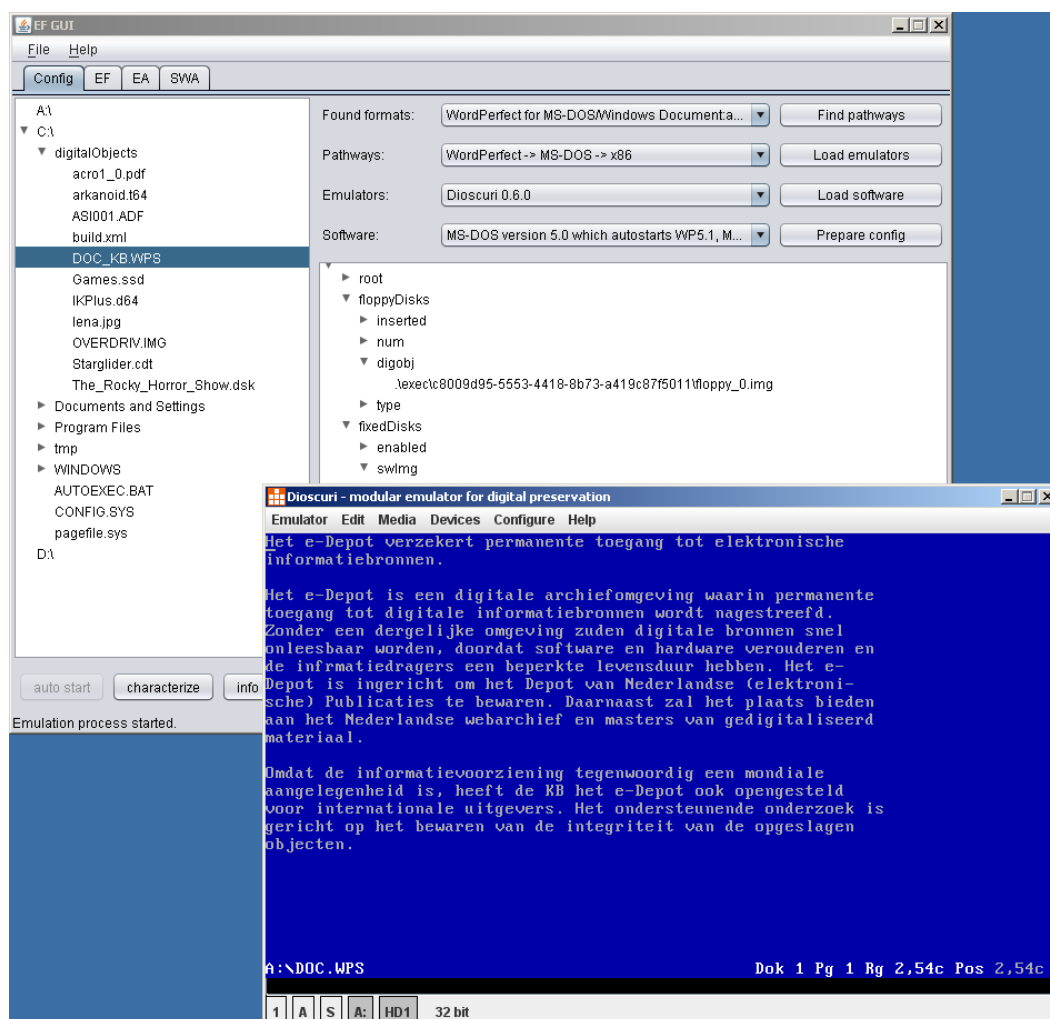




Keeping Emulation Environments Portable
FP7-ICT-231954

**System User Guide
for the Emulation Framework
version 1.0 (May 2011)**



Deliverable number	<i>Part of deliverable D2.3 (based on I2.2)</i>
Nature	<i>Report</i>
Dissemination level	<i>CO</i>
Status	<i>Draft / <u>Finalised</u> / Reviewed / Final</i>
Workpackage number	<i>WP2</i>
Lead beneficiary	<i>TES</i>
Author(s)	<i>Bram Lohman (Tessella), Jeffrey van der Hoeven (KB)</i>

Document history

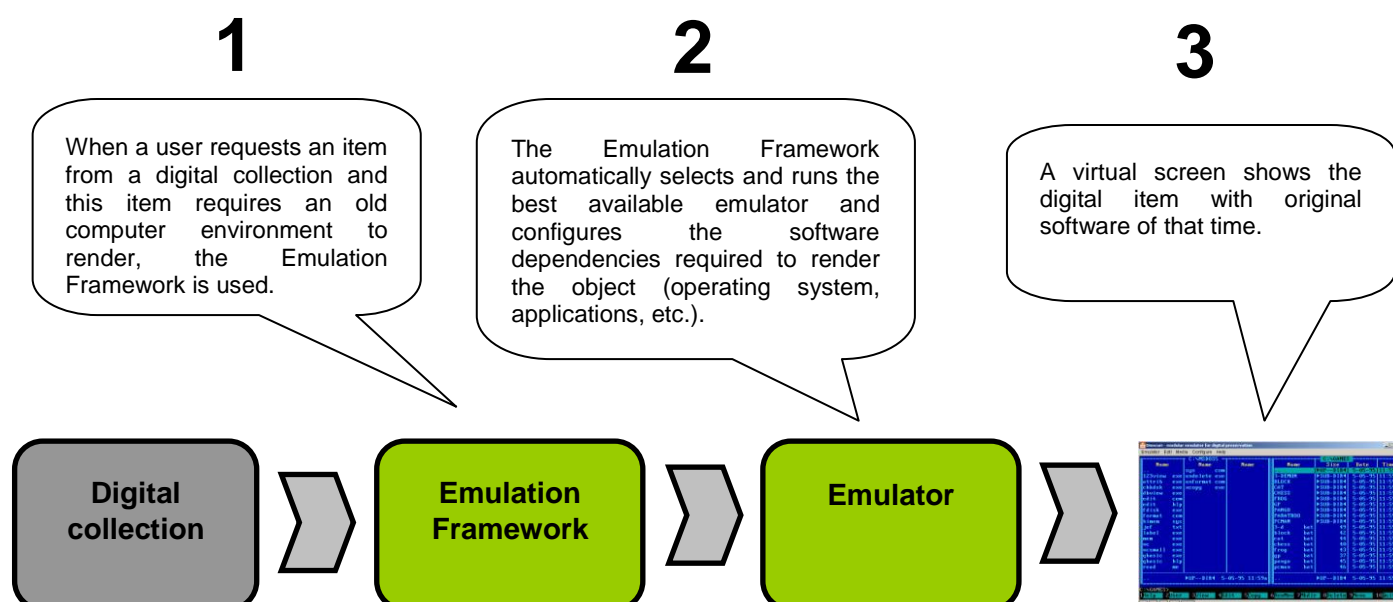
Revisions

Version	Date	Author	Changes
0.1	22-05-2011	Bram Lohman	Initial version
0.2	24-05-2011	Jeffrey van der Hoeven	Revised text
1.0	30-05-2011	Bram Lohman	Incorporated review changes

Executive Summary

This document 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 in their native environment. This 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.

**List of Related Documents**

User Requirements Document [URD]	Requirements for the Emulation Framework
Architectural Design Document [ADD]	Design for the Emulation Framework
System Maintenance Guide [SMG]	Maintenance guide for the Emulation Framework

Abbreviations

Emulation Framework	EF
Graphical User Interface	GUI
Java Runtime Environment	JRE
Keeping Emulation Environments Portable	KEEP

Table of Contents

Executive Summary	3
Table of Contents	5
1 Introduction	6
1.1 What is the Emulation Framework?	6
1.2 About the KEEP project	7
1.3 About the software in this release	7
2 End-user guide	8
2.1 Starting the Emulation Framework	8
2.2 Choosing a file or application	9
2.3 Automatic mode	10
2.4 Manual mode	11
2.4.1 Characterisation	11
2.4.2 Finding pathways	11
2.4.3 Emulator selection	12
2.4.4 Software selection	13
2.4.5 Environment configuration	14
3 Administrator guide	16
3.1 Installation	16
3.1.1 System requirements	16
3.1.2 Installation instructions	16
3.2 Uninstall	22
3.3 Emulation Framework configuration	22
3.3.1 Core configuration	23
3.3.2 Core embedded database management	24
3.3.3 Emulator Archive configuration	26
3.3.4 Emulator Archive database population	27
3.3.5 Software Archive configuration	27
3.3.6 Software Archive database population	28
4 Problems	31

1 Introduction

This System User Guide (SUG) describes how to use the Emulation Framework (EF) developed by the KEEP project¹. The document offers support to **end-users** and **administrators**.

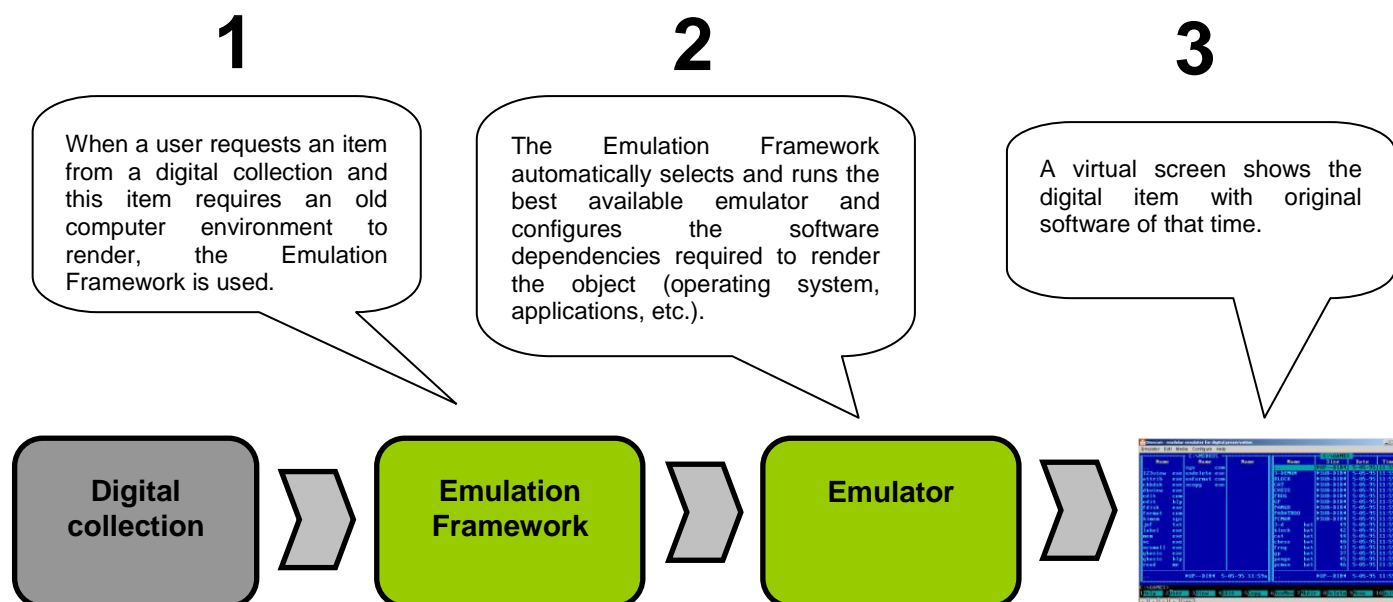
If you are a library visitor, librarian, archivist or other kind of user you are recommended to read the chapter **End-user guide**. This guide will explain how to use the EF and will give you insight into all features available.

If you are the administrator that would like to install or change the EF in a computer environment, please follow the chapter **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 games 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 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 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: a **Core**, **Software Archive** and **Emulator Archive**. The Core EF is the technical heart of the system, performing the automatic characterisation 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 interacts with external services such as technical registries containing file format classifications, the Software Archive that contains software captured in predefined disk images and the Emulator Archive that contains the emulators available for the EF.

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 GUI is developed by the National Library of the Netherlands.

This document was written for version 1.0 (May 2011) of the Emulation Framework.

2 End-user guide

The Emulation Framework consists of three separate components: The Core (and its Graphical User Interface); the Emulator Archive that provides the EF with emulator packages that contain emulators it can run; and the Software Archive that provides the EF with software packages that contain the software required by the emulators.

As an end-user, you will only deal with the GUI and Core directly; your system administrator has installed the Emulator and Software Archives for you and set up the connection to them.

2.1 Starting the Emulation Framework

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

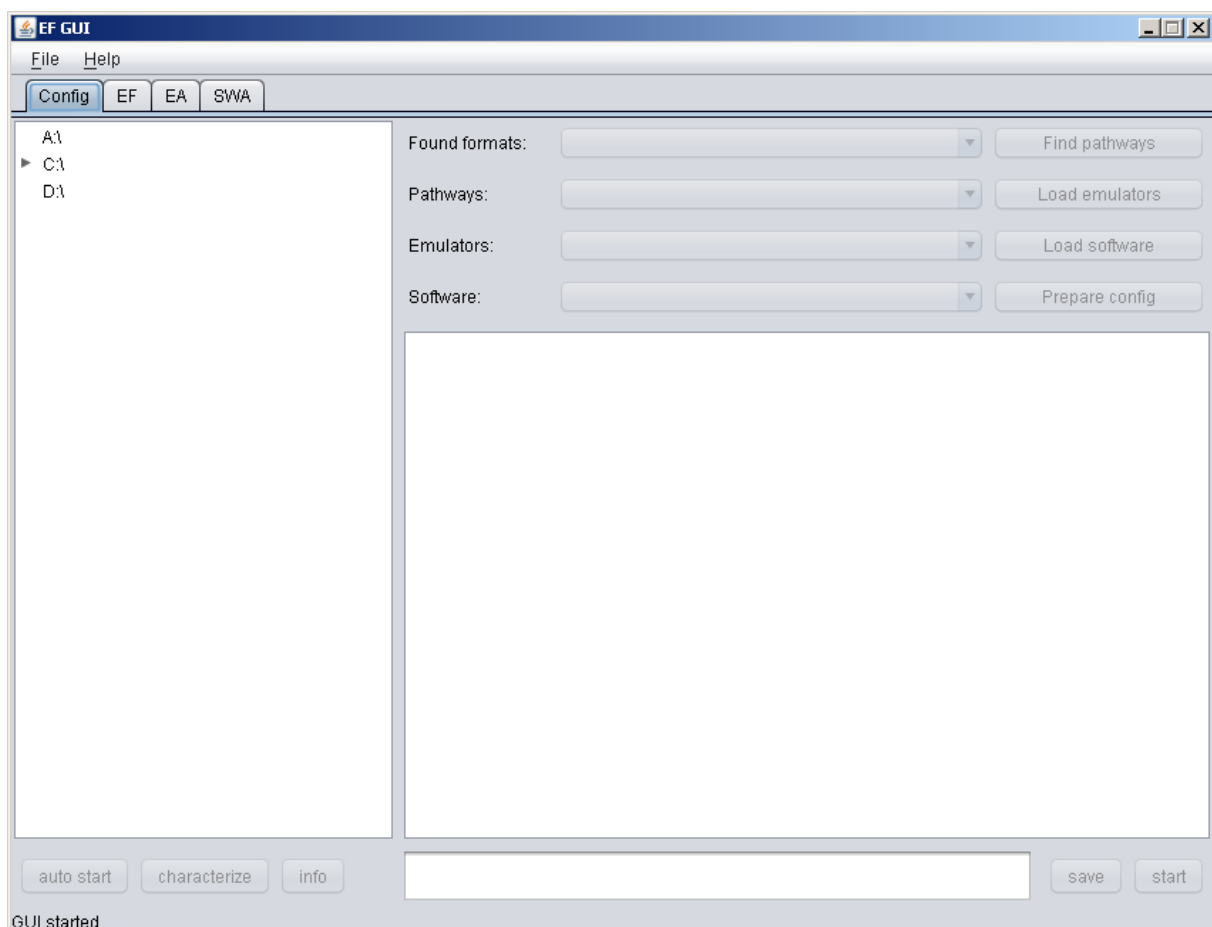
`runEF.bat` (for Windows platforms)

or

`runEF.sh` (for Linux platforms)

Note: your administrator should have set up the system so it can connect to the Software and Emulator Archives automatically.

The following window should appear:



At the top of the window various tabs are shown which address particular parts of the EF.

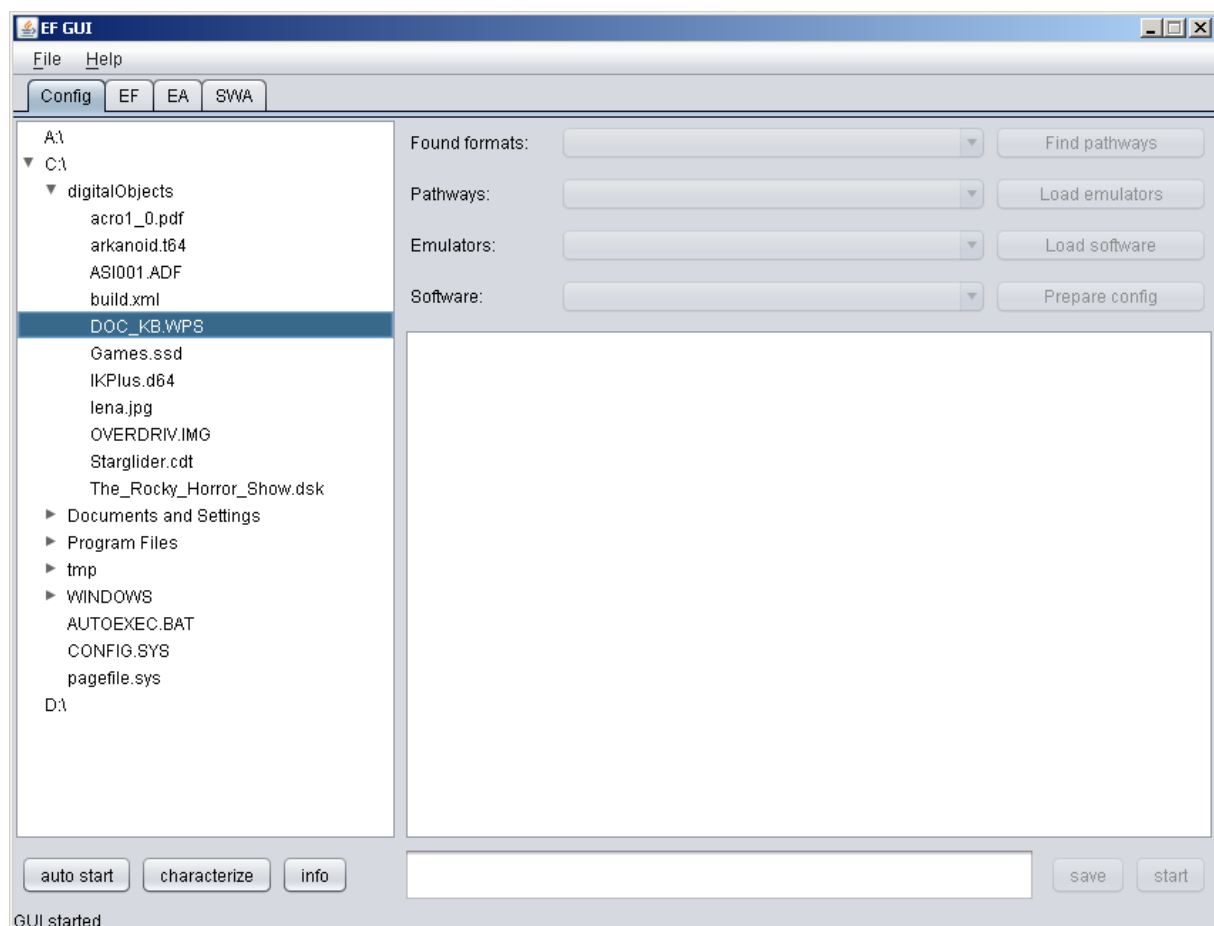
- **Config (default):** shows a file chooser which lets you select a file from your local or network drives and lets you initiate an emulation process automatically or manually.
- **EF:** shows configuration settings of the Emulation Framework.
- **EA (Emulator Archive):** shows the contents and structure of the emulators contained in the Emulator Archive.
- **SWA (Software Archive):** shows the contents and structure of the software contained in the Software Archive.

At the bottom of the window, a status bar is visible. All important status information will be shown here.

Note: not all tabs may be visible depending on the configuration of the EF.

2.2 Choosing a file or application

Click on the tree in the left pane of the Config tab to select a file. By expanding the nodes of the tree you can browse the directories. The following screen shows a file selected:

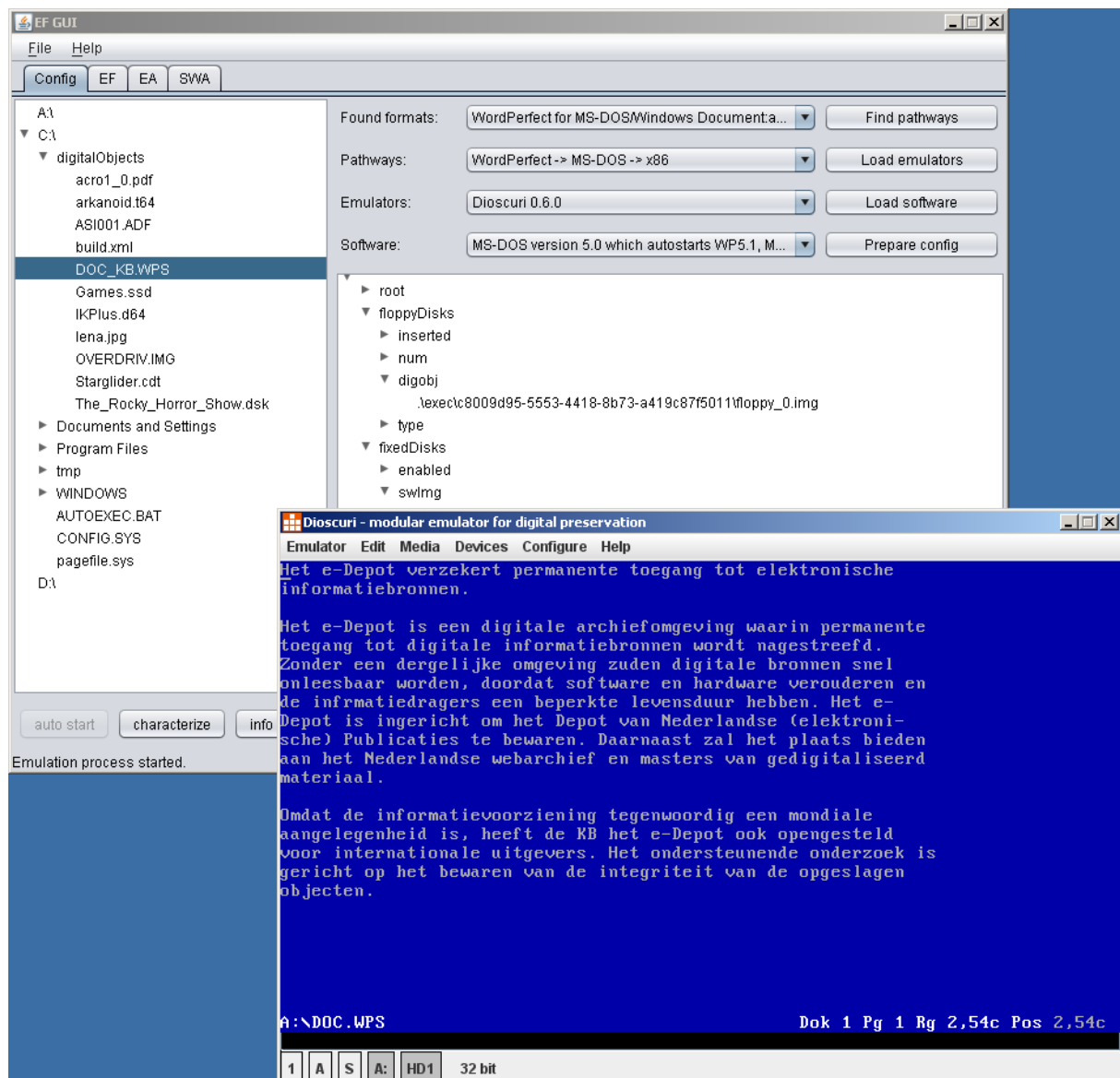


At the bottom of the screen three buttons are enabled. At this point you can choose between the automatic mode by clicking the **auto start** button, and manual mode by clicking the **characterise** button. These two modes are explained in detail in the following sections.

2.3 Automatic mode

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 process depends on the correctness of the characterisation and the availability of emulators and software.

Once a file has been selected, the option to **auto start** is enabled. When this button is pressed, the EF will automatically try to render the selected digital object. This process may take a few seconds, and when it is successful it should start an automatically selected emulator, as shown in the following screen:



Note: not all information in the right pane (formats, pathways, etc.) may be filled in; this is a result of the manual mode explained in the next section.

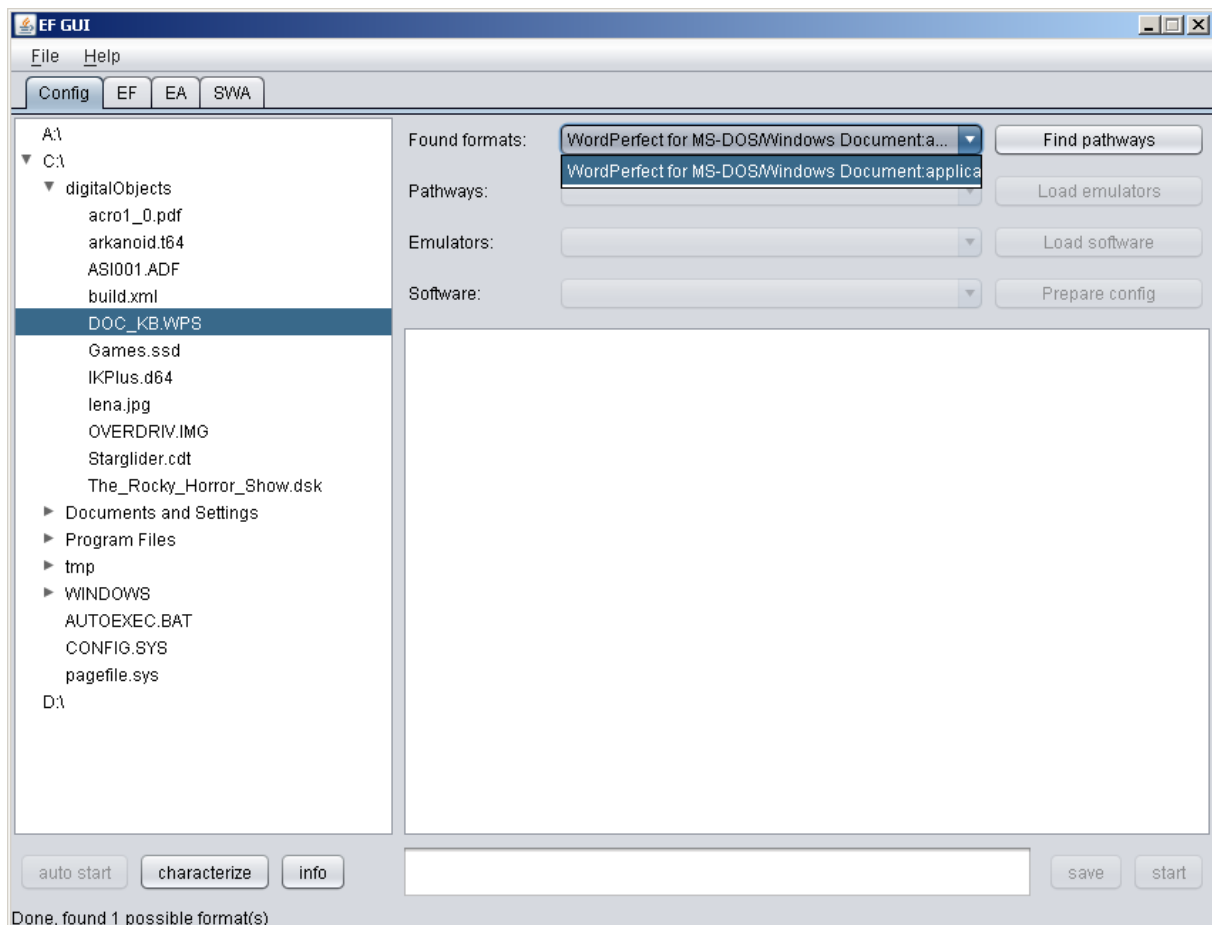
When finished the emulator can be closed by the user and the emulation process ends automatically.

2.4 Manual mode

2.4.1 Characterisation

Once a digital object is selected, the **characterise** button will be enabled. Pressing this button starts the manual mode. The first step is the characterisation of the digital object. Characterisation is the determination of the type of format the digital object has. This forms the basis of finding the pathways that can render the digital object.

After a few seconds, the top slot of the right pane should contain the results of the characterisation process, as shown in the following figure:

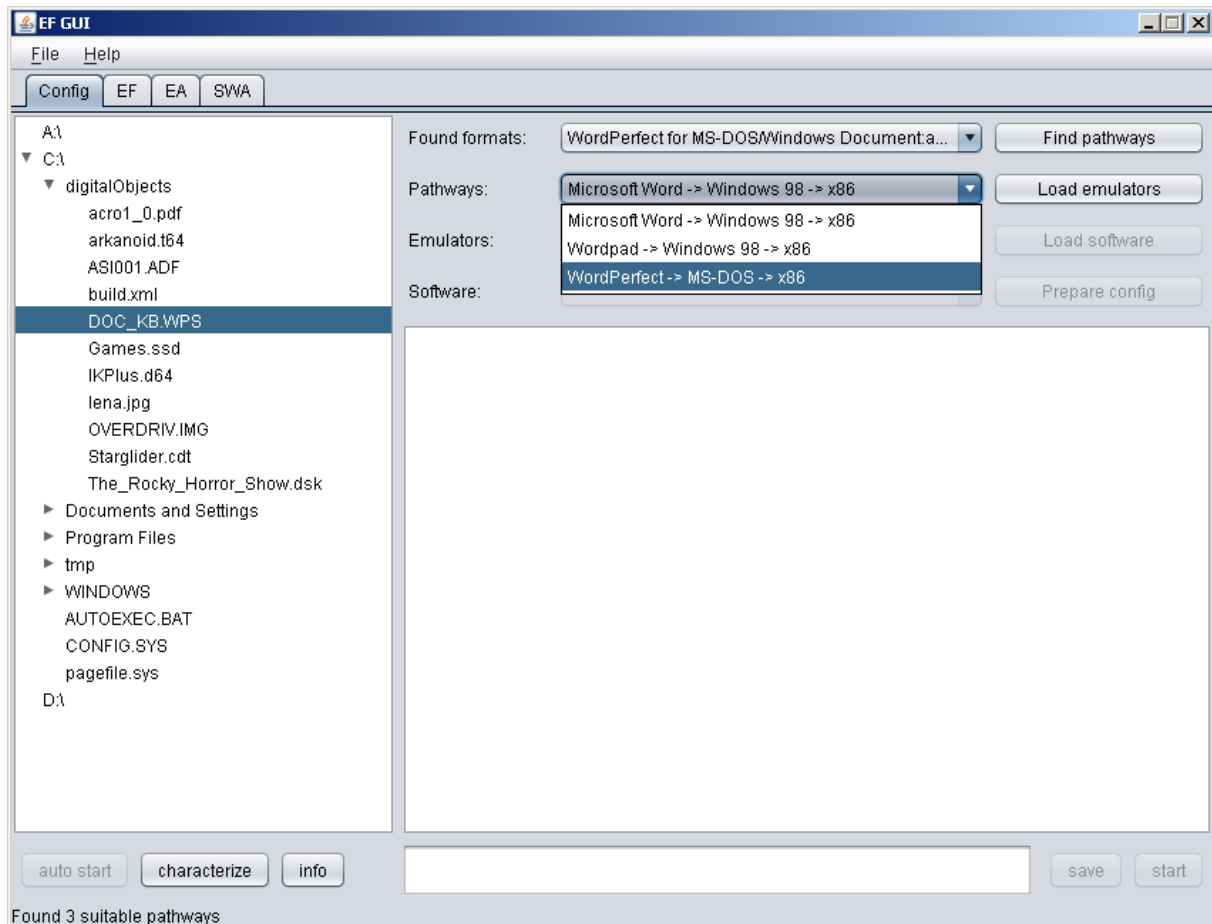


In this example, the EF has found a single result, and it is automatically selected. The following step is to find possible pathways for this file format. If no formats are found, the EF has not been able to successfully identify the file, and the process cannot continue from this point.

2.4.2 Finding pathways

Once a file format has been found, the **Find pathways** button becomes enabled. A pathway is a computer environment consisting of an application, operating system and hardware platform. Pressing this button will make the EF contact an appropriate registry (the default is a basic technical registry that is part of the Software Archive) to find pathways that can render this file format.

An example of the pathways 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) and x86 (platform)

Note: if no pathways can be found, a message will be displayed at the bottom of the screen, where it currently says '*Found 3 suitable pathways*'. Sometimes, the field for application is defined as N/A. In this case, no application is required to run the digital object (e.g. the digital object is a computer game depending only on operating system and hardware).

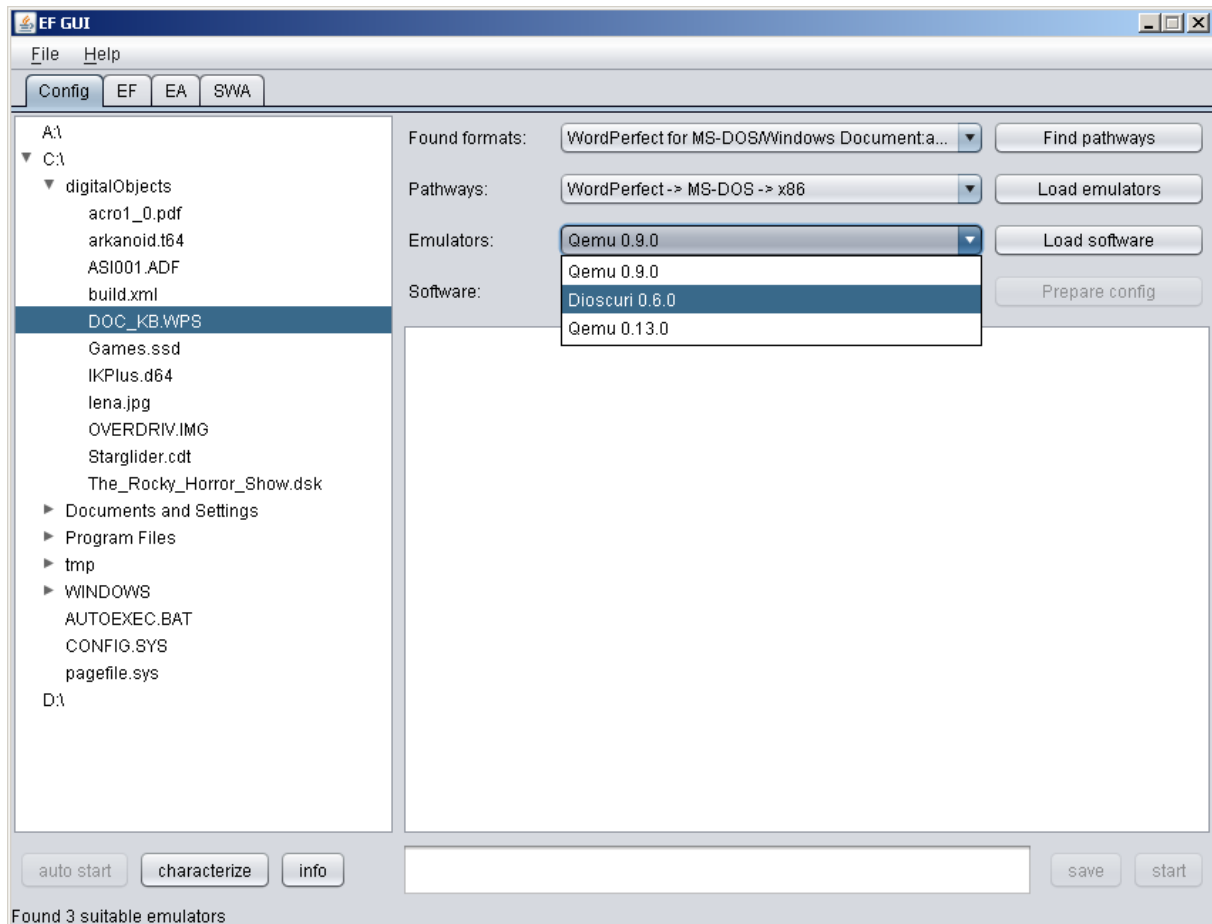
The next step in the process is to load a selection of emulators that support the selected pathway.

2.4.3 Emulator selection

Once a pathway has been selected, the Emulator Archive can be queried for emulators that can run the software that has been selected as part of the pathway. Pressing the **Load emulators** button will start this process.

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

A list of valid results is shown in the following screenshot:

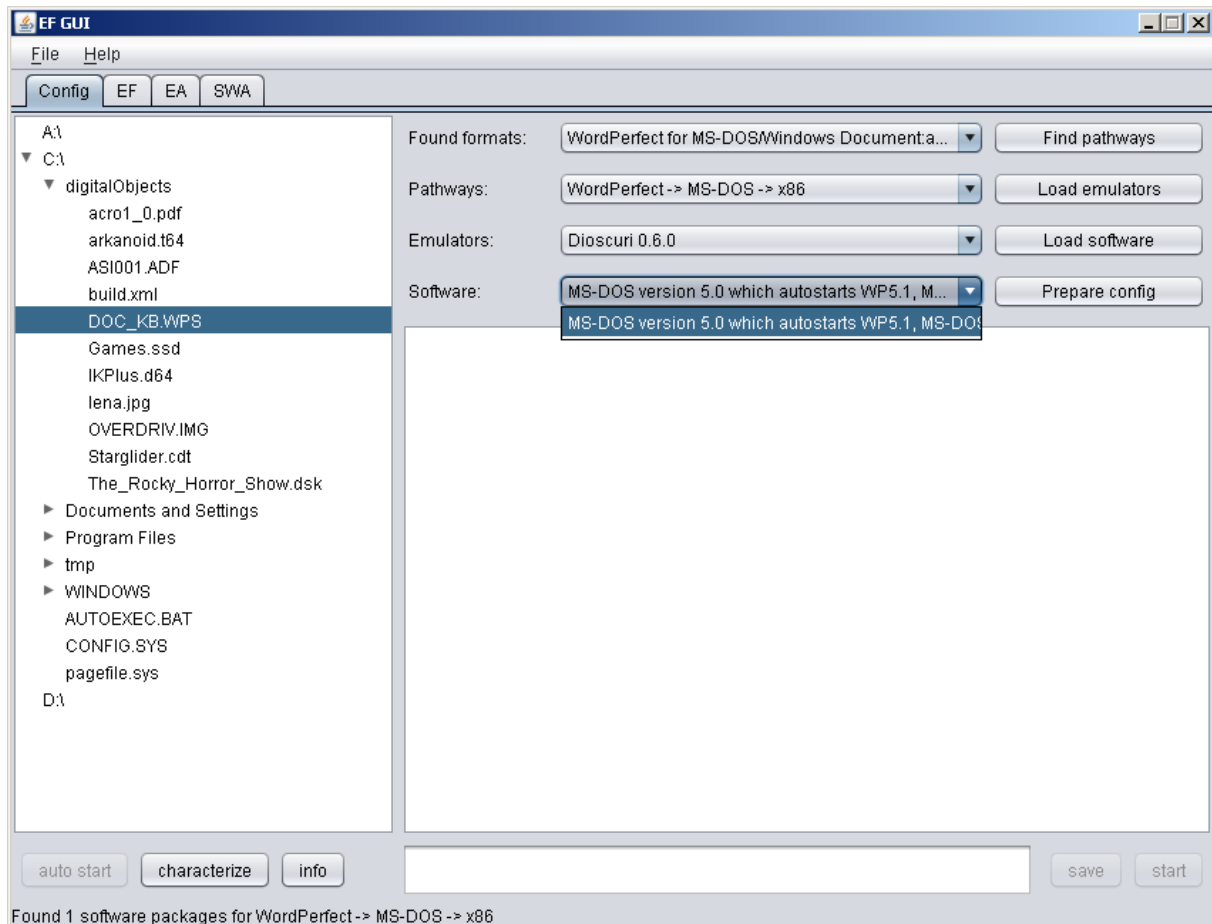


In this case, three emulators have been found and one of these, Dioscuri, is being selected. Once this selection is made the next step in the process can be started: the selection of the required software.

2.4.4 Software selection

Once a valid emulator has been selected, the required software that is part of the pathway needs to be selected and downloaded. The EF will query the Software Archive for software images (pre-installed operating system and applications) that may match the required software stack of the pathway. To start this process, the button **Load software** must be pressed.

Similar to querying the Emulator Archive either a warning at the bottom of the screen will inform the user that no software can be found, or the valid software will be selectable in the result box, as can be seen in the following screenshot:

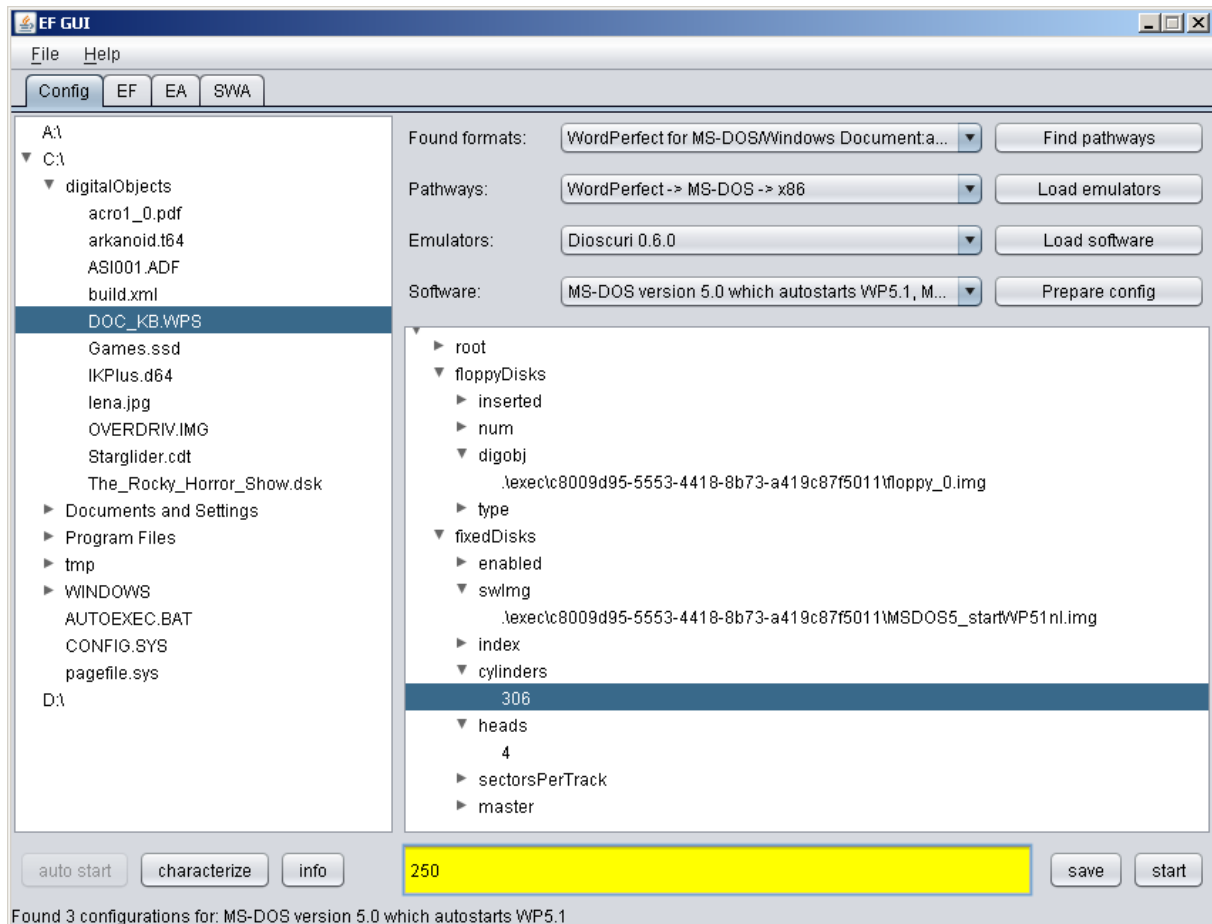


This example has returned one software package that can supply the required operating system (MS-DOS) and application (WordPerfect).

Now that the environment has been completely selected, all that remains is to start the selected emulator and software image to render the pathway the EF has made predefined settings for the emulation environment. The user can accept these settings and directly start the process by clicking the **start** button or check and alter the settings manually.

2.4.5 Environment configuration

The last manual step before starting the environment is to configure the emulator parameters. By pressing the 'Prepare config' button, a default configuration is generated by the EF, which is shown in the bottom right panel. A sample such configuration is as follows:



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.

As can be seen in the example, 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 expert users should change parameters in the configuration; entering the wrong parameters may stop the environment from rendering 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 object. At this point, the result will be similar to the figure shown in section 2.3.



3 Administrator guide

This guide is intended for administrators that need to deploy the EF on a server or local computer, configure the components, and are responsible for creating and updating the contents of the EF, 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.

3.1.1 System requirements

Before you install the EF software, please make sure your computer meets the following minimum system requirements:

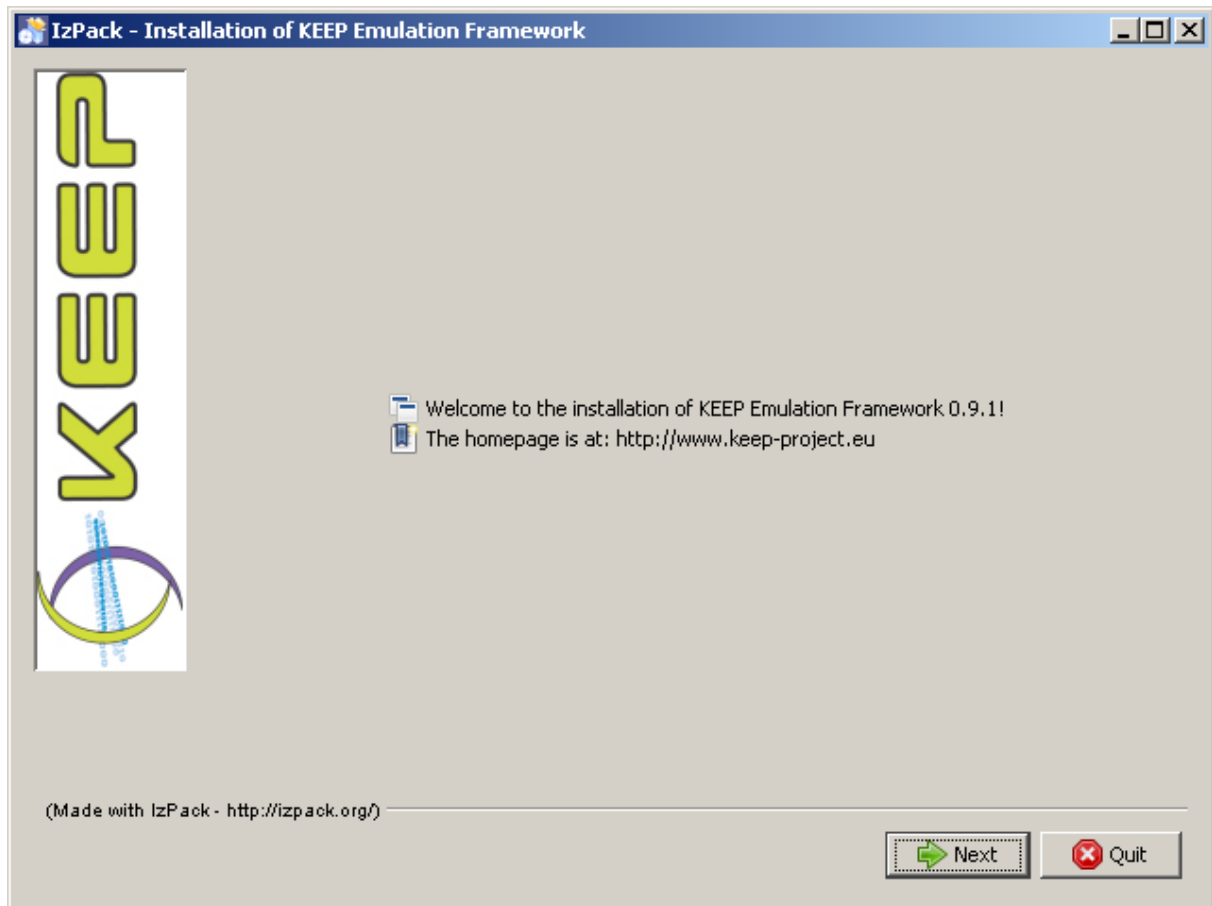
Processor	x86 32/64 bit 1.5 GHz or faster
Memory	at least 2GB of memory
Disk space	200 Megabytes of free available space for the base install Depending on the number of emulators and software images, from 1 GB upwards
Operating system	Linux or Windows, with a compatible JRE and network support
Java Runtime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

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

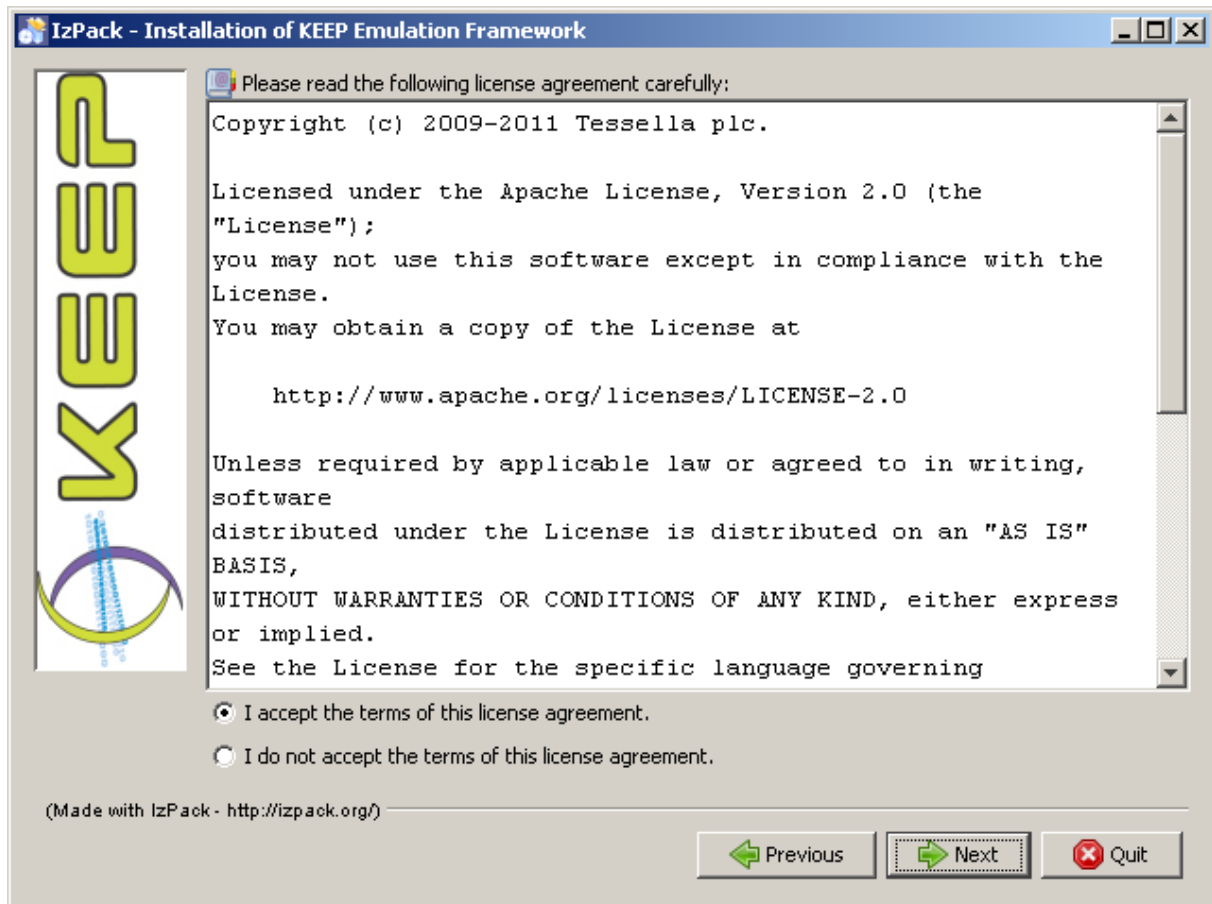
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

3.1.2 Installation instructions

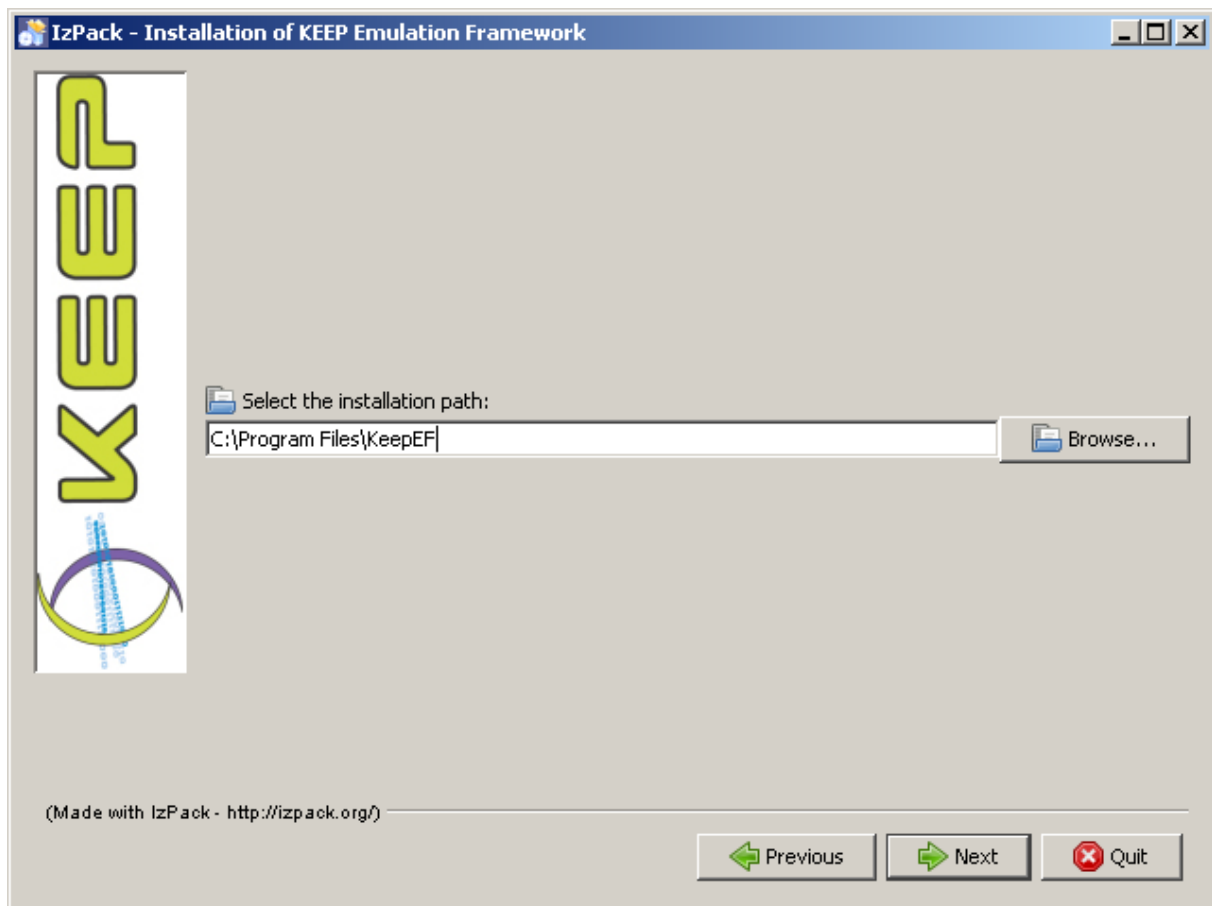
After you have downloaded the installation package, start the JAR-file by double clicking or starting it via the command prompt.



A welcome screen is displayed which contains a brief introduction information about the project and EF. Click **Next**.



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**.

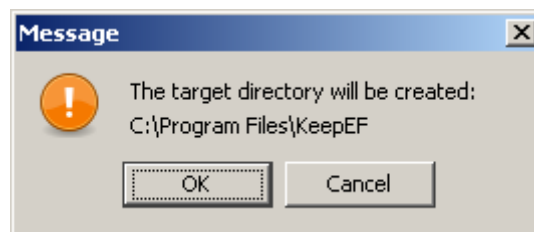


Once the license agreement is accepted, the user is prompted for a location to install the software. The default is set to:

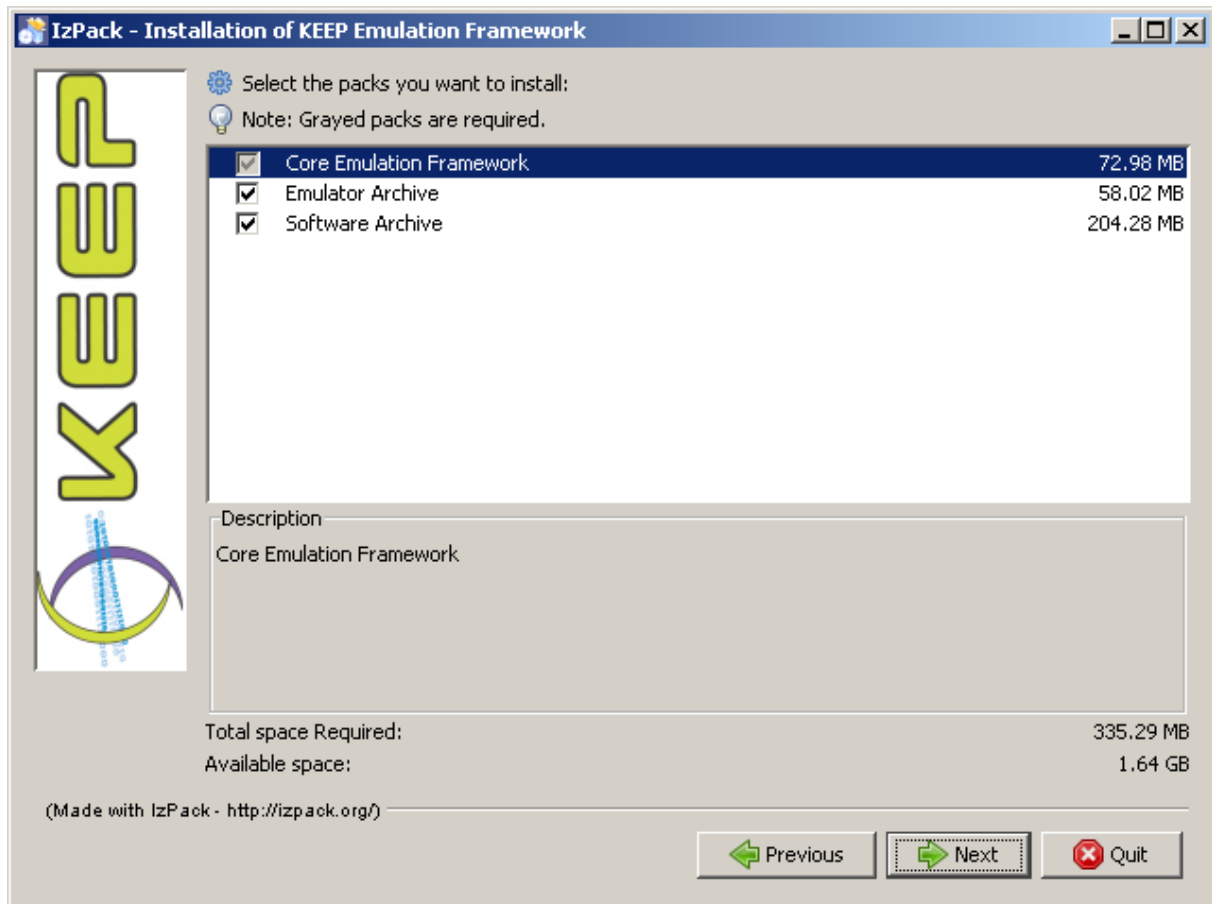
C:\Program Files\KeepEF (MS Windows)

~/KeepEF (Linux)

Click **Next**.



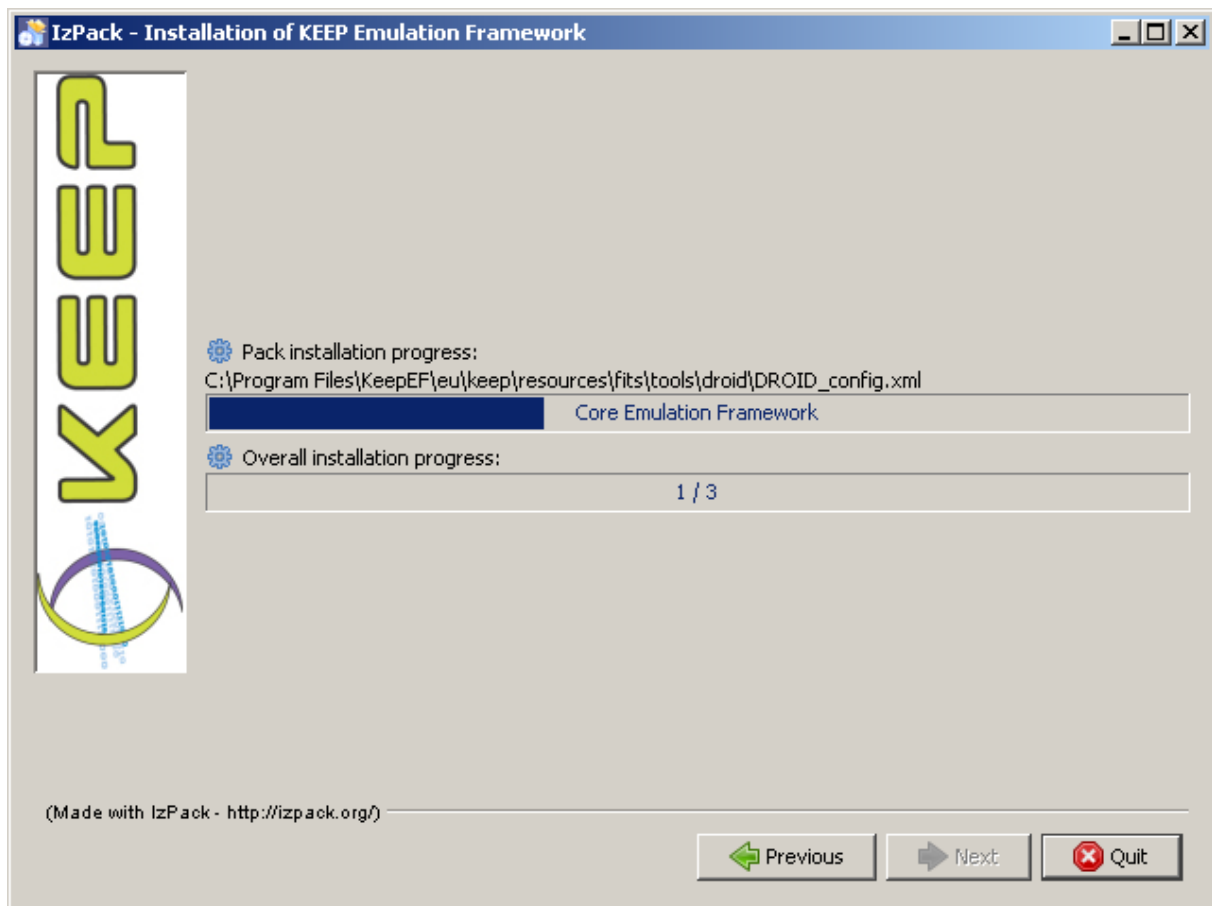
If the directory specified doesn't exist, the user will be asked for a confirmation to create that directory.



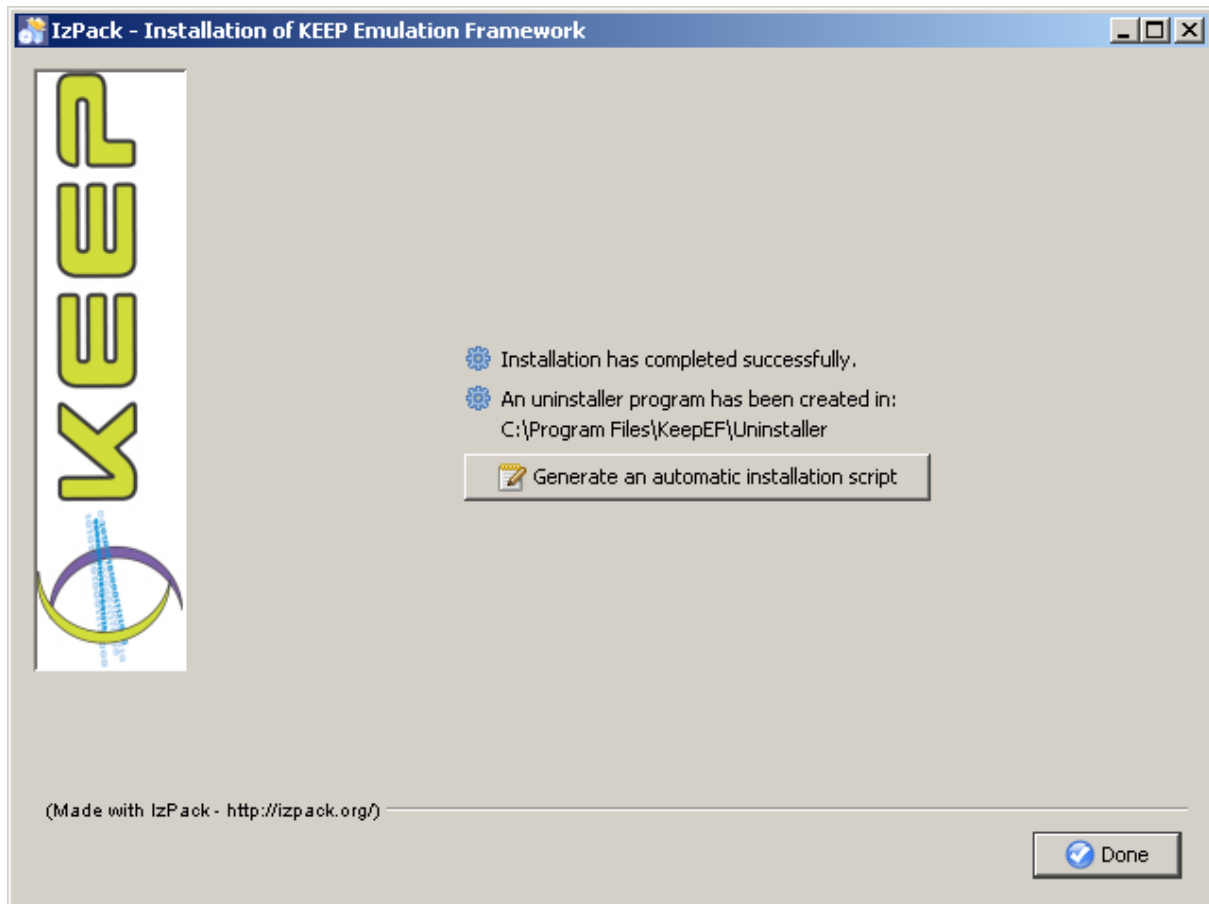
The next screen displays a list of the available components to be installed for the EF. The Core Emulation Framework is the only required component and thus always selected. The Emulator Archive and Software Archive can optionally be selected.

Note: Although the Emulator Archive and Software Archive are optional, the Core still requires these to function correctly. However, if these are available remotely (e.g. on a local LAN or WAN), then there is no need for a local installation.

Click **Next**.



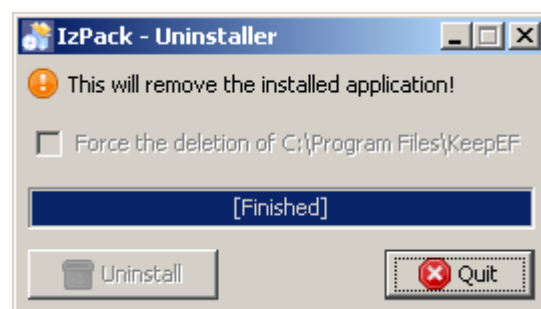
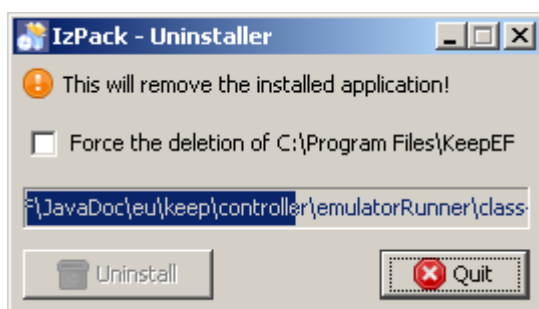
Finally, a progress of the installation process is shown for each component and the overall system. Installation might take a couple of minutes depending on your computer specifications.



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 program 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 Emulation Framework configuration

The three EF components can all be separately configured by editing a text file. Although each text file configures each component separately, there are some parameters that need

to match between them for the Core to be able to connect to the Software and Emulator Archive successfully. The details for each configuration file are outlined in the next sections.

3.3.1 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 if it doesn't exist (FALSE), or fail if it doesn't exist (TRUE)
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	http://localhost:9000/softwarearchive/	Location of the Software Archive
emulator.archive.url	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)
tmp.dir	./tmp	Directory used by the host system to store temporary files

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

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 access level, 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 if it doesn't exist (FALSE), or fail if it doesn't exist (TRUE)
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.userassw	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 if it doesn't exists (FALSE), or fail if it doesn't exist (TRUE)
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.userassw	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 if it doesn't exists (FALSE), or fail if it doesn't exist (TRUE)
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.userassw	swa	Database user password

3.3.2 Core embedded database management

An embedded (internal) database is used in the Core Emulation Framework for managing metadata and files for the various emulators and registries. For example, if you wish to change the default technical registry for file format characterisation, you can do so by changing the settings in this embedded database.

An administrator can directly access this database (without the Emulation Framework) by using the H2 database engine web console (available at <http://www.h2database.com>).

An administrator can also access the database via the GUI, on the EF tab. Simple database record insertion and deletion is supported.

English ▼ Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded) ▼

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:~/h2/EF_engine

User Name: cef

Password:

Connect Test Connection

Test successful

By simply pointing the JDBC URL (in blue) to the location of the database, specified in the property **h2.db.url** of the **user.properties** file, the connection is set up. The 'Test Connection' button can be used to check if the user name and password, also specified in the **user.properties** file, as well as connection details are correct.

English ▼ Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded) ▼

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:~/h2/EF_engine

User Name: cef

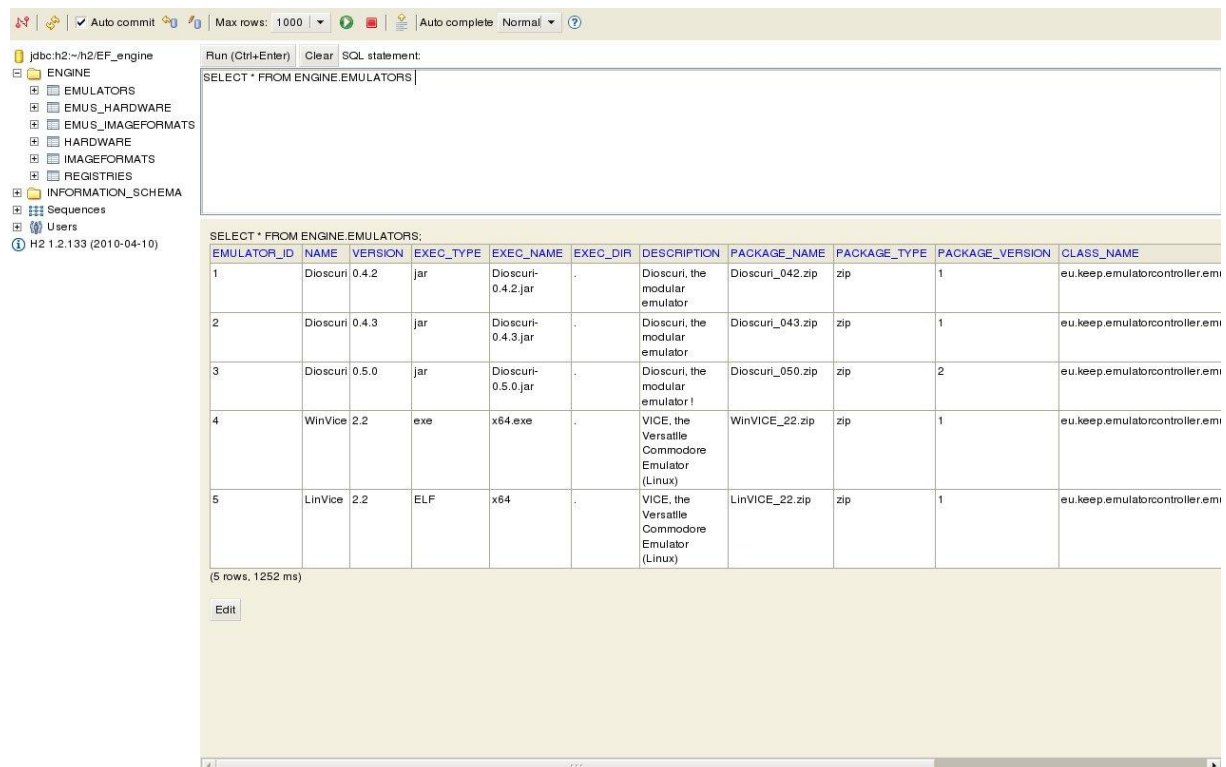
Password:

Connect Test Connection

Database may be already in use: "Locked by another process". Possible solutions: close all other connection(s); use the server mode; SQL statement: null/12a47d993675c6c9dffacc009b549e970dbda7d0bc1 [90020-133] 90020/90020 (Help)

If a connection to the database is already open, an error message will appear (as expected for all embedded databases where only one connection can be established at a time)

Once the connection is established, you can navigate between the various schemas/tables of the database and execute SQL statements directly.



Run (Ctrl+Enter) | Clear | SQL statement:

```
SELECT * FROM ENGINE.EMULATORS
```

SELECT * FROM ENGINE.EMULATORS;

EMULATOR_ID	NAME	VERSION	EXEC_TYPE	EXEC_NAME	EXEC_DIR	DESCRIPTION	PACKAGE_NAME	PACKAGE_TYPE	PACKAGE_VERSION	CLASS_NAME
1	Dioscuri	0.4.2	jar	Dioscuri-0.4.2.jar	.	Dioscuri, the modular emulator	Dioscuri_042.zip	zip	1	eu.keep.emulatorcontroller.em
2	Dioscuri	0.4.3	jar	Dioscuri-0.4.3.jar	.	Dioscuri, the modular emulator	Dioscuri_043.zip	zip	1	eu.keep.emulatorcontroller.em
3	Dioscuri	0.5.0	jar	Dioscuri-0.5.0.jar	.	Dioscuri, the modular emulator	Dioscuri_050.zip	zip	2	eu.keep.emulatorcontroller.em
4	WinVice	2.2	exe	x64.exe	.	VICE, the Versatile Commodore Emulator (Linux)	WinVICE_22.zip	zip	1	eu.keep.emulatorcontroller.em
5	LinVice	2.2	ELF	x64	.	VICE, the Versatile Commodore Emulator (Linux)	LinVICE_22.zip	zip	1	eu.keep.emulatorcontroller.em

(5 rows, 1252 ms)

Edit

One table that needs to be populated for the Core to function correctly is the EMULATOR_WHITELIST table. This table contains the IDs of emulators that are allowed to be run. For more details, please see the Architectural Design Document.

3.3.3 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
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 if it doesn't exist (FALSE), or fail if it doesn't exist (TRUE)
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
server.soap.address	http://localhost:9001/emulatorarchive/	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.

3.3.4 Emulator Archive database population

Although the Emulator Archive contains a number of default emulators, it can be desirable to insert other emulators. To successfully do this, a number of parameters relating to the emulator must be set correctly. The following example demonstrates this process

1. Add hardware
Add a new hardware id and name to the HARDWARE table, if necessary. Note: This hardware must also be available in the Software Archive
2. Add image format
Add a new image format id and name to the IMAGEFORMATS table, if necessary
3. Add emulator
Add a new emulator id, name, version, etc. to the EMULATORS table. See the Architectural Design Document for a full description of the columns
4. Link the emulator to the hardware
Add the emulator id and hardware id link in the EMUS_HARDWARE table, indicating the emulator runs on the specified hardware
5. Link the emulator to the image format
Add the emulator id and image format id link in the EMUS_IMAGEFORMATS table, indicating the emulator runs on the specified hardware. Note: the emulator may support more than one image format.

An administrator can access the Emulator Archive database via the GUI, on the EA tab. Simple database record insertion and deletion is supported. Note that due to the size of some of the emulators, BLOBs are not shown in the GUI, and cannot be inserted. To successfully insert these records, it is suggested to use the H2 web interface (see section 3.3.2)

3.3.5 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 if it doesn't exist (FALSE), or fail if it doesn't exist (TRUE)
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.userassw	swa	Database user password
server.soap.address	http://localhost:9000/softwarearchive/	Server address

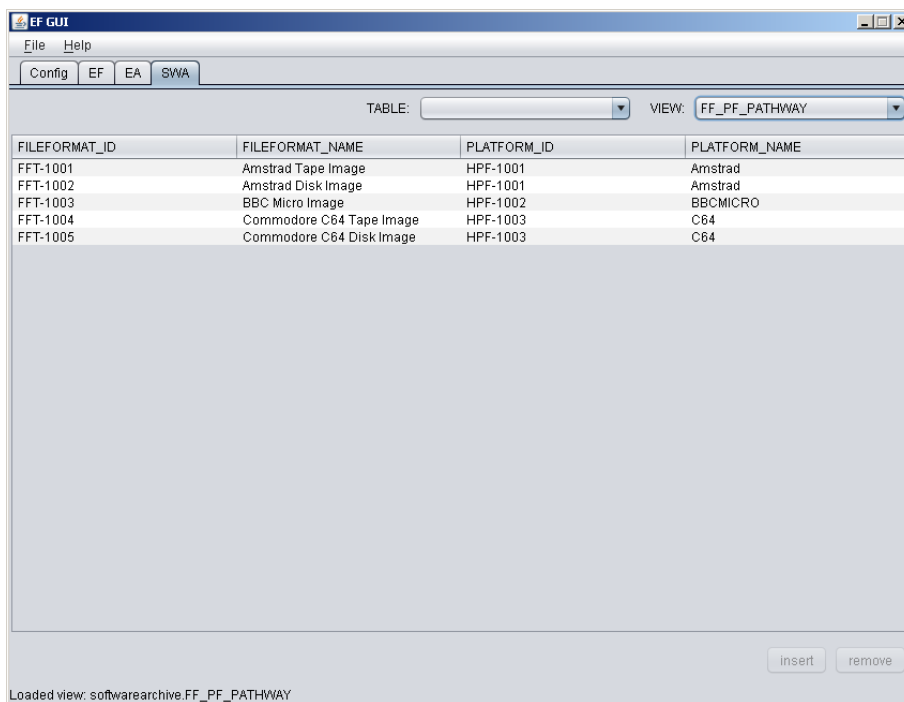
Note that the `server.soap.address` property must match the `EF software.archive.url` property for the connection between the two components to be successful.

3.3.6 Software Archive database population

Although the Software Archive contains a number of default software images, it can be desirable to insert others. To successfully do this, a number of parameters relating to the software image must be set correctly, plus a supporting pathway must be created. The following example demonstrates this process

1. Add file format
Add a new file format id and name to the `FILEFORMATS` table, if necessary. Note: This file format must also be available in the Core database.
2. Add application
Add a new application id and name to the `APPS` table, if necessary.
3. Add operating system
Add a new operating system id and name to the `OP_SYS` table, if necessary.
4. Add platform
Add a new platform id and name to the `PLATFORMS` table, if necessary.
5. Add an image format
Add a new image format id and name to the `IMAGEFORMATS` table, if necessary. This image format must also be available in the EA database.
6. Add a software image
Add a software image id, name, image format id and platform id to the `IMAGES` table. Add the image file to the `IMAGEBLOBS` table.
7. Create the pathway
 - a. Link the file format to an application, operating system and/or platform
Depending on the pathway, a file format must either be run in an application (e.g. a WordPerfect file runs in the WordPerfect application), operating system (e.g. an ISO file is run directly by Windows) or a hardware platform (e.g. a C64 file runs directly on the Commodore 64). In the appropriate table (`FILEFORMATS_APPS`, `FILEFORMATS_OPSYS`, `FILEFORMATS_PLATFORM`), link the file format to the appropriate level in the pathway.
 - b. Link the application to an operating system and software image
An application requires an operating system to run, and the software image needs to know what applications are contained in it. The appropriate links between the application and operating system/software image need to be made in the `APPS_OPSYS/APPS_IMAGES` tables, respectively.
 - c. Link the operating system to a platform and software image
An operating system runs on a platform, and the software image needs to know what operating systems are contained in it. The appropriate links between the operating system and platform/software image need to be made in the `OPSYS_PLATFORMS/OPSYS_IMAGES` tables, respectively.
8. Check the pathway is correctly generated
Using the `PATHWAYS` view, the pathway generated in step 7 should be visible. This view contains all pathways, be they file format to platform; file format to operating system to platform; or file format to application to operating system to platform. For those pathways that do not have all the parts of the software stack, the appropriate columns will be null.

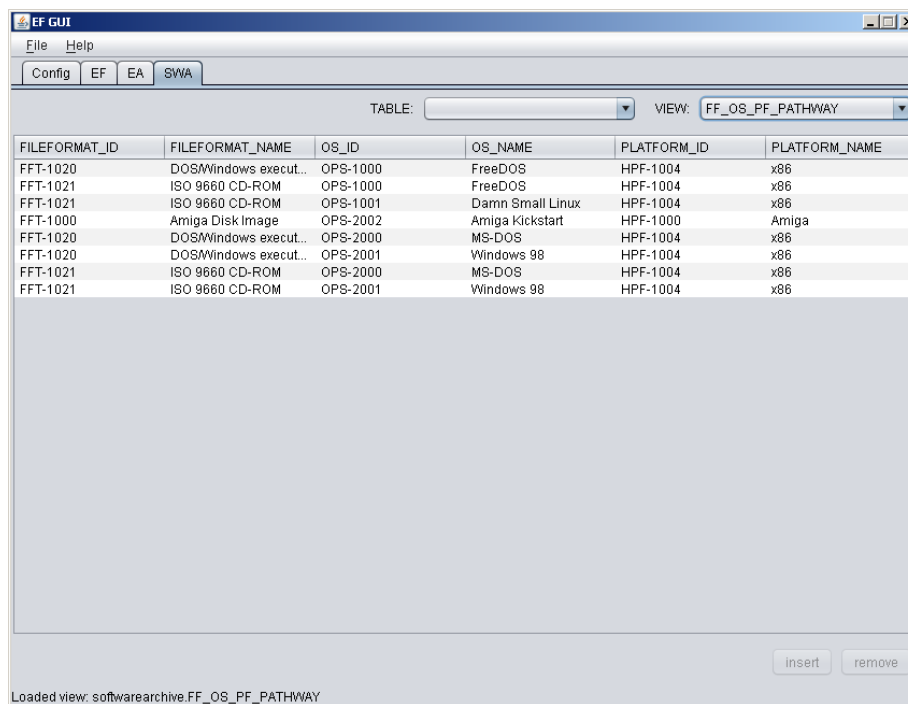
Example of file formats to platform pathways:



Loaded view: softwarearchive.FF_PF_PATHWAY

FILEFORMAT_ID	FILEFORMAT_NAME	PLATFORM_ID	PLATFORM_NAME
FFT-1001	Amstrad Tape Image	HPF-1001	Amstrad
FFT-1002	Amstrad Disk Image	HPF-1001	Amstrad
FFT-1003	BBC Micro Image	HPF-1002	BBCMICRO
FFT-1004	Commodore C64 Tape Image	HPF-1003	C64
FFT-1005	Commodore C64 Disk Image	HPF-1003	C64

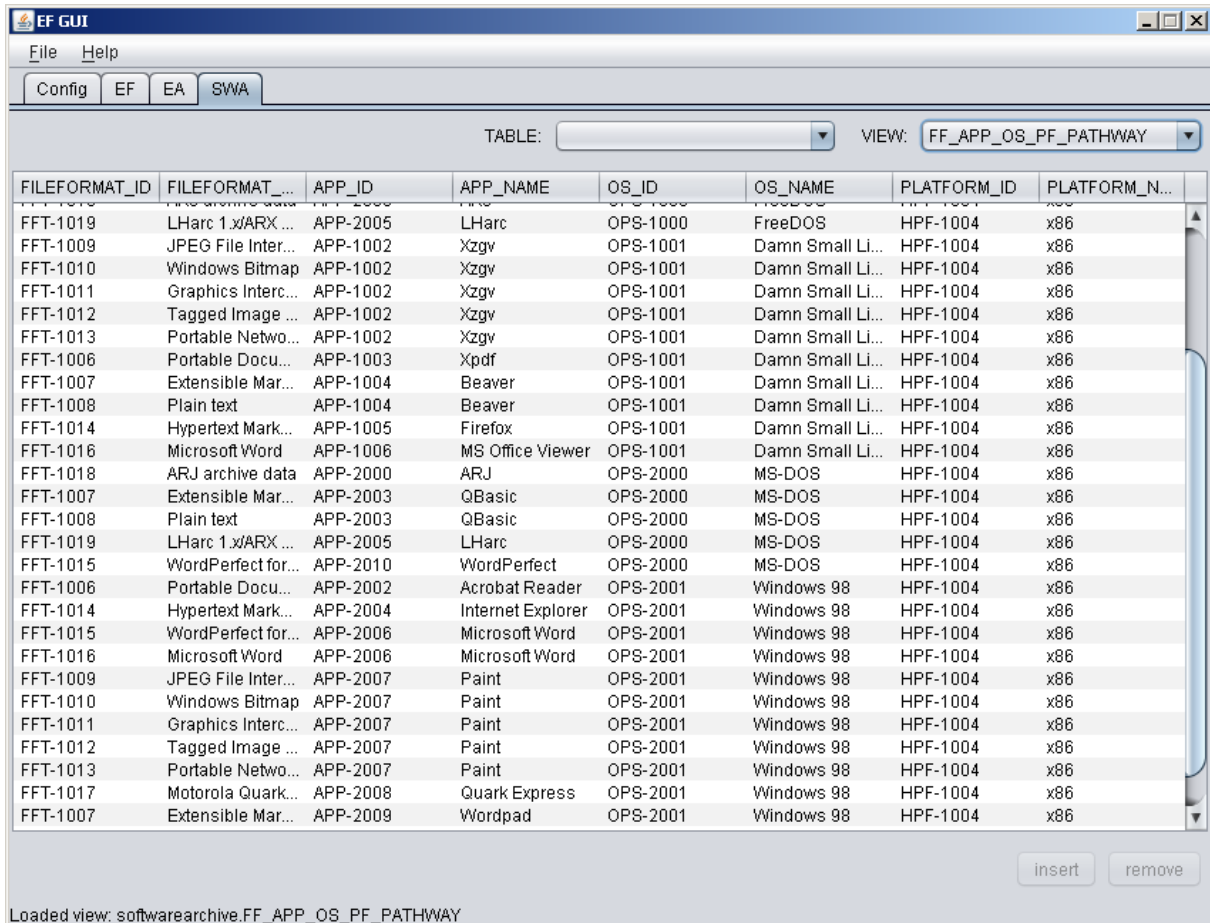
Example of file formats to operating system to platform pathways:



Loaded view: softwarearchive.FF_OS_PF_PATHWAY

FILEFORMAT_ID	FILEFORMAT_NAME	OS_ID	OS_NAME	PLATFORM_ID	PLATFORM_NAME
FFT-1020	DOS/Windows execut...	OPS-1000	FreeDOS	HPF-1004	x86
FFT-1021	ISO 9660 CD-ROM	OPS-1000	FreeDOS	HPF-1004	x86
FFT-1021	ISO 9660 CD-ROM	OPS-1001	Damn Small Linux	HPF-1004	x86
FFT-1000	Amiga Disk Image	OPS-2002	Amiga Kickstart	HPF-1000	Amiga
FFT-1020	DOS/Windows execut...	OPS-2000	MS-DOS	HPF-1004	x86
FFT-1020	DOS/Windows execut...	OPS-2001	Windows 98	HPF-1004	x86
FFT-1021	ISO 9660 CD-ROM	OPS-2000	MS-DOS	HPF-1004	x86
FFT-1021	ISO 9660 CD-ROM	OPS-2001	Windows 98	HPF-1004	x86

Example of file formats to applications to operating system to platform pathways:



EF GUI

File Help

Config EF EA SWA

TABLE: VIEW: FF_APP_OS_PF_PATHWAY

FILEFORMAT_ID	FILEFORMAT_NAME	APP_ID	APP_NAME	OS_ID	OS_NAME	PLATFORM_ID	PLATFORM_NAME
FFT-1019	LHarc 1.x/ARX...	APP-2005	LHarc	OPS-1000	FreeDOS	HPF-1004	x86
FFT-1009	JPEG File Inter...	APP-1002	Xzgv	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1010	Windows Bitmap	APP-1002	Xzgv	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1011	Graphics Inter...	APP-1002	Xzgv	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1012	Tagged Image ...	APP-1002	Xzgv	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1013	Portable Netwo...	APP-1002	Xzgv	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1006	Portable Docu...	APP-1003	Xpdf	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1007	Extensible Mar...	APP-1004	Beaver	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1008	Plain text	APP-1004	Beaver	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1014	Hypertext Mark...	APP-1005	Firefox	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1016	Microsoft Word	APP-1006	MS Office Viewer	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1018	ARJ archive data	APP-2000	ARJ	OPS-2000	MS-DOS	HPF-1004	x86
FFT-1007	Extensible Mar...	APP-2003	QBasic	OPS-2000	MS-DOS	HPF-1004	x86
FFT-1008	Plain text	APP-2003	QBasic	OPS-2000	MS-DOS	HPF-1004	x86
FFT-1019	LHarc 1.x/ARX...	APP-2005	LHarc	OPS-2000	MS-DOS	HPF-1004	x86
FFT-1015	WordPerfect for...	APP-2010	WordPerfect	OPS-2000	MS-DOS	HPF-1004	x86
FFT-1006	Portable Docu...	APP-2002	Acrobat Reader	OPS-2001	Windows 98	HPF-1004	x86
FFT-1014	Hypertext Mark...	APP-2004	Internet Explorer	OPS-2001	Windows 98	HPF-1004	x86
FFT-1015	WordPerfect for...	APP-2006	Microsoft Word	OPS-2001	Windows 98	HPF-1004	x86
FFT-1016	Microsoft Word	APP-2006	Microsoft Word	OPS-2001	Windows 98	HPF-1004	x86
FFT-1009	JPEG File Inter...	APP-2007	Paint	OPS-2001	Windows 98	HPF-1004	x86
FFT-1010	Windows Bitmap	APP-2007	Paint	OPS-2001	Windows 98	HPF-1004	x86
FFT-1011	Graphics Inter...	APP-2007	Paint	OPS-2001	Windows 98	HPF-1004	x86
FFT-1012	Tagged Image ...	APP-2007	Paint	OPS-2001	Windows 98	HPF-1004	x86
FFT-1013	Portable Netwo...	APP-2007	Paint	OPS-2001	Windows 98	HPF-1004	x86
FFT-1017	Motorola Quark...	APP-2008	Quark Express	OPS-2001	Windows 98	HPF-1004	x86
FFT-1007	Extensible Mar...	APP-2009	Wordpad	OPS-2001	Windows 98	HPF-1004	x86

Loaded view: softwarearchive.FF_APP_OS_PF_PATHWAY

Insert remove

An administrator can access the Software Archive database via the GUI, on the SWA tab. Simple database record insertion and deletion is supported. Note that due to the size of some of the images, BLOBs are not shown in the GUI, and cannot be inserted. To successfully insert these records, it is suggested to use the H2 web interface (see section 3.3.2)



4 Problems

If you experience any troubles during installation or operation of the EF, please contact the Emulation Framework development team: info@keep-project.eu