



Install Guide

Version 3.1.5222.0

October 16, 2014

Copyright 2005-2014 MyARM GmbH

Copyright © 2005-2014 MyARM GmbH

MyARM GmbH
Altkönigstraße 7
65830 Kriftel
Germany

Web: <http://myarm.com/>
Mail: info@myarm.com

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.

1 Introduction

This document describes the installation of MyARM. First the requirements and environment needed to deploy MyARM for C/C++, Java, C# and Python are described. Followed by a detailed description how to install MyARM on your system and finally setting up the required license key.

1.1 Requirements

The following conditions must be fulfilled to deploy the MyARM agent and you need at least 160 MB (enterprise edition) of free disk space. MyARM supports 32-bit and 64-bit environments. Currently on Windows only a single 32-bit and a single 64-bit installation is supported. On all other platforms 32- and 64-bit applications are supported by the 64-bit release. If you need support for any other operating system, platform, database or compiler feel free to contact us.

1.1.1 Platforms

1. AIX:
 - Power3 or higher.
 - AIX 5.3 TL 10, AIX 6.1
 - 32/64-bit (ppc64) support
2. Linux:
 - Intel i586 or higher CPU (Pentium, AMD), PowerPC G3/G4 or higher.
 - Linux kernel 2.4.x or 2.6.x
 - glibc 2.3.6 or higher.
 - 32bit (ARM, x86) and 32/64-bit (amd64) support
3. Solaris:
 - UltraSparc CPU or Intel i586 or higher CPU (Pentium, AMD).
 - Solaris 2.10 or higher.
 - 32bit (x86), 32/64-bit (amd64, sparc64) support
4. Windows:
 - Intel i586 or higher CPU (Pentium, AMD).
 - Windows XP, Windows 2000, Windows 7.
 - 32bit (x86) and 64-bit (amd64) support

1.1.2 Databases

1. MySQL 5.0-5.5
2. MariaDB 5.5
3. Oracle 10g
4. SQLite3 (file based)

1.1.3 Compiler

1. Any ANSI-C/C++ compliant C/C++ Compiler.
2. Java 1.4 RTE or higher
3. C# Visual Studio 9 or higher and .NET 3.5 for C# or higher on Windows and Mono 2.01 for Linux.

1.2 Naming of MyARM archives

MyARM is only distributed over the net by downloading the appropriate distribution from the MyARM download page: <http://myarm.com/support.html#support-dl>.

The following naming scheme applies to all MyARM distribution archive files under UNIX.

myarm-version-edition-os-arch-type.extension

where the parts set in *italic* are:

version – the version of the MyARM distribution. The MyARM version is build with the following pattern: “<major>.<minor>.<buildid>.<revision>”.

edition – the edition of MyARM. Either *community* for the community edition, *professional* for the professional edition, *enterprise* for the enterprise edition or *client* for the client edition.

os – the operating system the distribution was build for. Currently *aix*, *linux*, *solaris* or *win32* are supported.

arch – the architecture (CPU) the distribution was build for. Currently

- *armv7* for ARMv7 CPU based systems
- *x86* for Intel/AMD 32bit CPU based systems
- *amd64* for Intel/AMD 64bit CPU based systems
- *ppc64* for PowerPC 64bit CPU based systems
- *sparc64* for Sparc 64bit CPU based systems

type – the distribution type. Either *agent* containing a libraries and programs needed to deploy instrumented applications, *tools* containing libraries and programs to analyse measured data, *docs* containing the documentation and manual pages of MyARM programs or *sdk* containing header files and examples how to use MyARM or to instrument applications with ARM. A type of *bin* is an archive containing the all above described archives for easy downloading.

extension – For unix systems this is a simple tar.gz file. Under *win32* the msi package is used.

2 Installing the MyARM distribution on a workstation under UNIX

2.1 Install directory

MyARM was build using the `/opt/myarm` directory and if used shared libraries are found automatically. We recommend to install the MyARM distribution under the directory `/opt` as this is the standard for programs from third-party vendors. The install procedure installs MyARM into a parent directory called `myarm-1.4.2584.0`. All you have to do after an installation is to create a symbolic link from `/opt/myarm` to the current version of MyARM you want to use:

```
# cd /opt
# ln -s myarm-1.4.2584.0 myarm
```

Therefore its easy to change to a new version of MyARM.

2.2 Archive extraction

All files and directories belonging to the MyARM distribution reside under a common parent directory named `myarm-1.4.2584.0`. The MyARM distribution is divided into several parts. Therefore a two step extraction exists:

1. extract the MyARM distribution archive into a temporary directory mostly in your home directory.
2. install the appropriate MyARM parts onto your system using either the provided installation script or manually.

If you have downloaded a MyARM professional edition for linux for Intel 32-bit you have an archive named:

```
myarm-1.4.2584.0-professional-linux-x86-bin.tar.gz
```

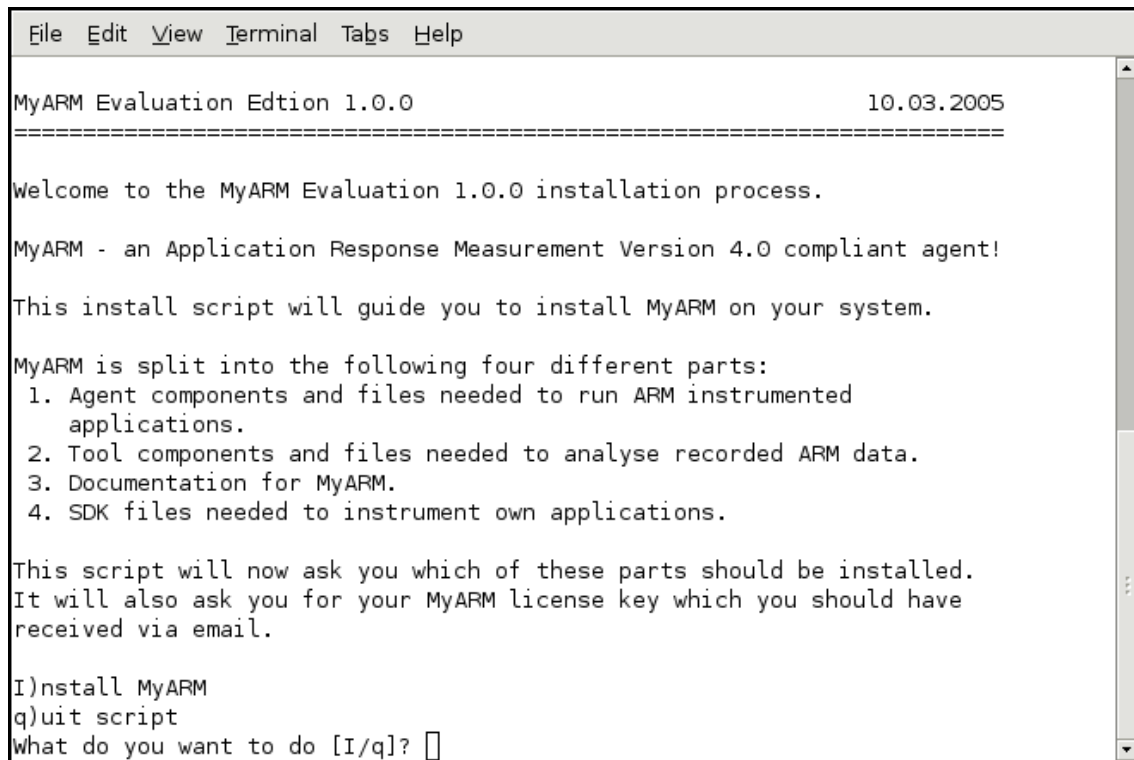
The following command sequence will extract the archive in a new directory named `myarm-1.4.2584.0` and starts the installation process with the provided installation script:

```
# tar xvzf myarm-1.4.2584.0-professional-linux-x86-bin.tar.gz
# cd myarm-1.4.2584.0
# ./install.sh
```

Next section will guide you through the installation script. If you want to extract and install MyARM manually skip it.

2.3 Installing using provided installation script

Executing the provided install script will present you the greeting page shown in [Figure 1](#). It asks if you want to proceed with the installation. Enter `Y` and you will be prompted to answer few questions about which parts of MyARM you want to install.

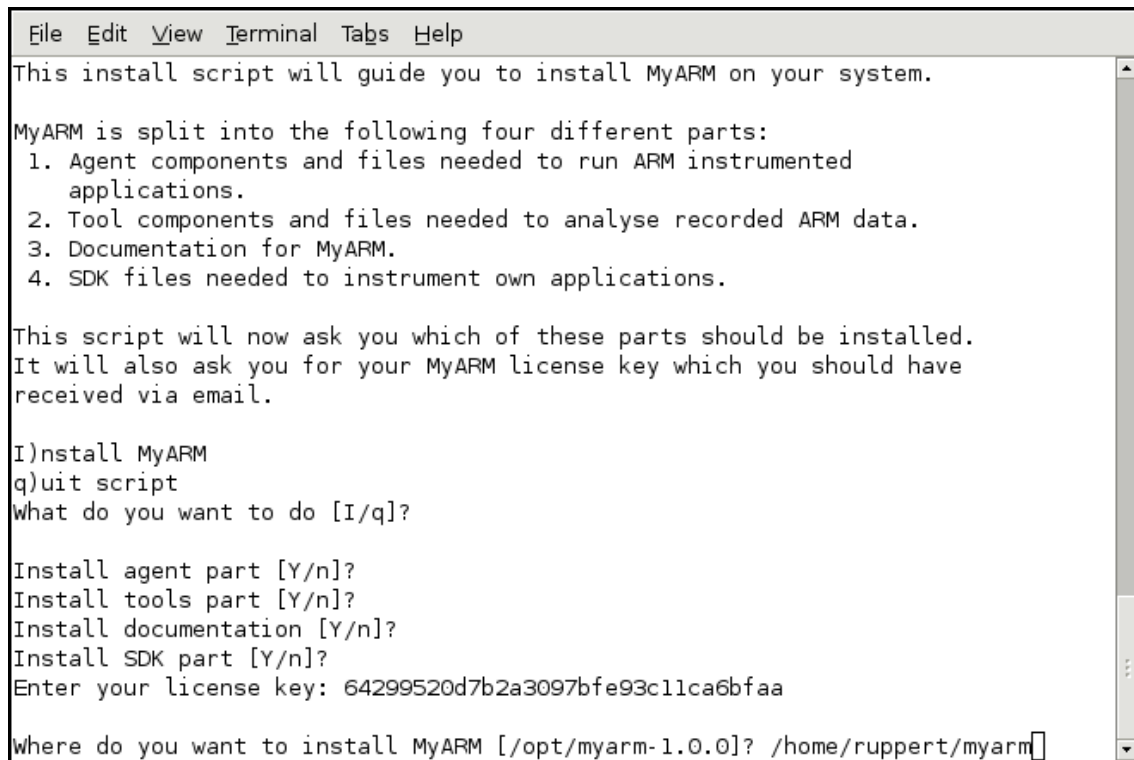


```
File Edit View Terminal Tabs Help
MyARM Evaluation Edition 1.0.0 10.03.2005
=====
Welcome to the MyARM Evaluation 1.0.0 installation process.
MyARM - an Application Response Measurement Version 4.0 compliant agent!
This install script will guide you to install MyARM on your system.
MyARM is split into the following four different parts:
 1. Agent components and files needed to run ARM instrumented
    applications.
 2. Tool components and files needed to analyse recorded ARM data.
 3. Documentation for MyARM.
 4. SDK files needed to instrument own applications.
This script will now ask you which of these parts should be installed.
It will also ask you for your MyARM license key which you should have
received via email.
I)nstall MyARM
q)uit script
What do you want to do [I/q]? 
```

Figure 1: Install greeting page

You will be asked to install:

1. the agent part; needed to deploy instrumented applications.
2. the tools part; needed to analyse the measured data.
3. the documentation part.
4. the SDK part (needed to instrument own applications).
5. the license key you received via email.
6. and the directory where to install MyARM to.



```
File Edit View Terminal Tabs Help
This install script will guide you to install MyARM on your system.

MyARM is split into the following four different parts:
1. Agent components and files needed to run ARM instrumented
   applications.
2. Tool components and files needed to analyse recorded ARM data.
3. Documentation for MyARM.
4. SDK files needed to instrument own applications.

This script will now ask you which of these parts should be installed.
It will also ask you for your MyARM license key which you should have
received via email.

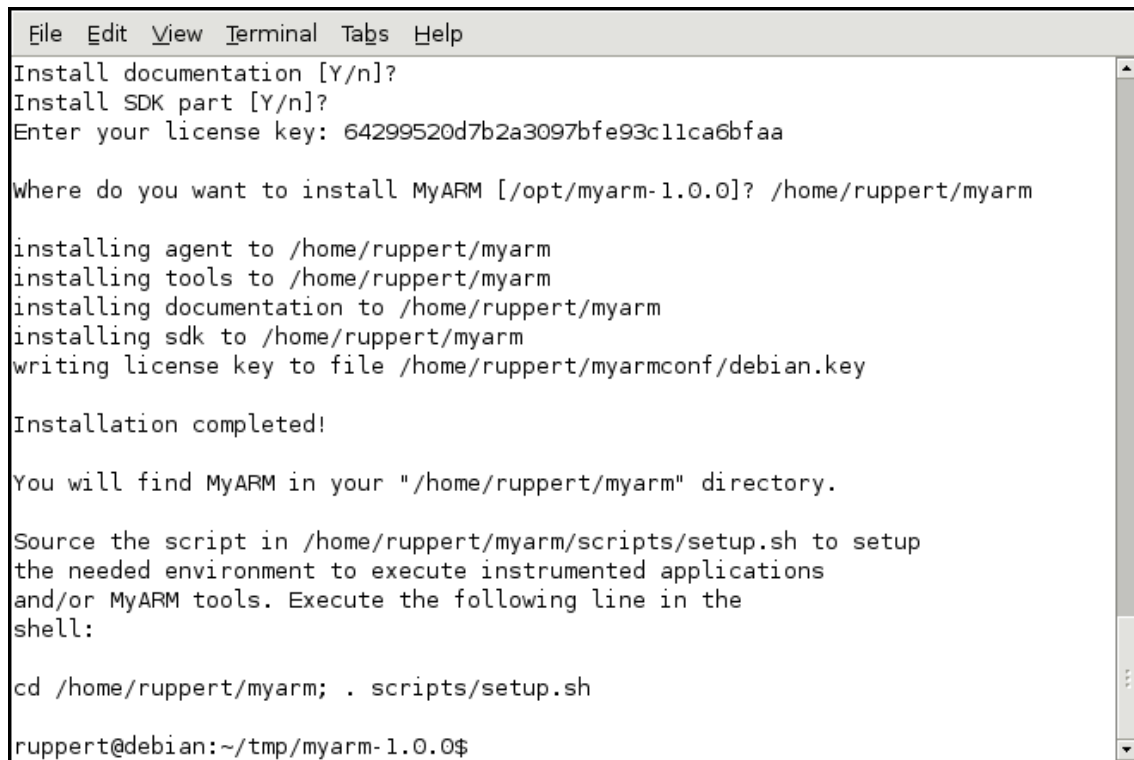
I)nstall MyARM
q)uit script
What do you want to do [I/q]?

Install agent part [Y/n]?
Install tools part [Y/n]?
Install documentation [Y/n]?
Install SDK part [Y/n]?
Enter your license key: 64299520d7b2a3097bfe93c11ca6bfaa

where do you want to install MyARM [/opt/myarm-1.0.0]? /home/ruppert/myarm
```

Figure 2: Install questions

Figure 3 shows a completed installation and you should execute the `setup.sh` script as recommended to setup the MyARM environment.

A terminal window with a menu bar (File, Edit, View, Terminal, Tabs, Help) and a scroll bar on the right. The text inside the terminal shows the installation process for MyARM, including prompts for documentation, SDK, license key, and installation path, followed by progress messages and a final completion notice.

```
File Edit View Terminal Tabs Help
Install documentation [Y/n]?
Install SDK part [Y/n]?
Enter your license key: 64299520d7b2a3097bfe93c11ca6bfaa

Where do you want to install MyARM [/opt/myarm-1.0.0]? /home/ruppert/myarm

installing agent to /home/ruppert/myarm
installing tools to /home/ruppert/myarm
installing documentation to /home/ruppert/myarm
installing sdk to /home/ruppert/myarm
writing license key to file /home/ruppert/myarmconf/debian.key

Installation completed!

You will find MyARM in your "/home/ruppert/myarm" directory.

Source the script in /home/ruppert/myarm/scripts/setup.sh to setup
the needed environment to execute instrumented applications
and/or MyARM tools. Execute the following line in the
shell:

cd /home/ruppert/myarm; . scripts/setup.sh

ruppert@debian:~/tmp/myarm-1.0.0$
```

Figure 3: Install completed

2.4 Installing manually

MyARM provides an installation script for easy installation. So please use this script to install MyARM on your system. However, in some cases it might be necessary to install it manually. For this purpose MyARM is divided into the following parts:

MyARM agent – contains all necessary files to deploy an ARM instrumented application with the MyARM agent.

MyARM tools – contains all necessary files to analyse recorded ARM data.

MyARM documentation – contains all documentation of MyARM.

MyARM SDK – contains all necessary files to develop ARM instrumented applications.

All these archives extract into one tree if they are unpacked into the same directory. The following command will unpack and install MyARM, assuming archives are in the current working directory and you want to install MyARM into the current working directory. It will create a sub-directory called myarm-1.4.x.0.

```
tar xvzf myarm-*.tar.gz
```


3 Installing the MyARM distribution on a workstation under MS-Windows

The installation under windows is very easy and straightforward. Just double-click on the received msi package. It starts with a welcome dialog (Figure 4) and requires the user to press the next button several times until it is fully installed.

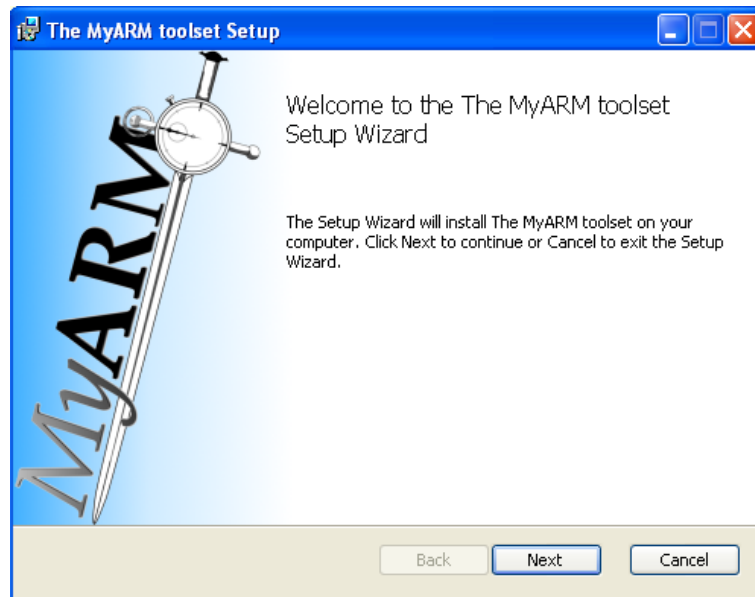


Figure 4: Installation welcome dialog

The next dialog (Figure 5) shows the license under which MyARM will be installed. You must accept the license by clicking on the respective checkbox. The next button can be clicked only when the license was accepted.



Figure 5: Installation MyARM license dialog

The license key dialog (Figure 6) allows to type in the MyARM license that was received via e-mail. This is a string with about 32 characters or more. If you did not receive a license yet leave this as is and change the license key afterwards as described in the chapter below. You will only be able to read the documentation provided as pdf and html files if you did not receive any license.

This dialog is skipped for the community edition since here no license is necessary.

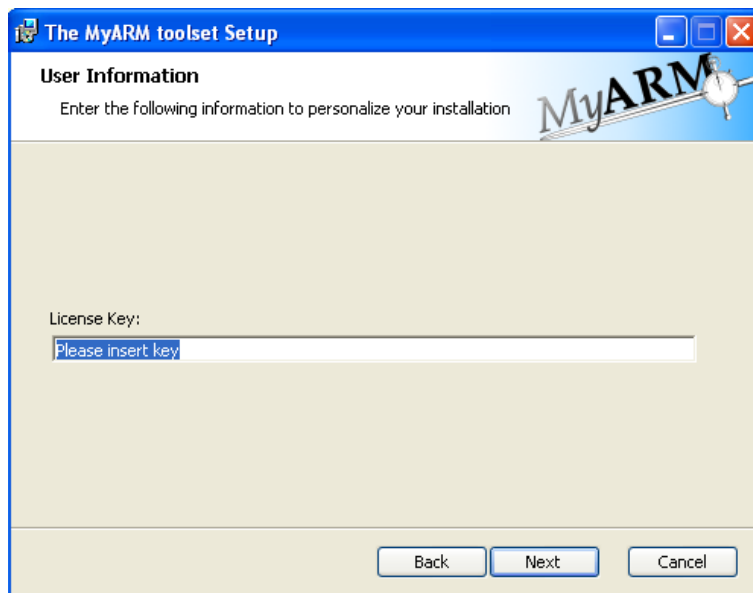


Figure 6: License key dialog

Next the directory where MyARM will be installed can be selected in the installation directory dialog (Figure 7). Type in the name of the directory you want to install MyARM into or use the "Change" button

to open a standard directory dialog where you can select the target directory graphically.

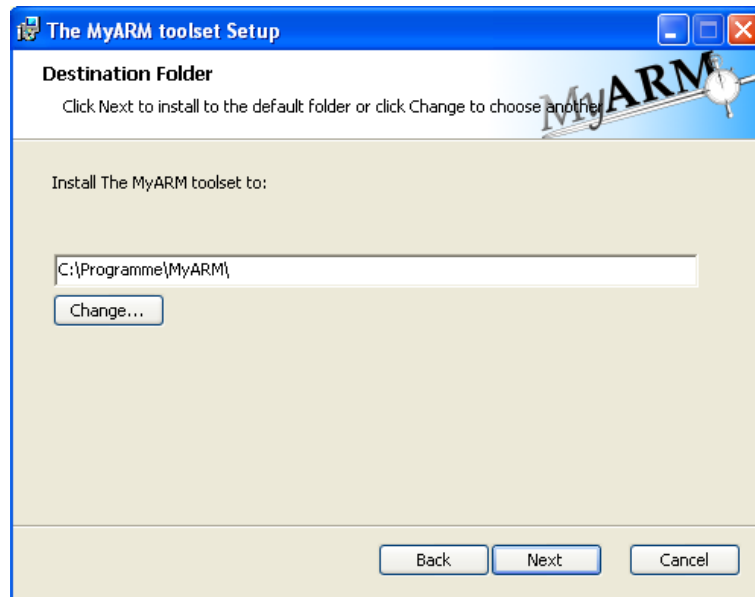


Figure 7: Installation directory dialog

When you are at this stage you are almost done. You will be prompted to start the installation by clicking Next in the “finished dialog” (Figure 8). You will then see the progress in the respective dialog (Figure 9)

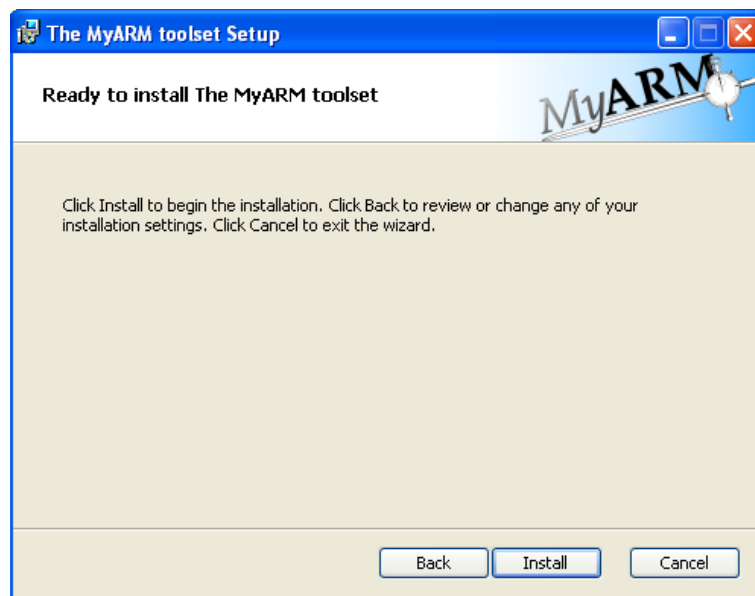


Figure 8: Installation finished dialog

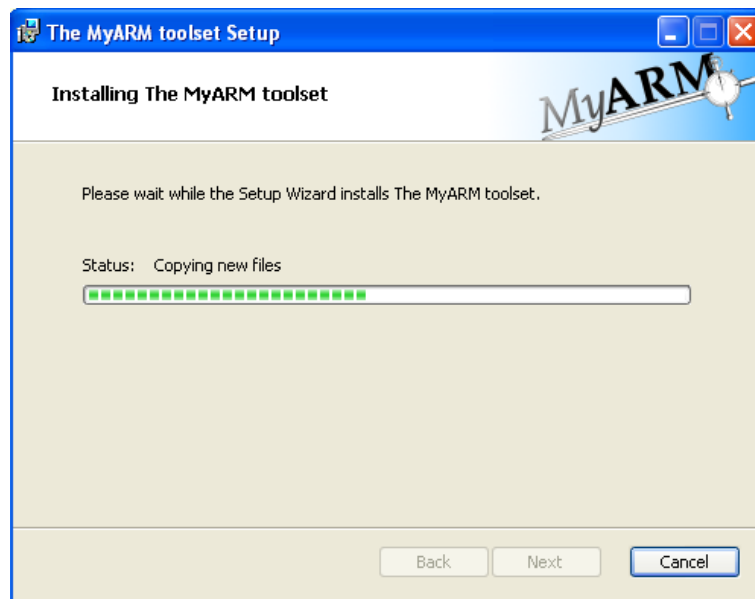


Figure 9: Installation progress dialog

When the installation has completed you will be asked to press next to finish the installation (Figure 10).

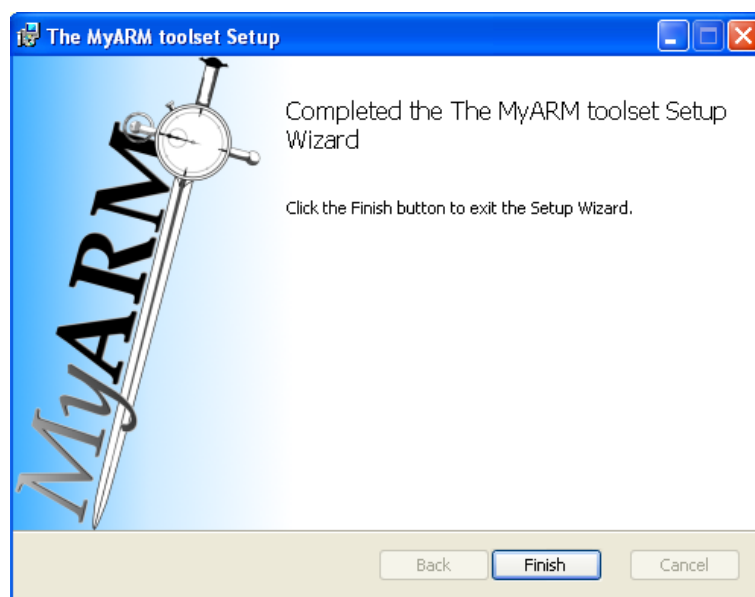


Figure 10: Installation completed dialog

3.1 Changing the license key after installation

If you did not receive a license key before installation then you were not able to fill in the required key in the respective dialog (Figure 6). The installation just creates the registry key `HKEY_LOCAL_MACHINE/Software/MyARM/Environment/MYARM_LICENSE_KEY` with an empty value so you can easily change this by adapting this key. Just open the registry editor by typing

“regedit” into the execution window. (see [Figure 11](#)) and type in the received license key. You should then be able to execute all MyARM command line tools or the manager if you’ve purchased the respective license.

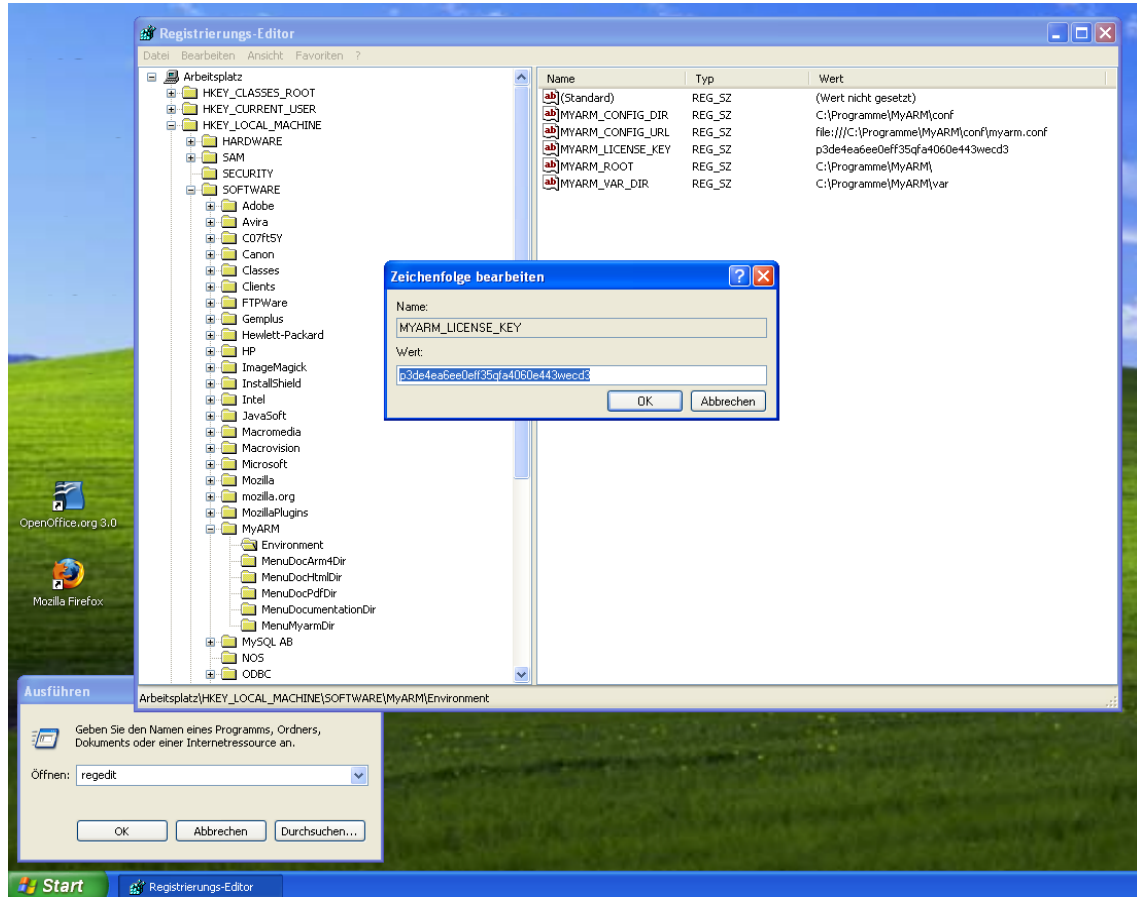


Figure 11: License key change after installation

4 Environment

The MyARM agent library can be configured through configuration files. MyARM needs to find the configuration files during start up. This is achieved by using environment variables under Unix or registry keys under Windows.

4.1 Unix setup

Under UNIX the following environment variables can be set to specify the location of the configuration files and to set a valid license key. The easiest way to do this is to use the provided `setup.sh` script found in the `scripts` directory of the MyARM installation.

4.1.1 scripts/setup.sh

The `setup.sh` script is used to setup a MyARM environment in an easy way. The script tests your shell environment and modifies the following system environment variables to enable the full usage of MyARM:

PATH

is modified to add the path to the MyARM programs so that any MyARM program can be executed.

LIBPATH, LD_LIBRARY_PATH

is modified to enable the runtime linker to find MyARM shared libraries (`LIBPATH` on AIX, `LD_LIBRARY_PATH` on Linux and Solaris).

MANPATH

is modified to add the MyARM man page directory to the search path of the man system.

MYSQL_UNIX_PORT

is set to the mysql port as used by the current system. If this could not be determined a warning message is issued.

MONO_PATH

is modified if MyARM comes with C# support to add the path for the MyARM assemblies.

PYTHONPATH

is modified if MyARM was installed with ARM 4.0 python support. It adds the directory for the appropriate `arm4` python module.

To work properly the following MyARM specific environment variables are set:

MYARM_ROOT

root directory of the MyARM installation.

MYARM_VARRUN_DIR

directory where to store any runtime data such as pid files or temporarily used files

MYARM_VARLOG_DIR

directory where to store any log files

MYARM_VARLIB_DIR

directory where to store any files which need to be persistent such as SQLite databases or MyARM ARM data files

MYARM_CONFIG_URL

contains the configuration end point in URL notation. For more information read section *MYARM_CONFIG_URL*.

MYARM_LICENSE_KEY

contains the license key of MyARM. This key is extracted from some configuration files. For more information see section “*License key*”.

Usage:

```
$ . scripts/setup.sh [-q] [config]
```

The “-q” option suppresses any warning and the basic configuration output. The optional “config” parameter specifies a configuration to be used. This configuration must reside in the MYARM_ROOT/conf or MYARM_ROOT/conf/templates directory.

Typical usage of the setup.sh script, which configures MyARM in the current shell to use the [myarmdaemon](#) to collect files written by the file datasink and store the ARM data into MySQL database:

```
user@localhost:/opt/myarm$ . scripts/setup.sh file_mysql.conf
```

4.2 Windows setup

MyARM for Windows supports registry keys instead of environment variables. During installation of MyARM for Windows the following registry keys are set using the HKEY_LOCAL_MACHINE/Software/MyARM/Environment/<var> key path, where <var> is a placeholder for the variable names in the following sections (e.g. MYARM_CONFIG_URL). Please note that under Windows no further action is necessary to make these variables known to the MyARM utilities.

4.3 MyARM environment variables**4.3.1 MYARM_CONFIG_URL**

The MYARM_CONFIG_URL environment variable is used to specify the MyARM configuration file. It uses the common URL notation of the form <protocol>://<host>:<port>/<filepath>. Currently, the following <protocol> types are supported:

file

specifies a local configuration file.

For example, if you want to specify a global <filepath> you have to use two slashes after the protocol indicator, resulting in the following setup:

```
MYARM_CONFIG_URL=file:///opt/myarm/conf/foo.conf
export MYARM_CONFIG_URL
```

```
myarmquery FooTran FooApp
```

To reference a file within the MYARM_CONFIG_DIR directory, just omit one slash. The following example locates the foo.conf file in the directory configured with the MYARM_CONFIG_DIR environment variable.

```
MYARM_CONFIG_URL=file:///foo.conf
export MYARM_CONFIG_URL
```

```
myarmquery FooTran FooApp
```

4.3.2 MYARM_CONFIG_DIR

If the `MYARM_CONFIG_DIR` environment variable is set it contains the absolute path to the directory where all ARM configuration files reside. Therefore you don't need to specify the full path of a configuration file; just use the default file name itself. For example, if you have a configuration stored in the file `fooproject.conf` which is located in the `/usr/local/myarm/conf` directory, you can use the following statements:

```
export MYARM_CONFIG_DIR=/usr/local/myarm/conf  
  
myarmquery -C fooproject.conf FooTran FooApp
```

4.3.3 MYARM_LICENSE_KEY

If the `MYARM_LICENSE_KEY` environment variable is set and its not empty it contains the license key to deploy the MyARM components on this host.

4.3.4 MYARM_VARRUN_DIR

This variable is used within the configuration files to reference the directory where to store runtime data like pid files or temporary data.

4.3.5 MYARM_VARLOG_DIR

This variable is used within the configuration files to reference the directory where to log messages.

4.3.6 MYARM_VARLIB_DIR

This variable is used within the configuration files to reference the directory where to store persistent data such as a SQLite database file or ARM data files.

4.3.7 MYARM_TRACE

With the `MYARM_TRACE` environment variable it is possible to enable diagnostic trace messages during configuration and initialization of the MyARM agent to standard error channel. Currently the following options are supported, which can be combined using a comma character as the delimiter:

none

no tracing at all.

all

enables all the following trace options.

init

enables tracing of initialization and cleanup of the MyARM agent.

env

enables tracing output of all used environment variables.

config

enables tracing output of all configuration properties.

readcfg

enables tracing of reading configuration.

For example:

```
export MYARM_TRACE=init,env  
export MYARM_TRACE=all
```

5 License key

MyARM checks for a valid license using the so-called *license key*. MyARM product family is divided into several editions. A license key is only valid for the given edition and release. You get your license key for MyARM only by email after finishing the ordering process and the license fee was paid.

The license key can be made available to the MyARM components with the following ways:

1. Using the `MYARM_LICENSE_KEY` environment variable which contains the license key provided to run MyARM components on the current host.
2. Adding the following line in each configuration file you use to deploy ARM instrumented applications and MyARM tools:

```
myarm.key = <license_key_string>
```

3. Using a `keyfile` containing the license key of the host MyARM is running on. The file must be located in the directory of the actual configuration file. The filename has to be either `myarm.key` or `<host>.key` where `<host>` is the name of the current host.