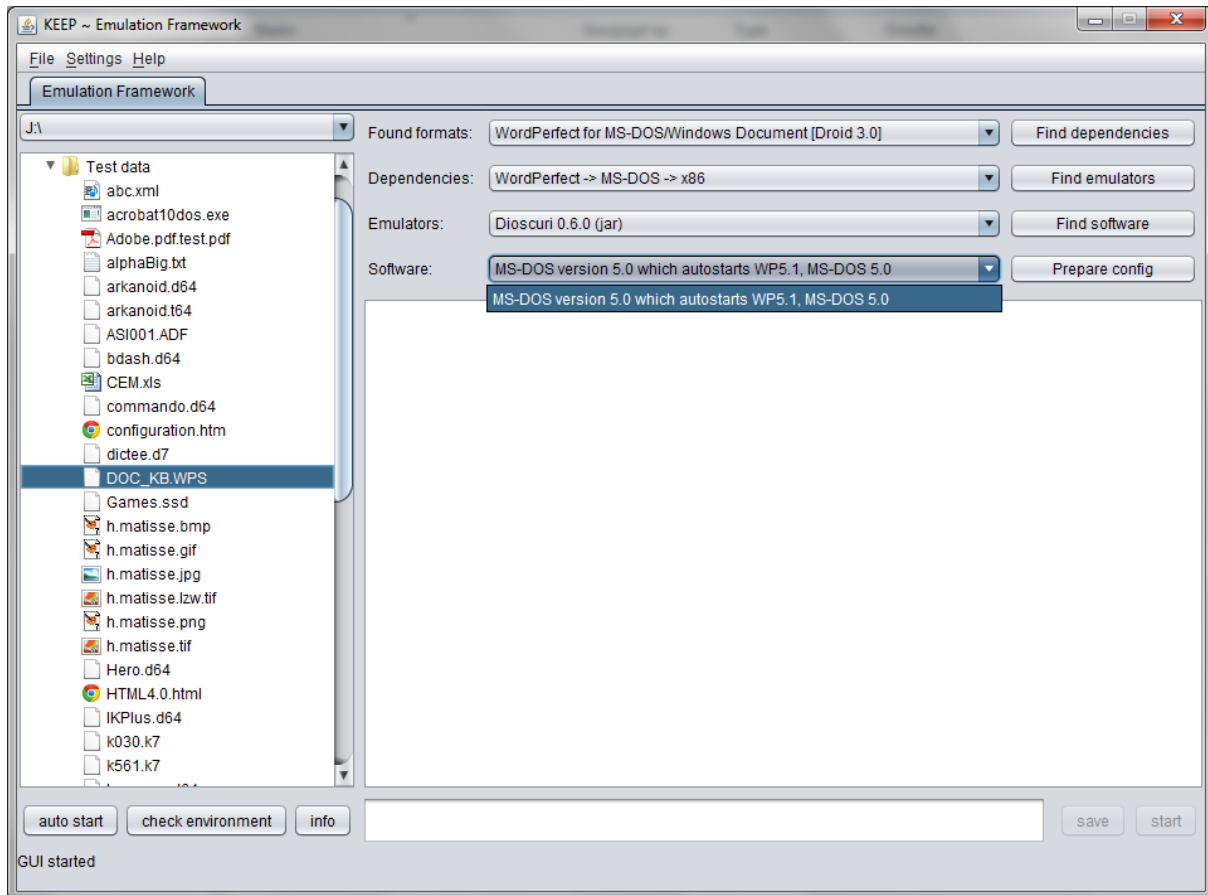
This example has returned one software package that offers the required operating system (MS-DOS 5.0) and application (WordPerfect 5.1).

Now the selection process is complete.

2.5.5   Step 5 – Prepare configuration & start

The final step before starting the emulator is configuration of the whole environment. By clicking the **Prepare config** button the EF automatically defines a suitable configuration. This configuration contains information about how the software and digital file should be loaded into the emulator.

The user can accept the predefined configuration and directly start the emulator by clicking the **start** button.

If you would like to view or alter the predefined settings, you can take a look at the bottom right panel of the GUI. A sample of such configuration is shown in the following view:
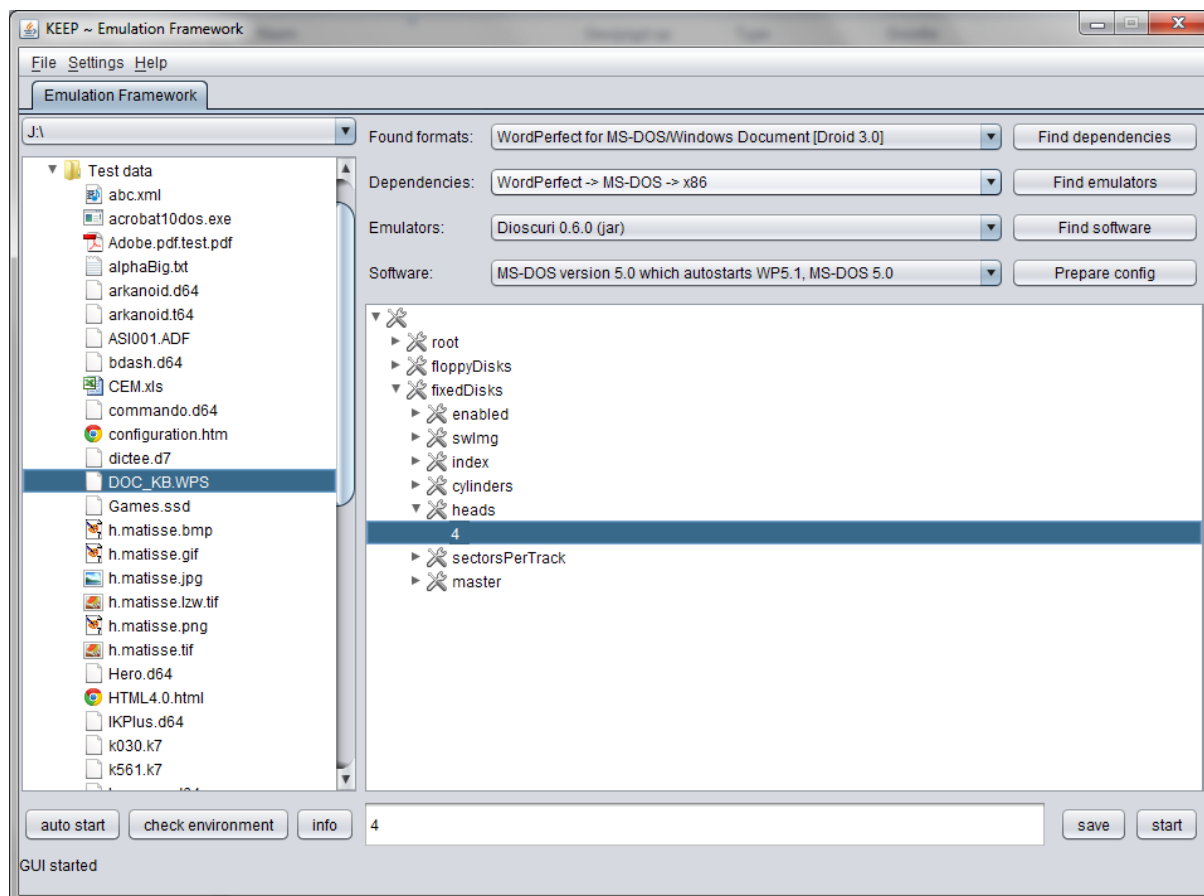
The configuration can be expanded by pressing the tree nodes. Parameters can be modified by selecting them, and in the textbox below the configuration tree, entering a new value.

Once a value is changed, the textbox will light up yellow to indicate a new value has been entered. The value is saved in the configuration by pressing the **save** button.

Note: it is recommended that only <u>expert users</u> change parameters in the configuration; entering the wrong parameters may stop the environment from running correctly!

Once the configuration update is complete (or no changes have been made), the **start** button can be pressed to load the environment and show the digital file.

2.6   User settings

The EF offers several user settings to:

- select the language of the EF GUI;

- select the preferred language(s) of the emulated computer environments;

- setup the connection to software and emulator archives.

You can find those settings by opening the menu item **Settings** in the menu bar.

2.6.1 Language settings

By clicking option **Languages** from the menu **Settings**, the following window will appear:
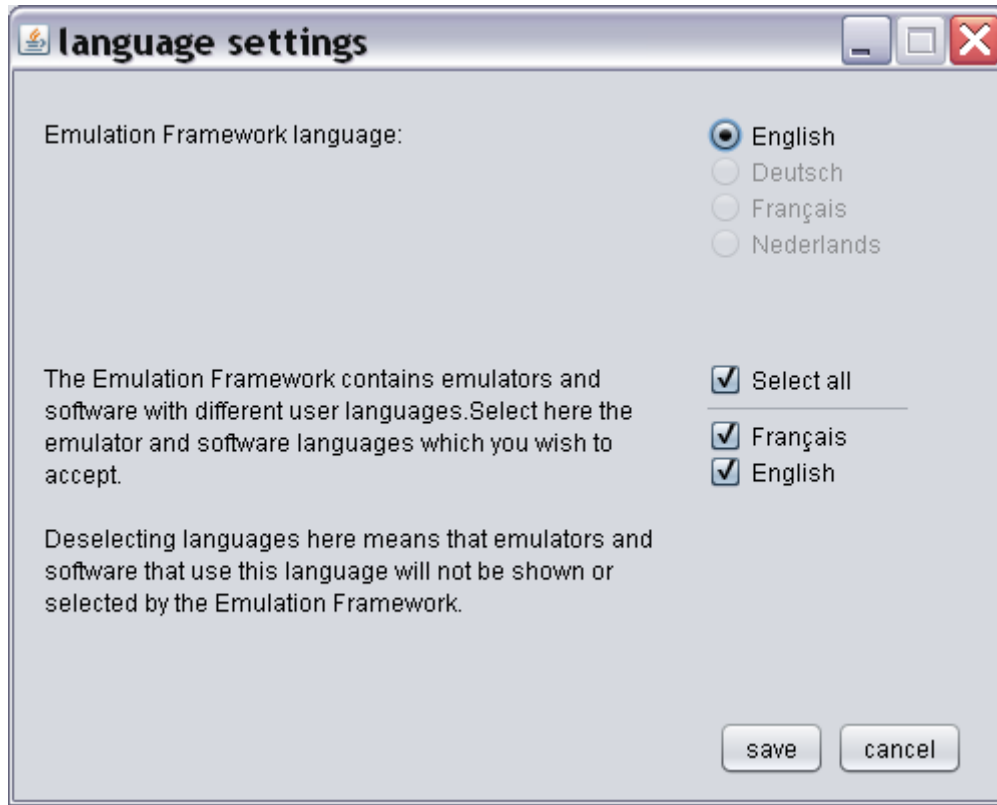

At the top of the window, you can select the language of the EF GUI.

Note: in the current release, English is only supported. Future releases will offer better support for other languages as well.

In the lower part of the window you can select which language should be supported by the emulator and software under emulation. The default is that both English and French are supported. You can uncheck one of the boxes if you wish.

2.6.2 Web service settings for software and emulator archives

By clicking option **Web-service addresses** from the menu **Settings**, the next window will be shown:
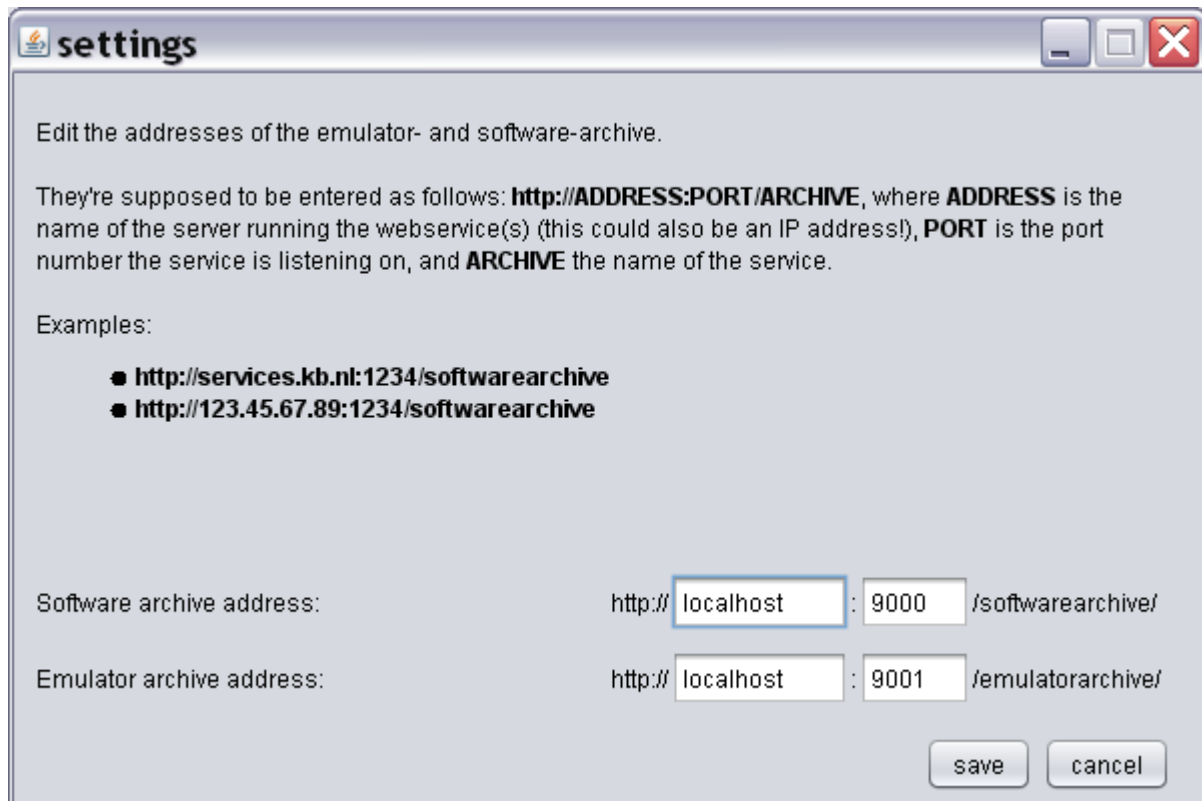
In this view, you can adjust the connection to the Software archive and Emulator archive. Please make sure you know the server name and port number of each of the web services. If you are unsure, please contact your system administrator.

Click the **Save** button to make the changes.

Note: the default settings for the web services are:

* Software archive: `local host` on port 9000

* Emulator archive: `local host` on port 9001

3 Administrator guide

This section is intended for administrators that would like to deploy the Emulation Framework (EF) on a server or local computer. Furthermore, this guide describes how to configure the components, and how to create and update the contents of the Emulator Archive and Software Archive.

3.1 Installation

The software is distributed via a single installation package. This is a self-extracting jar file that contains a wizard for installing the software. The software can be downloaded from:

http://sourceforge.net/projects/emuframework/files/

3.1.1 System requirements

Before you install the EF software, please make sure your computer and/or server(s) meet the following minimum system requirements:

Processor	x86 32/64 bit 1.5 GHz or faster
Memory	at least 2GB of RAM
Storage space	200 Megabytes of free available space for the base install. Depending on the number of emulators and software images, from 1 GB upwards.
Operating system	Linux or Windows, with a compatible JRE and network support
Java Runtime Environment (JRE)	Oracle/Sun JRE version 1.6 or higher / compatible.

Note: the latest Oracle/Sun JRE can be downloaded from:

http://www.oracle.com/technetwork/java/javase/downloads/index.html
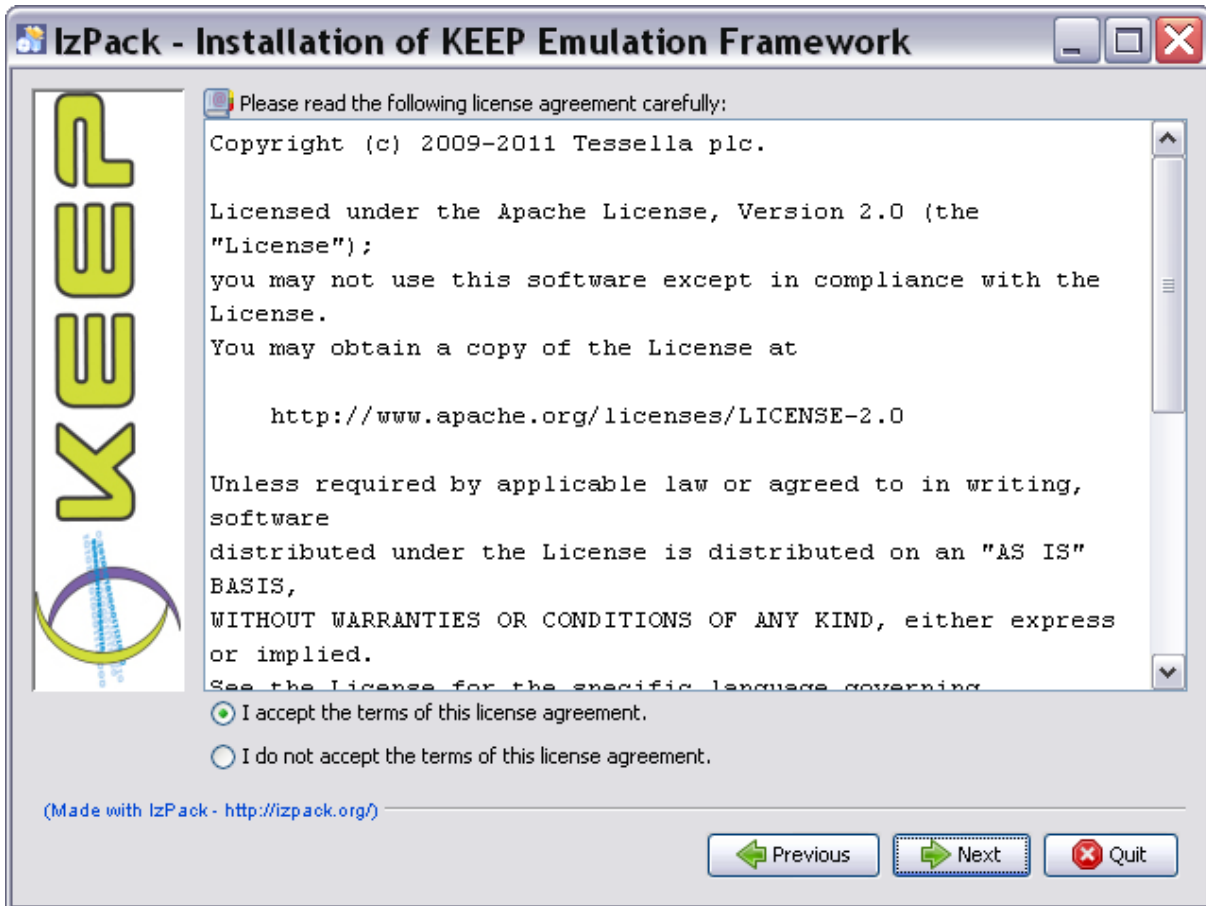
3.1.2 Installation instructions

After you have downloaded the installation package, start the JAR-file by double clicking or starting it via the command prompt. This will start the installation wizard.
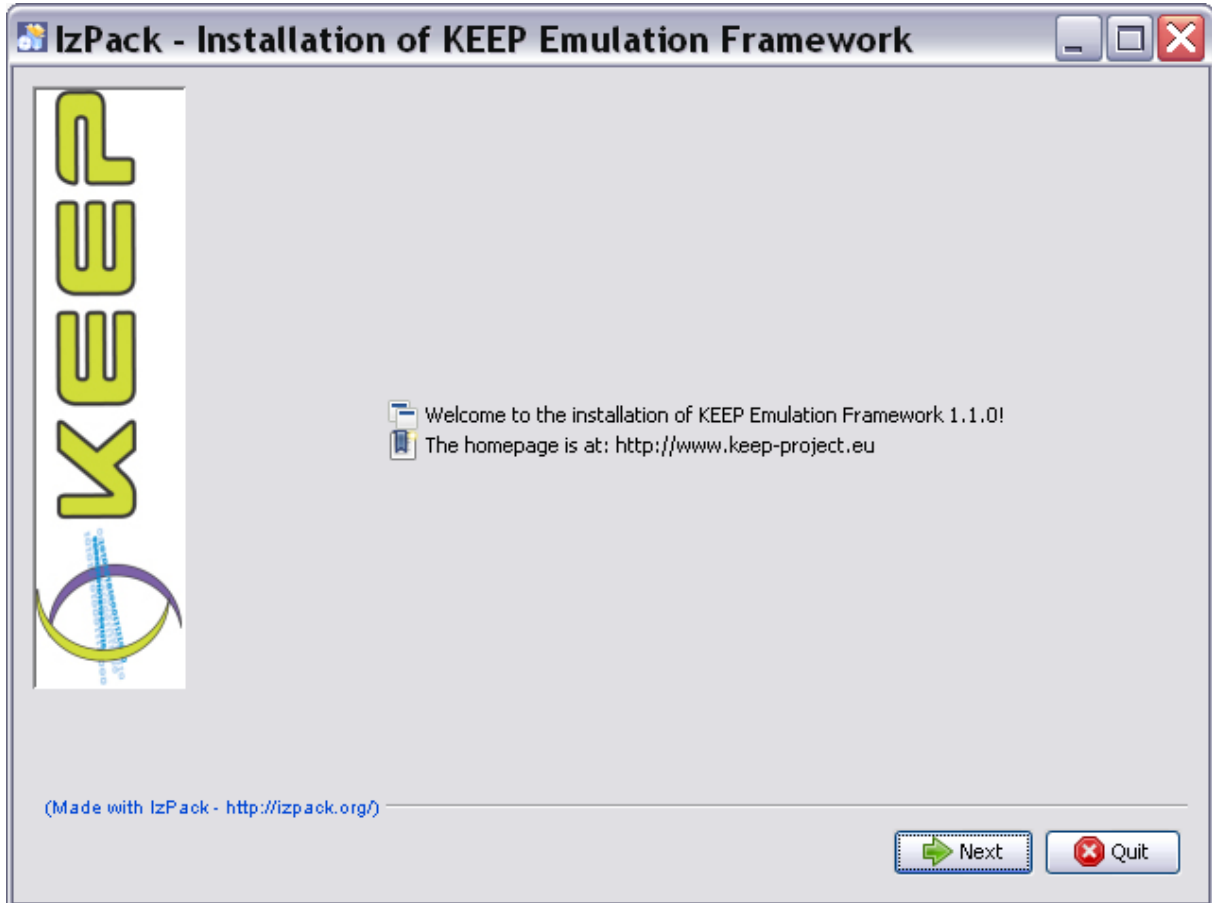
A welcome screen is displayed which contains a brief introduction to the project and Emulation Framework. Click **Next** to proceed.

A license agreement is then presented to the user. If you agree, tick the *"I accept the terms of this license agreement"* and click **Next**.
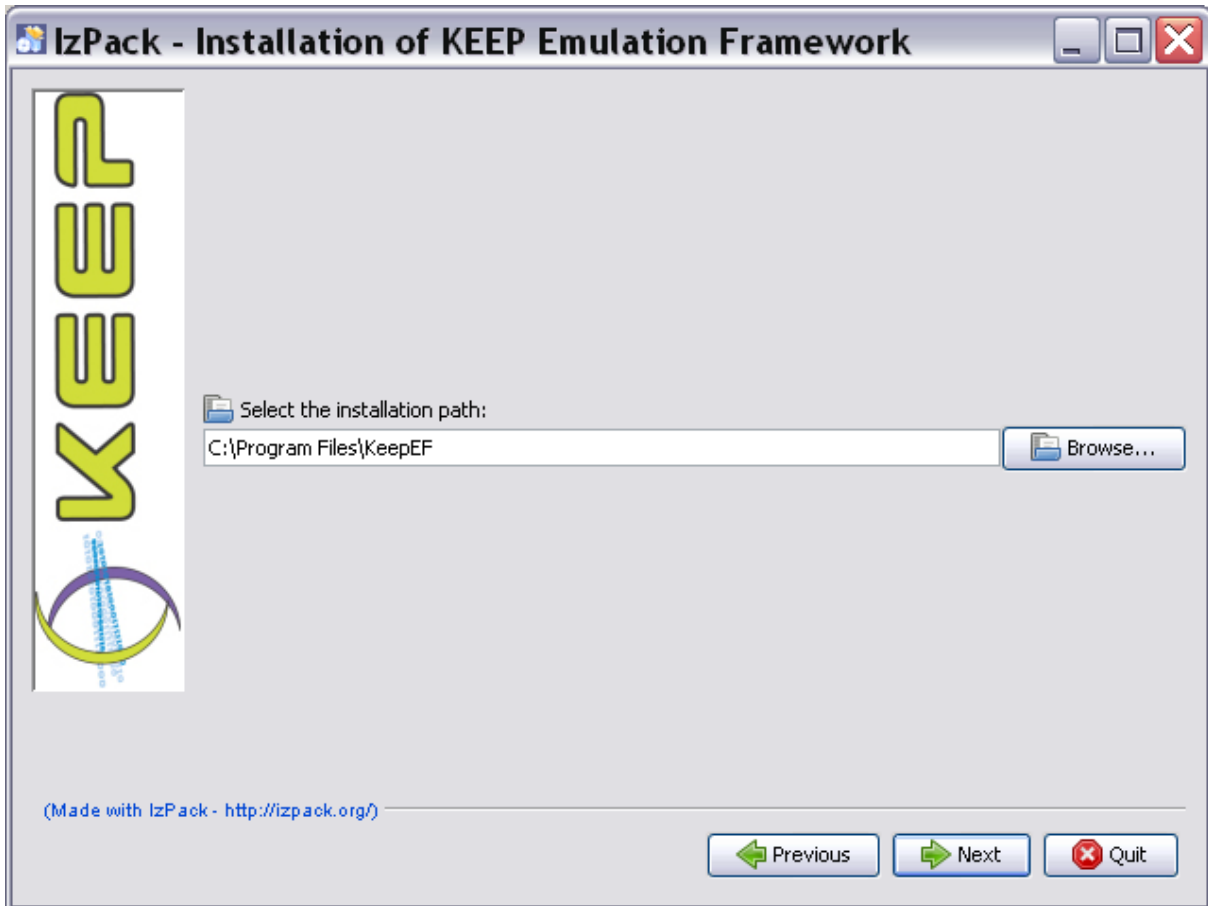
Once the license agreement is accepted, the user is prompted for a location to install the software. The default is set to:

    `C:\Program Files\KeepEF`   (MS Windows)

    `~/KeepEF`   (Linux)

Click **Next**.

Note: if the directory specified does not exist, the user will be asked for a confirmation to create that directory (see dialogue box). In case a write error is shown, you probably do not have enough rights to install the EF in the selected folder. Please try a different storage location or make sure you have enough access rights.
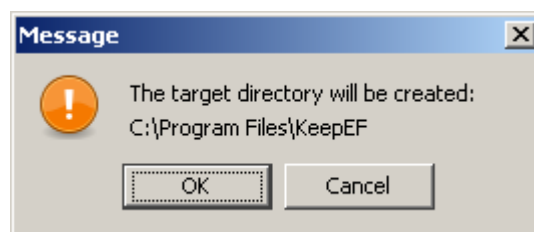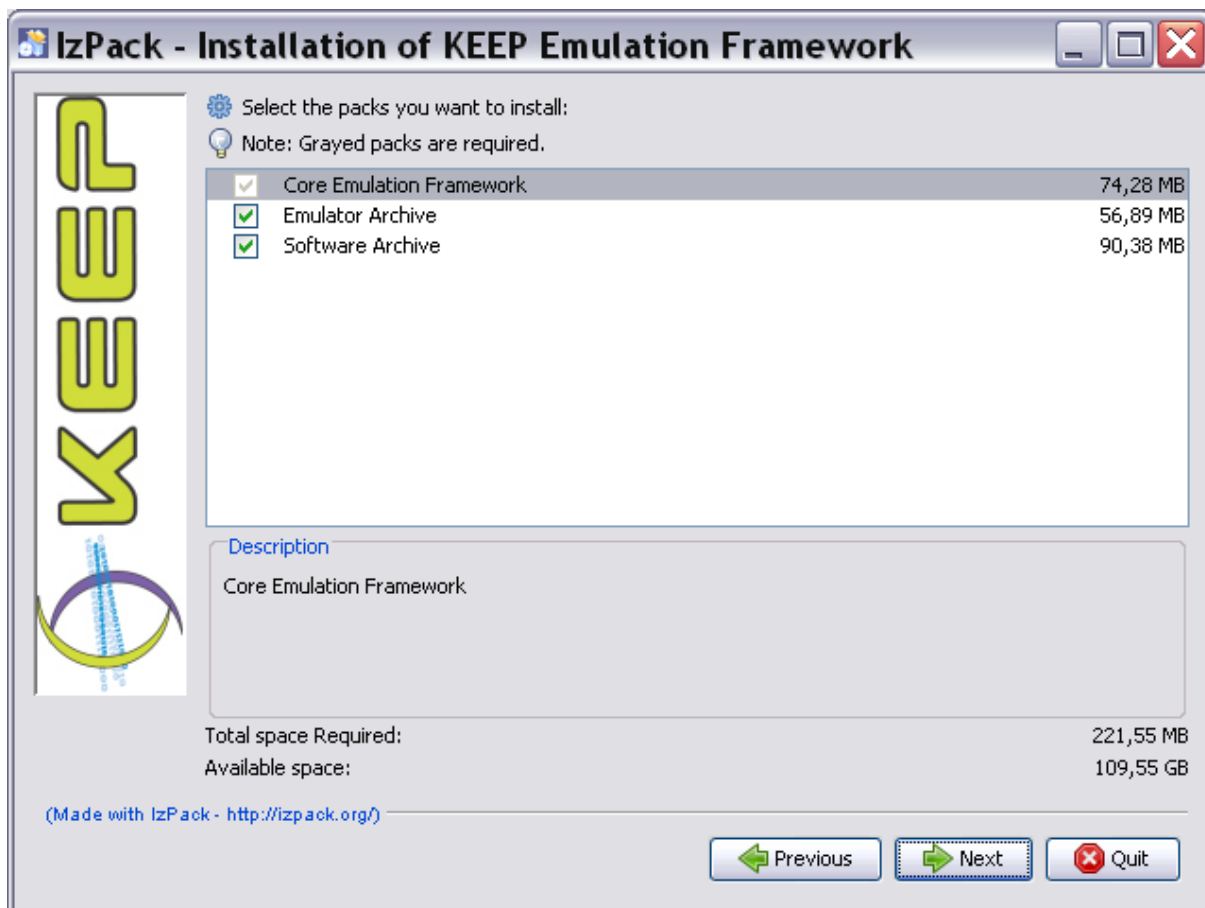
The next screen displays a list of the available components to be installed for the EF. The Core Emulation Framework is the only required component and thus always selected. The Emulator Archive and Software Archive can optionally be selected.

Note: although the Emulator Archive and Software Archive are optional, the Core still requires these to function correctly. However, if these are available on a central server via a remote http-connection, there is no need for a local installation and the Core Emulation Framework will be installed in 'client' mode. If the Emulator Archive and Software Archive are not available remotely, or if you are installing the Emulator Framework on a server, the Emulator Archive and Software Archive need to be included, and the Emulation Framework will be installed in 'server' mode.

Click **Next**.
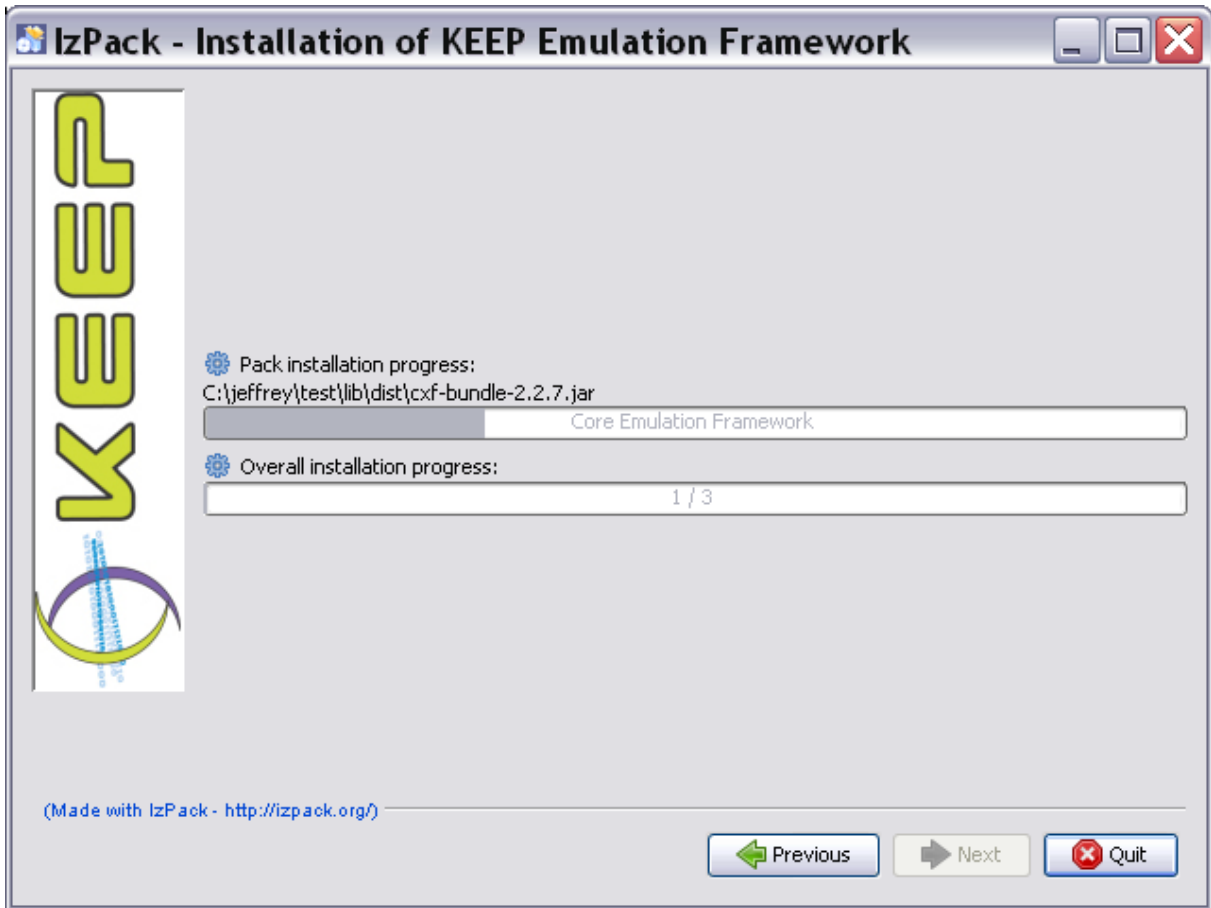
The progress of the installation process is shown for each component and the overall system. Installation might take a couple of minutes depending on your computer. After installation is finished, click **Next**.
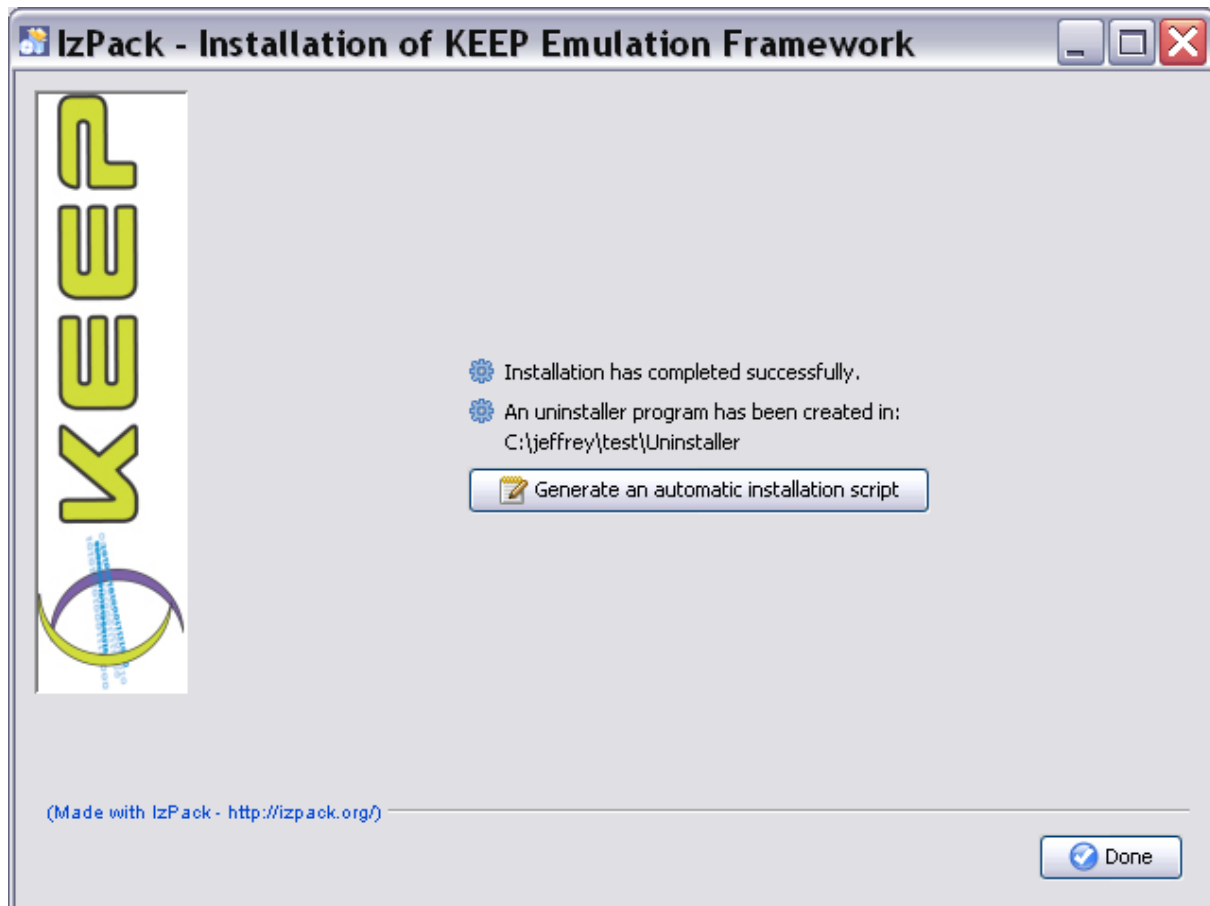
Finally, you are prompted that installation has completed. Click **Done** to close the installation wizard.

Note: The button 'Generate an automatic installation script' provides an XML file with the settings used during the install, which can be used to automate the process. This can be useful for installing the software on many computers, using the same settings.

3.2   Uninstall

An uninstall program is provided automatically by the installation wizard and can be found in the *uninstall* subdirectory of the chosen installation directory. It will remove all installed components, and will remove the installation directory (optional).
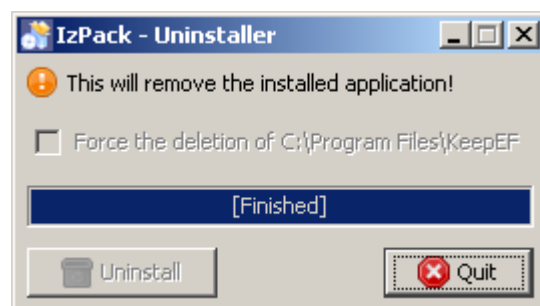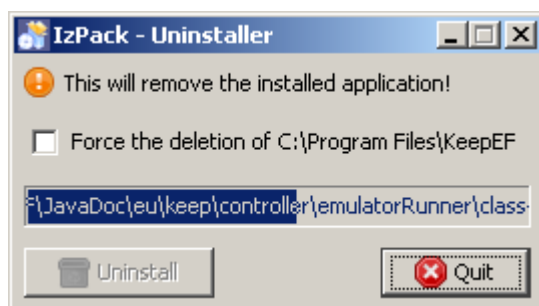
## 3.3 Managing the EF, Software Archive and Emulator Archive

If you have installed the EF including software and emulator archives locally or on a server, you will have the possibility to manage the EF and archives via the EF GUI. To start the GUI, click on:

`runEF.bat` (for Windows platforms)

or

`runEF.sh` (for Linux platforms)

Note: as an administrator you will have more functions available via the GUI than an end-user. The following view shows the GUI with administrator rights.
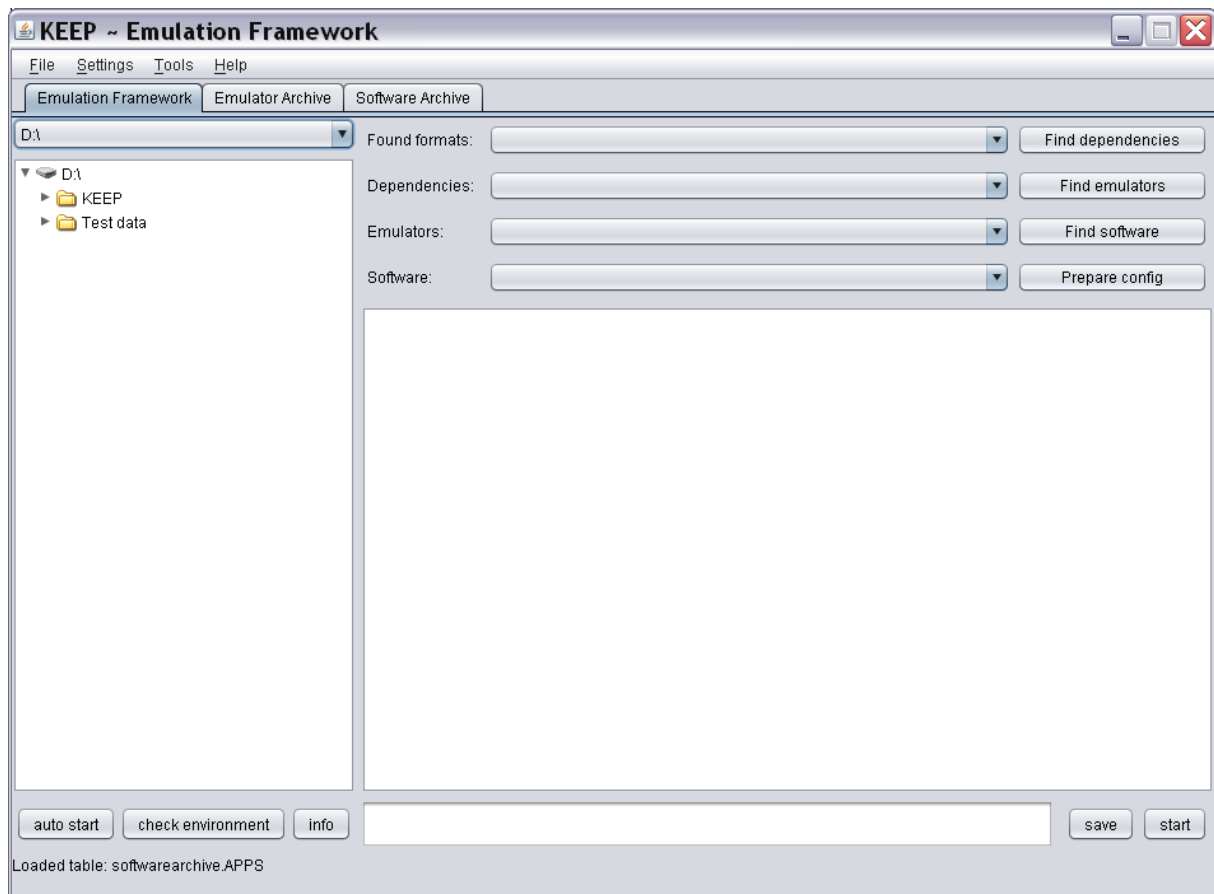

Aside from the default tab Emulation Framework, you also have two extra tabs:

- **Emulator Archive**

- **Software Archive**

Furthermore, the menu bar shows the menu item **Tools**. The next sections will explain in more detail how to use the extra functions.

3.3.1 Manage Emulator Archive

The tab **Emulator Archive** allows you to view the contents of the archive. The following screen shows a listing of all available emulators in the archive.
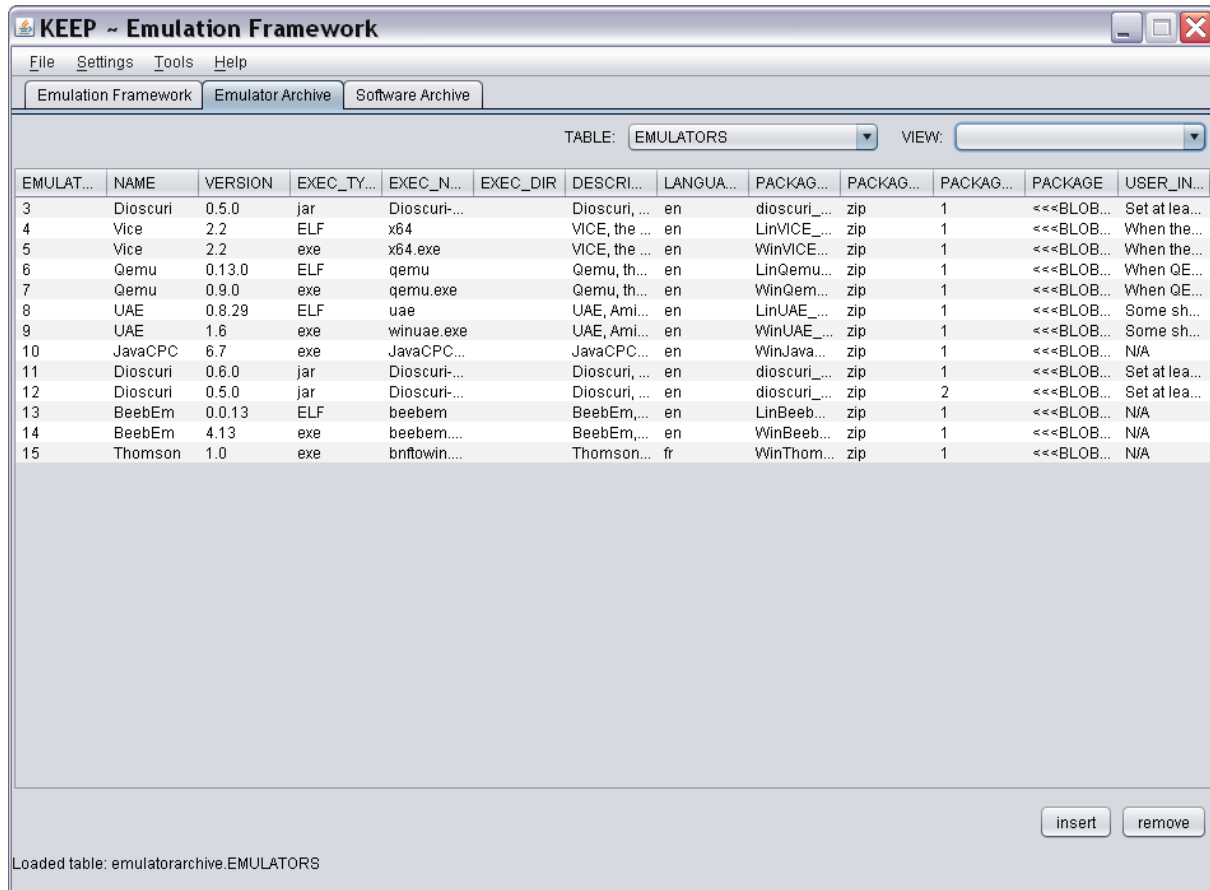
Click on the drop down box **Table** to select a database table. Click on **View** to select a particular merged view on several tables at once (if available).

Via the buttons **insert** and **remove** at the bottom on the screen you can create or delete an entry. Double-clicking on an existing entry allows you the possibility to change the value.

Note: be careful as this has direct consequences for the functionality of the EF. Perform changes to these tables only if you are really sure what you are doing!

Aside from viewing the contents of the Emulator Archive, the EF offers a wizard to automatically add a new emulator to the archive. To start the wizard, open the menu **Tools** and click on **Add emulator…**

The following pop-up window will appear.
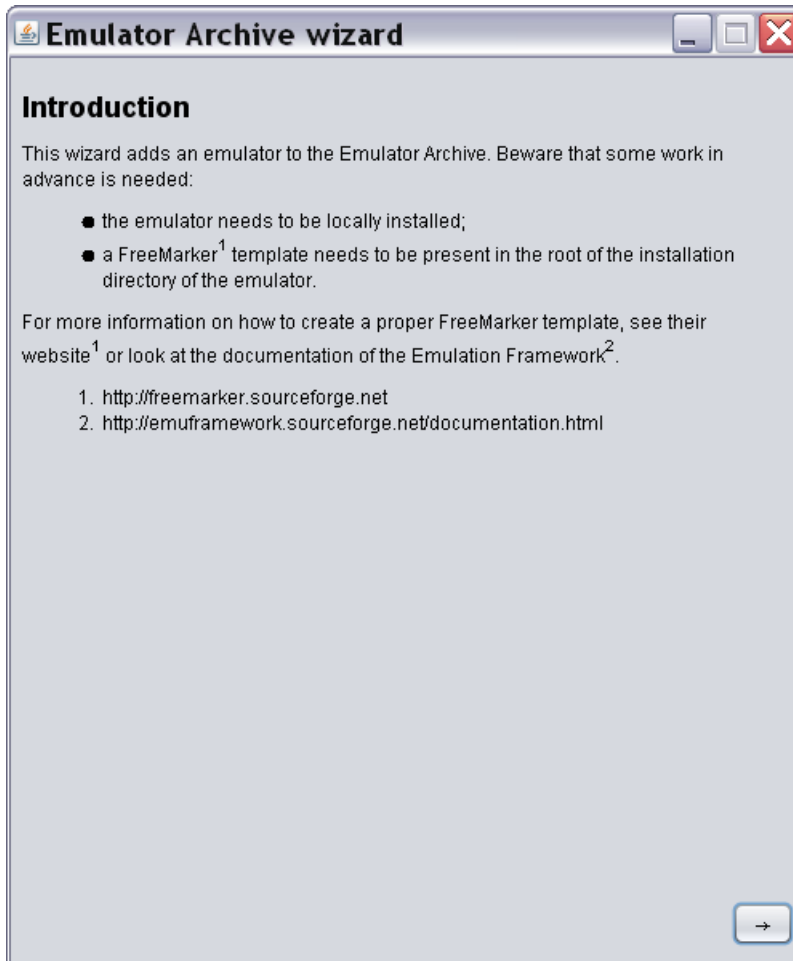
Before you can continue to add a new emulator to the Emulator Archive, please make sure you have done the following:

1. have a locally installed copy of the emulator;
2. have a Freemarker template for that particular emulator. This template file (ftl format) needs to be stored in the root of the folder of the installed emulator.

Depending on the type of template, the Freemarker file must comply to one of the following names:

```
templateCLI.ftl
```

```
templateXML.ftl
```

```
templateProps.ftl
```

Note: to learn more about how to create a Freemarker template file, please visit:

the Freemarker website, http://freemarker.sf.net

and the EF documentation page, http://emuframework.sf.net/documentation.html

If you have accomplished above preparations, please click the **-> [ARROW]** button at the bottom to continue. This will start the wizard containing two steps.

Step 1. In this step you are asked to enter some basic information about the emulator:

Name – enter the name of the emulator (e.g. Dioscuri).

Version – enter the version number of the emulator (e.g. 3.1.4).

Executable type – exe for DOS/Windows, JAR for Java, ELF for Linux.

Emulator executable – point to the file that launches the emulator (e.g. start.bat).

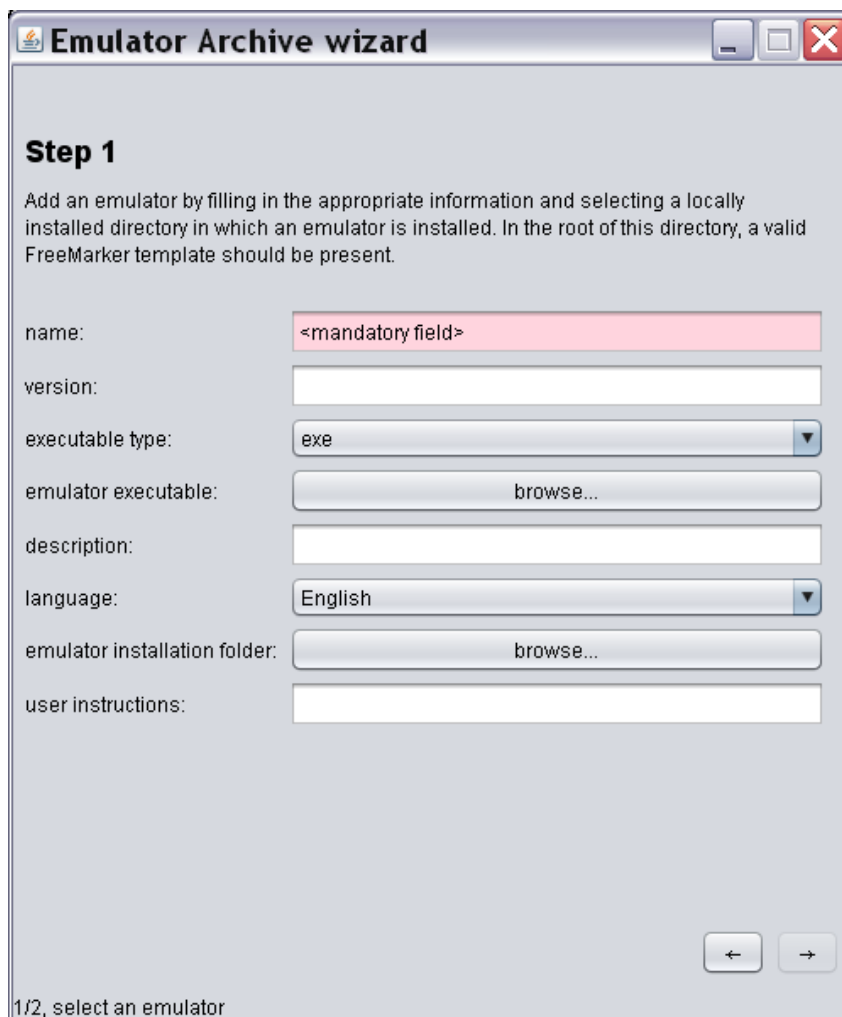Description – extra information about the emulator (e.g. 8086 emulator).

Language – the native language of the emulator (e.g. English).

Emulator installation folder – the root folder in which the emulator is installed.

User instructions – extra information for end-users (e.g. To run in full screen mode, press F4)

Note: the installation folder of the emulator needs to contain the Freemarker template file as well.

The following screen is shown.

Click **-> [ARROW]** button to continue. Or click **<- [ARROW]** button to go back.

Step 2. In this final step you are required to define which hardware platform this emulator emulates and which image file system format it supports. Use the drop down boxes to make a choice.

The following screen is shown.

Click **-> [ARROW]** button to continue. Or click **<- [ARROW]** button to go back.

After that, you will be asked to review your entered information and confirm it by clicking the **confirm** button.

If you would like to make any changes still you can step back clicking the **<- [ARROW]** button. Or **cancel** if you want to stop the wizard.

The following screen gives an example of how this could look like.

The process of adding a new emulator is successful if the status bar shows the message:

Emulator successfully committed!

3.3.2 Manage Software Archive

The tab **Software Archive** allows you to view the contents of the archive. The following screen shows a listing of all available applications in the archive.

Click on the drop down box **Table** to select a database table. Click on **View** to select a particular merged view on several tables at once (if available).

Via the buttons **insert** and **remove** at the bottom on the screen you can create or delete an entry. Double-clicking on an existing entry allows you the possibility to change the value.

Note: be careful as this has direct consequences for the functionality of the EF. Perform changes to these tables only if you are really sure what you are doing!
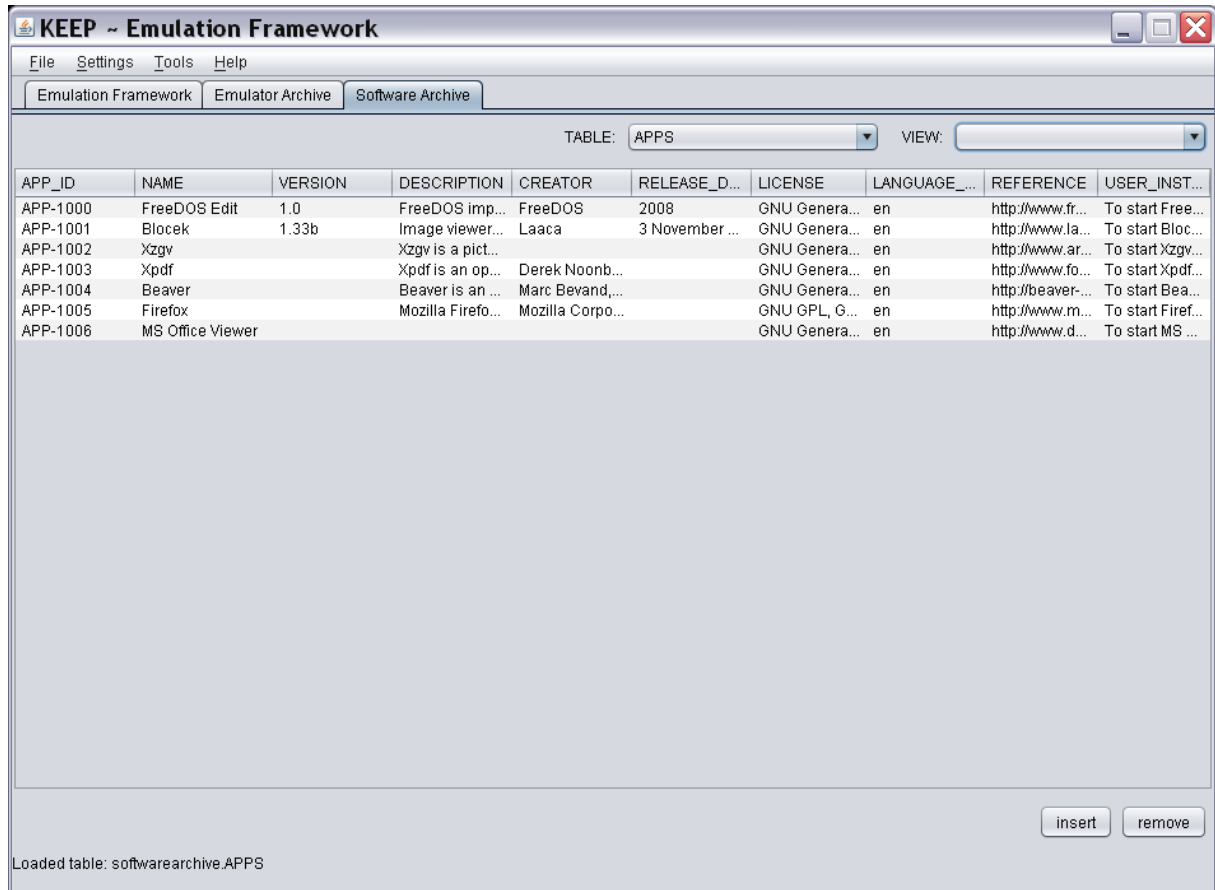
Aside from viewing the contents of the Software Archive, the EF offers a wizard to automatically add new software to the archive. To start the wizard, open the menu **Tools** and click on **Add software…**

The following pop-up window will appear.
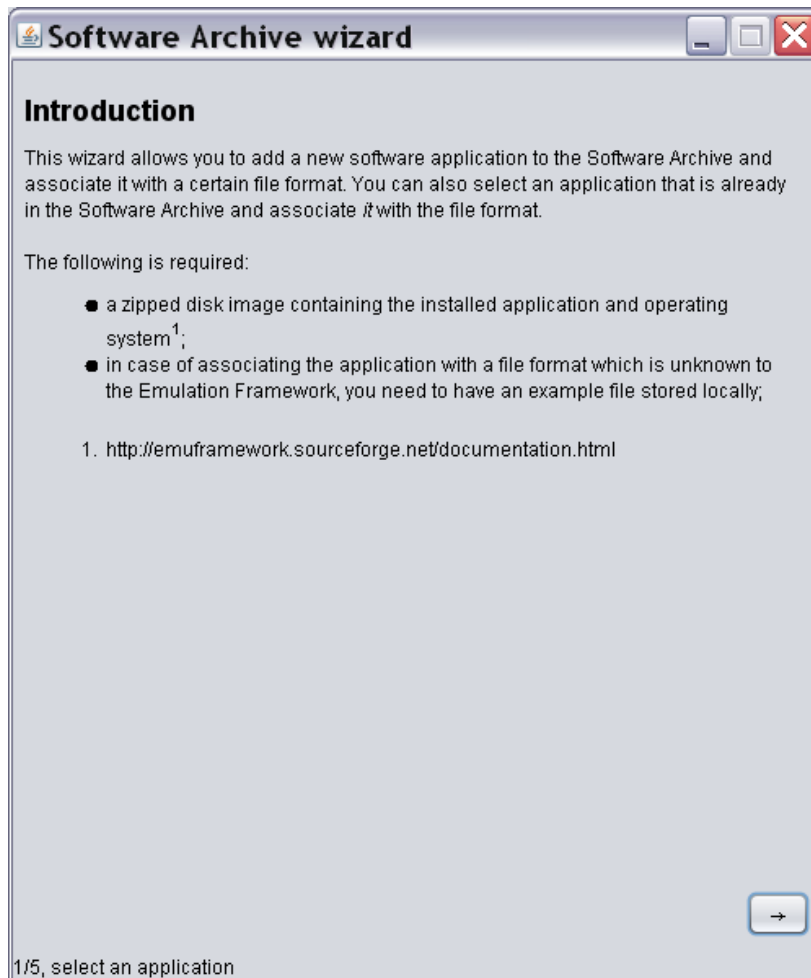
Before you can continue to add new software to the Software Archive, please make sure you have done the following:

1. prepared a zipped disk image containing the installed application and operating system. The disk image should be in a format that is known to one or more of the emulators in the Emulator Archive. For example, the emulator QEMU supports flat file disk images. The best approach is to create such a disk image using supporting tools often supplied by open source emulators such as QEMU. In that case, you are assured that the file will be recognised by the EF. After creating the disk image file the file should be zipped using a standard zip-tool (e.g. WinZIP, 7zip). The zipped file may either be compressed or uncompressed.

2. have an example file available in the file format that you would like to be opened by the software you are about to add. For example, a Wordperfect file if you would like to upload Wordperfect with MS DOS into the archive.

If you have accomplished above preparations, please click the **-> [ARROW]** button at the bottom to continue. This will start the wizard containing five steps.

Step 1. In the first step you will be asked to enter some basic information about the software you would like to add. If similar software is already in the database (i.e. might be an older version or running a different operating system), you can do a quick match selecting the right software from the drop down box.

If this software does not yet exist in the archive, make sure you enter the following field:

Name – the full name of the application (Microsoft Office 2000).

Version – the version number (e.g. 9.0).

Description – short description of the software (e.g. Office suite containing …).

Creator – the organisation or person(s) that created the software (e.g. Microsoft).

Release date – the official date the software was published (7 June 1999).

License – the type of license the software was distributed with.

Language – the language of the application or operating system.

Reference – information for further reading (e.g. http://en.wikipedia.org/wiki/Microsoft_Office_2000)

User instructions – extra information to guide the user through the software.

Click **-> [ARROW]** button to continue. Or click **<- [ARROW]** button to go back.

Step 2. In this step you are asked to give details on which operating system the software runs. Similar as with step 1, you can chose between selecting an existing operating system or to enter a new one.

Click **-> [ARROW]** button to continue. Or click **<- [ARROW]** button to go back.

Step 3. Now that the details of the application and operating system are known, the EF would like to know which hardware is required to run the software. Via the drop down boxes you can select one of the supported hardware platforms and the type of file system the disk image is organised.

Click **-> [ARROW]** button to continue. Or click **<- [ARROW]** button to go back.

Step 4. In this step you are asked to point the wizard to the location where the zipped disk image containing the software (preinstalled application and operating system) is kept.
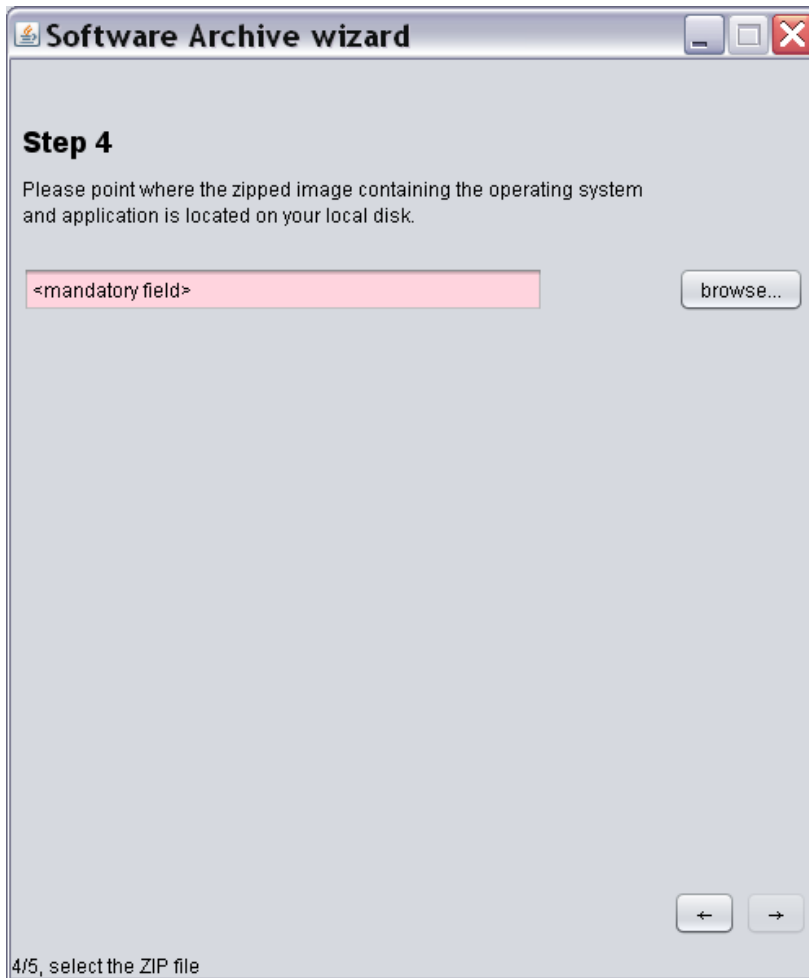
Click **-> [ARROW]** button to continue. Or click **<- [ARROW]** button to go back.

Step 5. Finally, you are asked to associate the software with a file format. This is necessary to let the EF understand that a certain file format can be opened with this software. To make more than one association, you can go through this wizard several times. Hereby you can select existing application and operating system rather than uploading the software again.

Again, two approaches for associating file format with software exist: select an existing file format or create a new file format.

Creating a new file format should be done by uploading an example file and let the EF automatically detect it. Furthermore, you are requested to enter the version, description and a reference for the format.

Click **-> [ARROW]** button to continue. Or click **<- [ARROW]** button to go back.

Finally, you are asked to review and confirm your entered information by clicking the **confirm** button.

If you would like to make any changes still you can step back clicking the **<- [ARROW]** button. Or **cancel** if you want to stop the wizard.

The following screen gives an example of how this could look like.
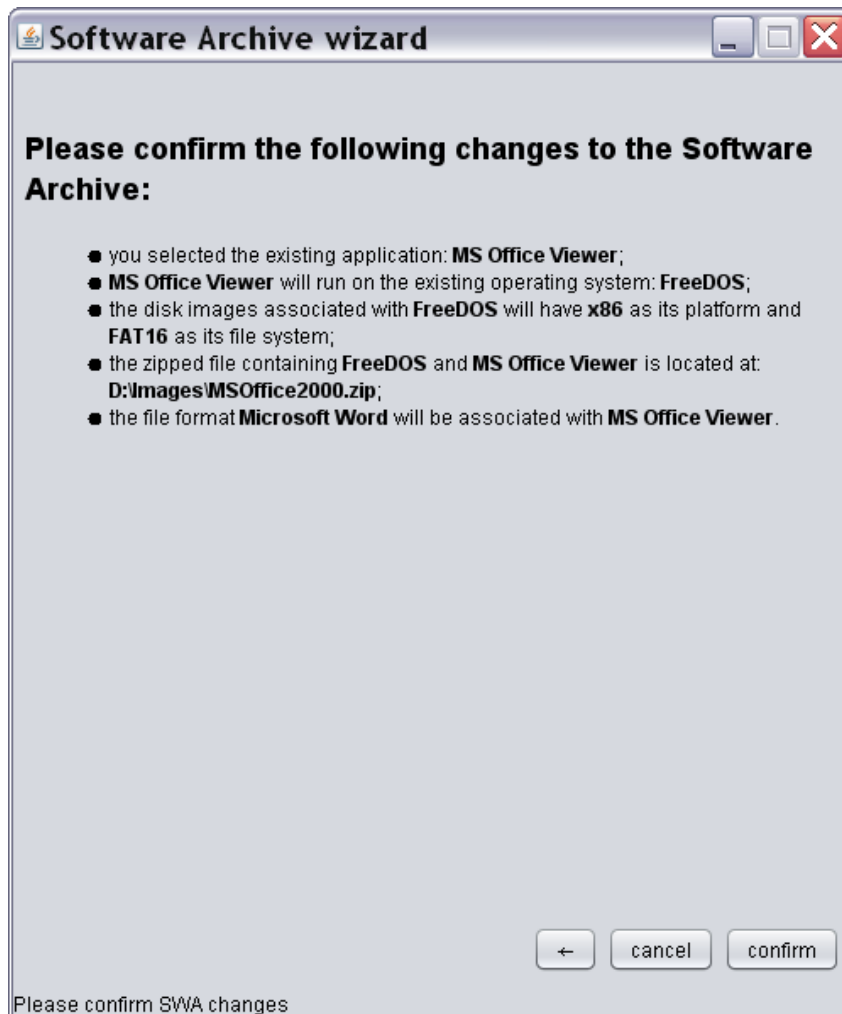
The process of adding new software is successful if the status bar shows the message:

Successfully committed changes!

3.4 Advanced configuration

The three EF components can all be separately configured by editing the configuration files. Although each file configures each component separately, there are some parameters that need to match between them to be able to connect to the Software and Emulator Archive successfully. The details for each configuration file are outlined in the next sections.

3.4.1 Core configuration

The configuration is stored in a file called **user.properties**, located in the ./eu/keep subdirectory, and contains the internal database connection properties, the location of the Emulator and Software Archive, as well as the various directories used by the system.

This file can be edited by hand. The table below outlines the configuration properties and their defaults.

Property key	Default property value	Comment
h2.db.driver	org.h2.Driver	JDBC driver class
h2.jdbc.prefix	jdbc:h2:	JDBC url prefix
h2.db.url	./database/h2/EF_engine	Database location on disk
h2.db.exists	;IFEXISTS=TRUE	Flag to indicate whether the database should be created (FALSE), or fail (TRUE) if it doesn't exist.
h2.db.server	;AUTO_SERVER=TRUE	The database connection type
h2.db.schema	;SCHEMA=engine	Name of the schema used for the internal database
h2.db.admin	sa	Database admin login name
h2.db.adminpassw	CEF_Engine	Database admin password
h2.db.user	cef	Database user login name
h2.db.userpassw	cef	Database user password
software.archive.url[2]	http://localhost:9000/softwarearchive/	Location of the Software Archive
emulator.archive.url[2]	http://localhost:9001/emulatorarchive/	Location of the Emulator Archive
exec.dir	./exec	Temporary directory where the emulators are installed for use (will be deleted after use)
system.tmpdir	./tmp	Directory used by the host system to store temporary files
accepted.languages[3]	all	Emulator and Software languages which are accepted. Emulators or Software which use other languages will not be selected or displayed. Languages are entered as a comma-separated list of 2-letter 'locale'-codes, e.g. en (English), nl (Dutch), fr (French), de (German). See the System Maintenance Guide for a complete list of language codes. To accept all available languages, enter 'all'.

Note: the './' in some path definitions describe the installation directory (see section *Installation*).

---

[2] The Software Archive and Emulator Archive url can also be configured via the EF GUI (see section 2.6).

[3] The list of accepted languages can also be configured via the EF GUI (see section 2.6).

The GUI has a separate configuration file called **gui.properties**, also located in the ./eu/keep subdirectory.

This file contains the database connection parameters for the different components; depending on whether the Emulation Framework was installed in 'client' or 'server' mode, this may be just the EF, or the EF, EA and SWA. This file can also be edited by hand.

The table below outlines the configuration properties and their defaults.

Property key	Default property value	Comment
ef.db.driver	org.h2.Driver	JDBC driver class
ef.jdbc.prefix	jdbc:h2:	JDBC url prefix
ef.db.url	./database/h2/EF_engine	Database location on disk
ef.db.exists	;IFEXISTS=TRUE	Flag to indicate whether the database should be created (FALSE), or fail (TRUE) if it doesn't exist.
ef.db.server	;AUTO_SERVER=TRUE	Database connection type
ef.db.schema.name	engine	Name of the schema used for the internal database
ef.db.schema	;SCHEMA=engine	Name of the schema used for the internal database
ef.db.admin	sa	Database admin login name
ef.db.adminpassw	CEF_Engine	Database admin password
ef.db.user	cef	Database user login name
ef.db.userpassw	cef	Database user password
ea.db.driver	org.h2.Driver	JDBC driver class
ea.jdbc.prefix	jdbc:h2:	JDBC url prefix
ea.db.url	./ea/database//EF_ea	Database location on disk
ea.db.exists	;IFEXISTS=TRUE	Flag to indicate whether the database should be created (FALSE), or fail (TRUE) if it doesn't exist.
ea.db.server	;AUTO_SERVER=TRUE	Database connection type
ea.db.schema.name	emulatorarchive	Name of the schema used for the internal database
ea.db.schema	;SCHEMA=emulatorarchive	Name of the schema used for the internal database
ea.db.admin	sa	Database admin login name
ea.db.adminpassw	EA_Engine	Database admin password
ea.db.user	ea	Database user login name
ea.db.userpassw	ea	Database user password
swa.db.driver	org.h2.Driver	JDBC driver class
swa.jdbc.prefix	jdbc:h2:	JDBC url prefix
swa.db.url	./database/h2/EF_swa	Database location on disk
swa.db.exists	;IFEXISTS=TRUE	Flag to indicate whether the database should be created (FALSE), or fail (TRUE) if it doesn't exist.
swa.db.server	;AUTO_SERVER=TRUE	Database connection type
swa.db.schema.name	softwarearchive	Name of the schema used for the internal database
swa.db.schema	;SCHEMA=softwarearchive	Name of the schema used for the internal database
swa.db.admin	sa	Database admin login name
swa.db.adminpassw	SWA_Engine	Database admin password
swa.db.user	swa	Database user login name
swa.db.userpassw	swa	Database user password

3.4.2 Emulator Archive configuration

Similar to the Core, the Emulator Archive also has a **user.properties** file for configuration. It contains the emulator database connection properties, and the location of the Emulator Archive.

The table below outlines the configuration properties and their defaults.

Property key	Default property value	Comment
h2.db.driver	org.h2.Driver	JDBC driver class
h2.jdbc.prefix	jdbc:h2:	JDBC url prefix
h2.db.url	./ea/database//EF_ea	Database location on disk
h2.db.exists	;IFEXISTS=TRUE	Flag to indicate whether the database should be created (FALSE), or fail (TRUE) if it doesn't exist.
h2.db.schema	;SCHEMA=emulatorarchive	Name of the schema used for the internal database
h2.db.admin	sa	Database admin login name
h2.db.adminpassw	EA_Engine	Database admin password
h2.db.user	ea	Database user login name
h2.db.userpassw	ea	Database user password
server.soap.address	http://localhost:9001/emulatorarchive/	webservice Server address

Note that the server.soap.address property must match the EF emulator.archive.url property for the connection between the two components to be successful.

3.4.3 Software Archive configuration

Similar to the Core, the Software Archive also has a **user.properties** file for configuration. It contains the software database connection properties, and the location of the Software Archive.

The table below outlines the configuration properties and their defaults.

Property key	Default property value	Comment
swa.db.driver	org.h2.Driver	JDBC driver class
swa.jdbc.prefix	jdbc:h2:	JDBC url prefix
swa.db.url	./database/h2/EF_swa	Database location on disk
swa.db.exists	;IFEXISTS=TRUE	Flag to indicate whether the database should be created (FALSE), or fail (TRUE) if it doesn't exist.
swa.db.schema	;SCHEMA=softwarearchive	Name of the schema used for the internal database
swa.db.admin	sa	Database admin login name
swa.db.adminpassw	SWA_Engine	Database admin password
swa.db.user	swa	Database user login name
swa.db.userpassw	swa	Database user password
server.soap.address	http://localhost:9000/softwarearchive/	Server address

Note that the server.soap.address property must match the EF software.archive.url property for the connection between the two components to be successful.

4   Problem solving

If you experience any troubles during installation or operation of the EF, please browse to the Emulation Framework support website, available at:

http://sourceforge.net/projects/emuframework/support

Or visit the overall project website: http://emuframework.sf.net