DevkitManager

Generated by Doxygen 1.8.8

Tue Dec 9 2014 14:35:24

Contents

Chapter 1

User Reference

1.1 Introduction

Devkit is a container based software deployment system. Software components and its dependencies are bundled in a so called DevkitContainer which can be distributed easily over network or as update-package on a storage-device. Software inside a DevkitContainer can be executed without any installation. DevkitManager is a tool to manage different instances of these software bundles. Because all data of a container is stored in a Git-repository it is also possible to switch between different versions of a DevkitContainer.

1.1.1 Basic Features

DevkiManager provides the following key-features:

- Create new DevkitContainer
- Verify actual state of a DevkitContainer
- Reset a DevkitContainer to factory default
- Switch between different versions of a DevkitContainer
- Check for updates and install new versions (online-workflow)
- Generate update-packages between two versions (offline-workflow)

1.1.2 Product Definitions

The subsequent list defines product specific terms which are used in this document.

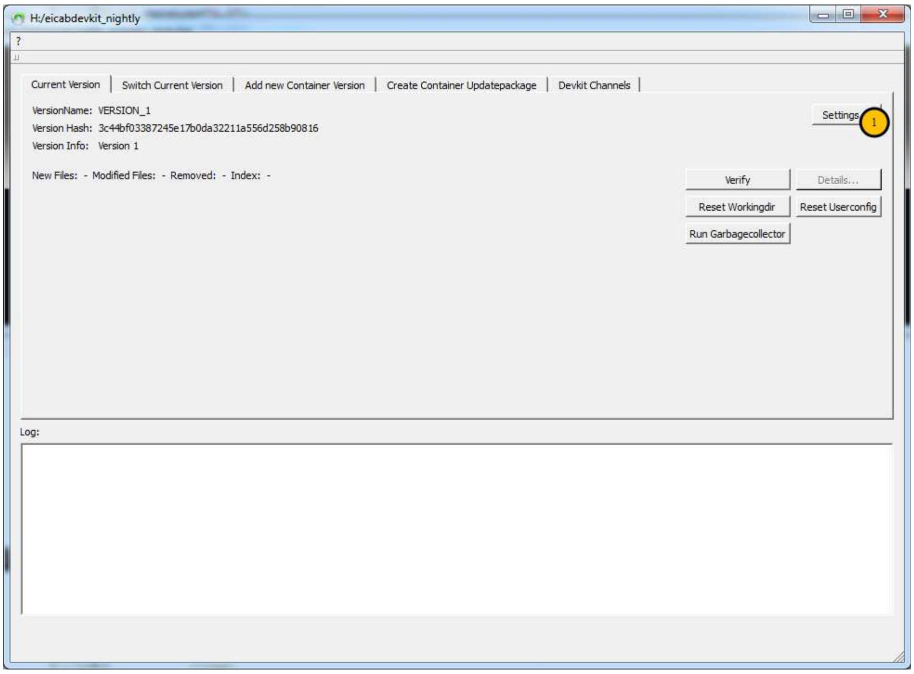
- **Devkit:** Container based software deployment/distribution system
- **DevkitContainer:** Software bundle which can be executed without installation
- **DevkitManager:** Tool to mange DevkitContainers
- **DevkitSandbox:** Runtime environment of a DevkitContainer
- **DevkitIDE:** DevkitContainer which includes the Integrated Development Environment (IDE) for the eiCAB-↩ HMI System.
- **DevkitChannel:** Source for online updates
- **DevkitChannelId:** Identification string of a DevkitChannel

1.2 Getting Started

DevkitManager can be executed without an installation copy the executable to a desired destination and start it. If no DevkitContainer is found on the system the 'settings'-dialog appears. Just follow the instructions as described in section DevkitManager Configuration to create an empty DevkitContainer.

1.3 DevkitManager Configuration

This sections describes how to configure DevkitManager and its DevkitContainer instances . All configuration settings are done in the 'settings'-dialog which can accessed from the main-view (1).



H:/eicabdevkit_nightly

VersionName: VERSION_1
Version Hash: 3c44bf03387245e17b0da32211a556d258b90816
Version Info: Version 1

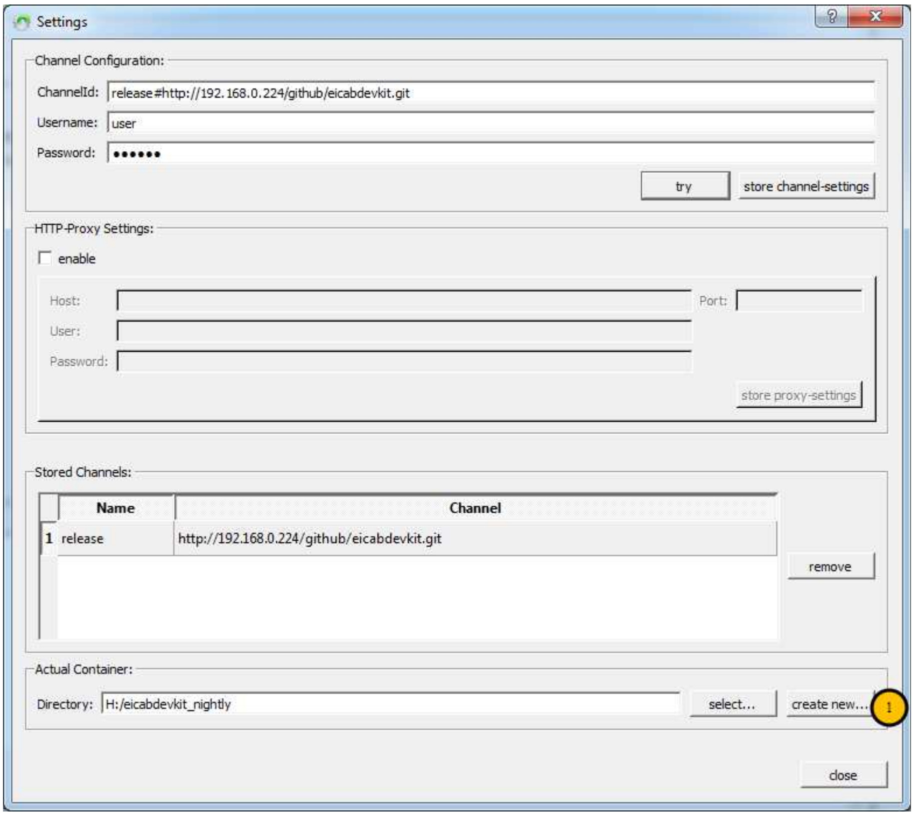
New Files: - Modified Files: - Removed: - Index: -

Log:

Figure 1.1: Open Settings Dialog

1.3.1 Create a new empty Container

To create a new empty DevkitContainer the 'create'-dialog has to be openend(1)



Settings

Channel Configuration:

ChannelId: release#http://192.168.0.224/github/eicabdevkit.git

Username: user

Password: ••••••

HTTP-Proxy Settings:

□ enable

Host:

Port:

User:

Password:

Stored Channels:

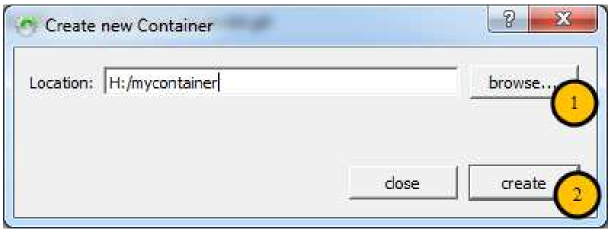
	Name	Channel
1	release	http://192.168.0.224/github/eicabdevkit.git

Actual Container:

Directory: H:/eicabdevkit_nightly

Figure 1.2: Open Create Dialog

In the 'create'-dialog you can select a location for the new DevkitContainer(1). The location path has to point to an empty or non existent directory. After the selection of a valid location a click of the 'create'-Button(2) will create a new empty DevkitContainer instance. If the operation is successfully DevkitManager will set its working-directory to the new location. At this point the DevkitContainer is ready to be filled with data either with the usage of an update-package as described in section Apply an Update-Package or with a configured DevkitChannel as described in section Configure a DevkitChannel and Fetch DevkitContainer over Network.



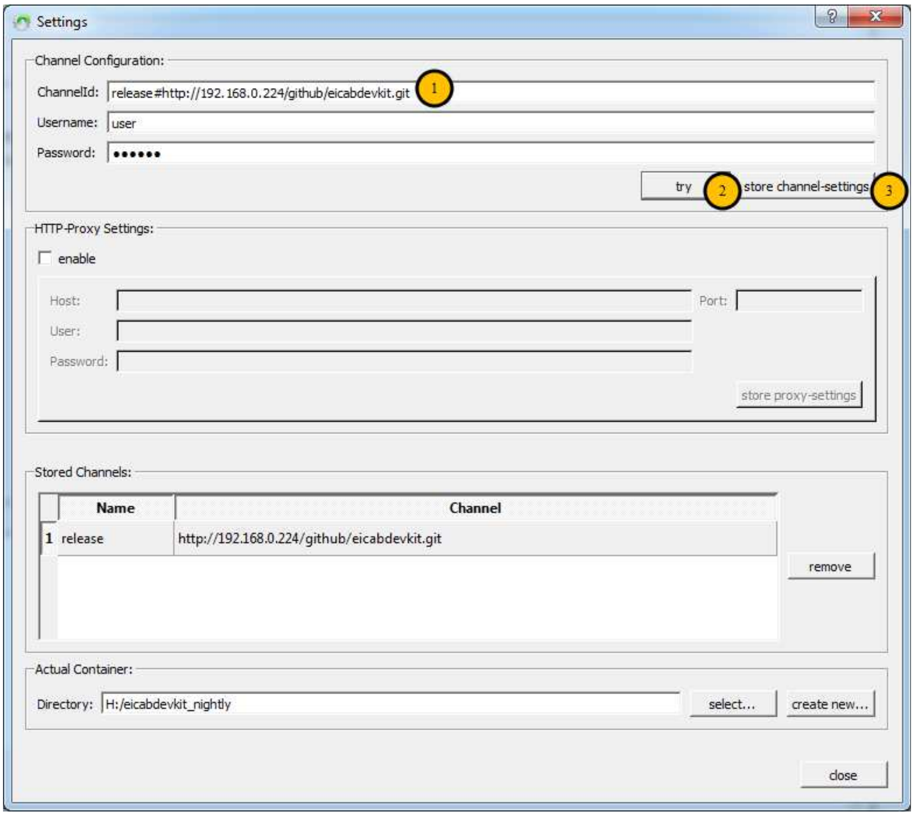
Create new Container

Location: H:/mycontainer

Figure 1.3: Select Target directory

1.3.2 Configure a DevkitChannel

DevkitChannels are used as a source for DevkitContainers. A DevkitChannelId(1) consists of a name and a URI separated by a '#'. As transport protocols currently "http" and "https" are supported. If a DevkitChannel has a restricted access policy "Username" and "Password" have to be specified. The 'try'-Button(2) can be used to validate the provided DevkitChannel-Settings. After a successful verification the DevkitChannel can be stored by pressing the 'store'-Button(3). All stored channels are listed in the channel list below and the 'drop down 'select'-box of the 'Devkit Channel'-tab as described in Fetch DevkitContainer over Network.



Settings

Channel Configuration:

ChannelId: release#http://192.168.0.224/github/eicabdevkit.git

Username: user

Password: ••••••

HTTP-Proxy Settings:

□ enable

Host:

Port:

User:

Password:

Stored Channels:

	Name	Channel
1	release	http://192.168.0.224/github/eicabdevkit.git

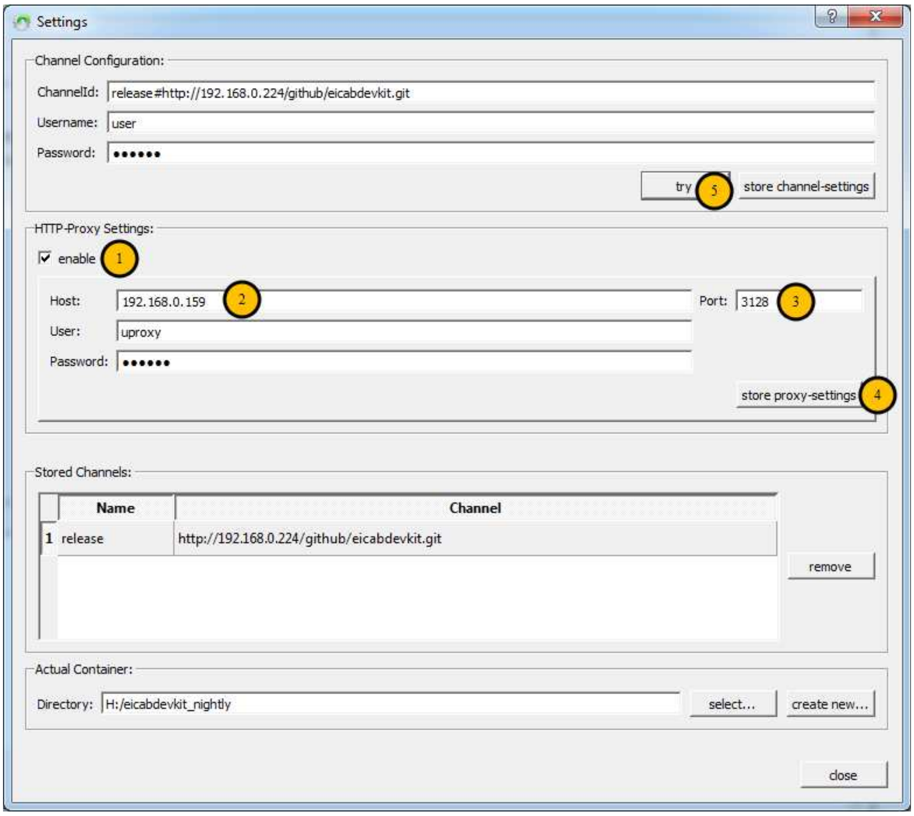
Actual Container:

Directory: H:/eicabdevkit_nightly

Figure 1.4: Configure a DevkitChannel

1.3.3 Configure DevkitManager behind a HTTP-Proxy

If you want to use DevkitManager behind a HTTP-Proxy you have to enable the HTTP-Proxy settings (1).



Settings

Channel Configuration:

ChannelId: release#http://192.168.0.224/github/eicabdevkit.git

Username: user

Password: ••••••

HTTP-Proxy Settings:

☑ enable

Host: 192.168.0.159

Port: 3128

User: uproxy

Password: ••••••

Stored Channels:

	Name	Channel
1	release	http://192.168.0.224/github/eicabdevkit.git

Actual Container:

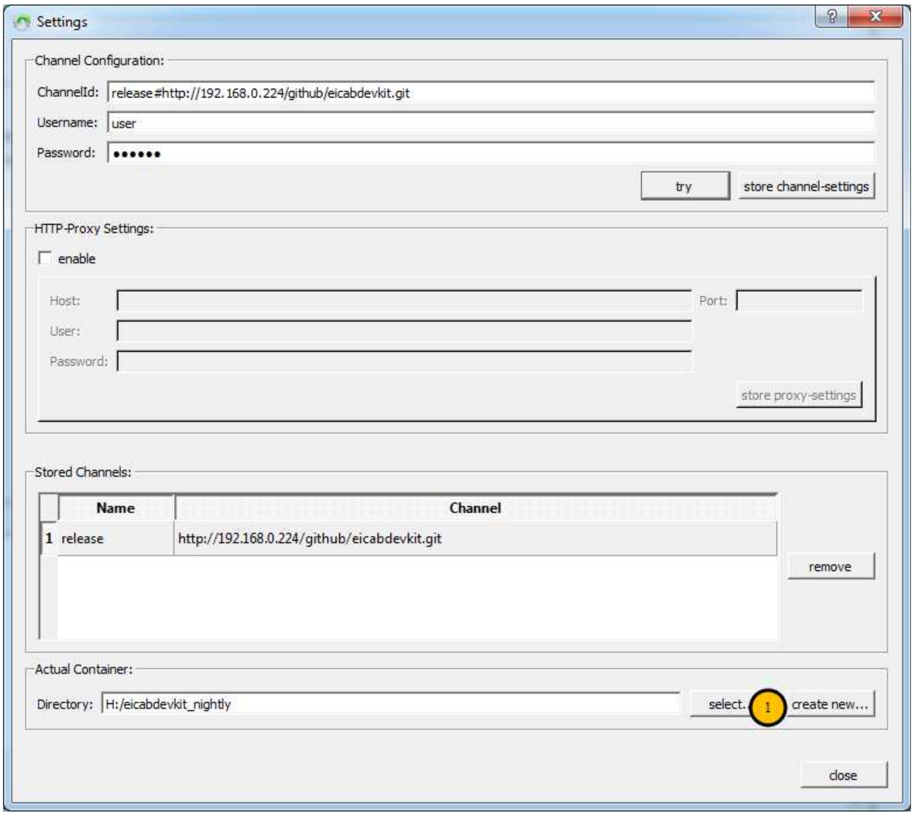
Directory: H:/eicabdevkit_nightly

Figure 1.5: HTTP-Proxy Settings

The Http-URI of the proxy-server can be specified in input-field(2). The port can be defined in (3). Optional it is also possible the provide a username and password (if note needed leave this fields empty). The connectivity including the proxy-settings can be validated by pressing (5). For a permanent storage of the proxy-settings press the 'store proxy-settings'-Button(4).

1.3.4 Set Working Directory (Switch between DevkitContainer Instances)

By pressing the 'select'-Button (1) the working directory of DevkitManager can be set to another DevkitConatiner.



Settings

Channel Configuration:

ChannelId: release#http://192.168.0.224/github/eicabdevkit.git

Username: user

Password: ••••••

HTTP-Proxy Settings:

□ enable

Host:

Port:

User:

Password:

Stored Channels:

	Name	Channel
1	release	http://192.168.0.224/github/eicabdevkit.git

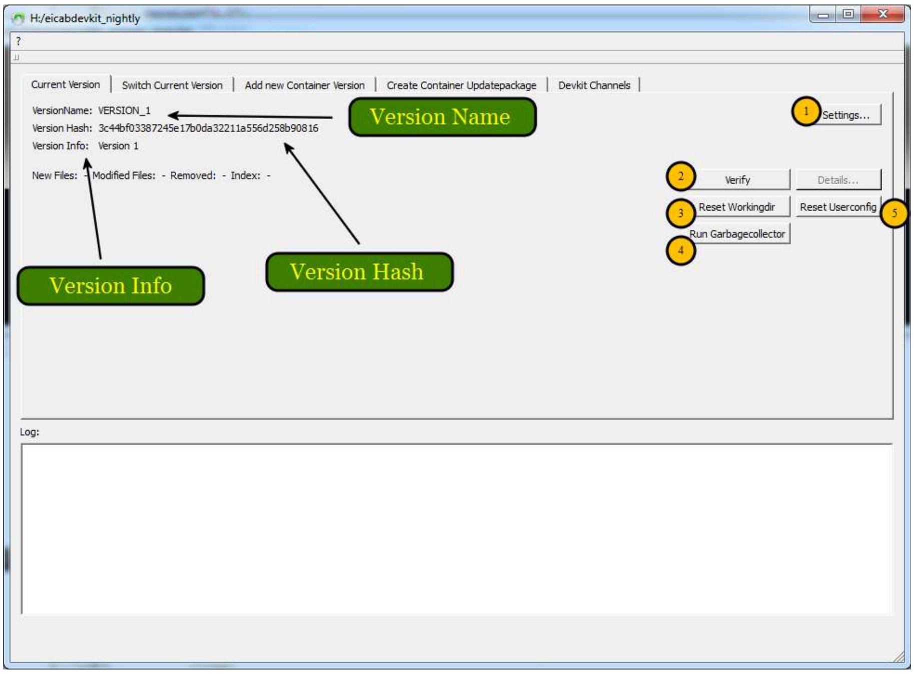
Actual Container:

Directory: H:/eicabdevkit_nightly

Figure 1.6: Select new DevkitContainer Working Directory

1.4 MainView (Current Version)

After a successful start-up , DevkitManager sets the current working directory to the previously used DevkitContainer and shows the 'main'-view. This tab summarizes the main-properties of the active DevkitContainer together with its version information. Every DevkitContainer can be identified by its version hash. This hash corresponds to the Git-TreeId of the working directory. It is derived from the content of the DevkitContainer and is therefore used as unique identifier. The version info and version name act as human readable version description. Pressing (3) starts a verification process as described in Verify a DevkitContainer. (1) opens the 'settings'-dialog to configure DevkitManager as described in DevkitManager Configuration. If a container version is deleted unnecessary objects still remain in the file-system. To remove this obsolete data the garbage collector has to be executed by pressing (4). This operation can last several minutes but lowers also the disk usage significant. User-specific configuration and temporary data is stored in the 'config'-directory which can be reset by pressing (5)



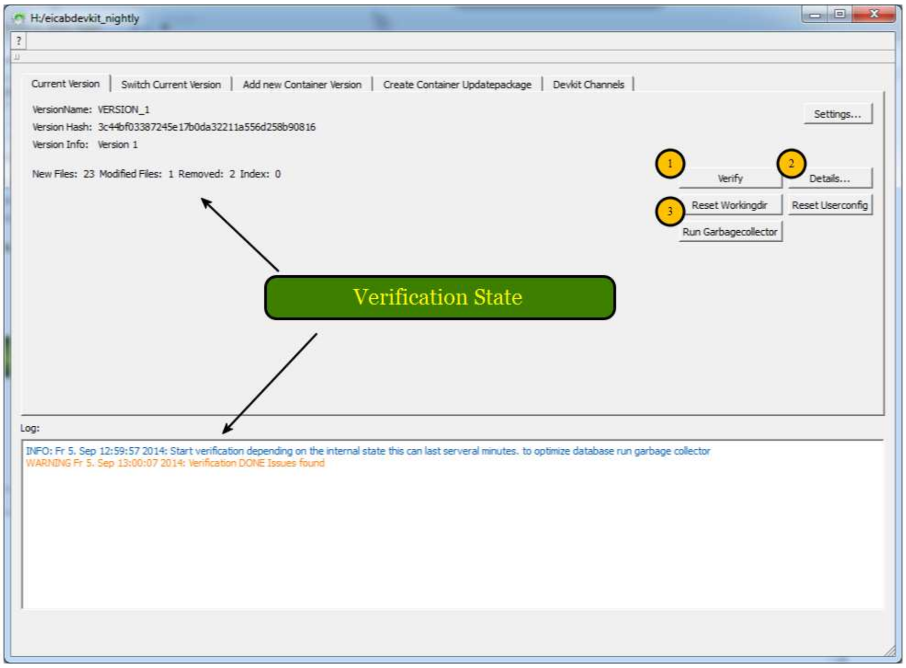
VersionName: VERSION_1
Version Hash: 3c44bf03387245e17b0da32211a556d258b90816
Version Info: Version 1

New Files: - Modified Files: - Removed: - Index: -

Figure 1.7: Main View

1.4.1 Verify a DevkitContainer

One of the key features of the Devkit software deployment system is the ability to check the integrity of a DevkitContainer. Every DevkitContainer-Version is signed by a SHA1- Hash Value. If a file of the working directory is removed or modified Devkit can detect and revert such changes in order to restore the integrity of a DevkitContainer. A verify process is started by clicking (1). This can last several minutes. Afterwards a summary of the verification result is provided on the main page. If less than 30 issues are found a detailed report is provided by pressing (2). If desired it is possible the revert the changes and restore the original state of the working directory by pressing (3)



VersionName: VERSION_1
Version Hash: 3c44bf03387245e17b0da32211a556d258b90816
Version Info: Version 1

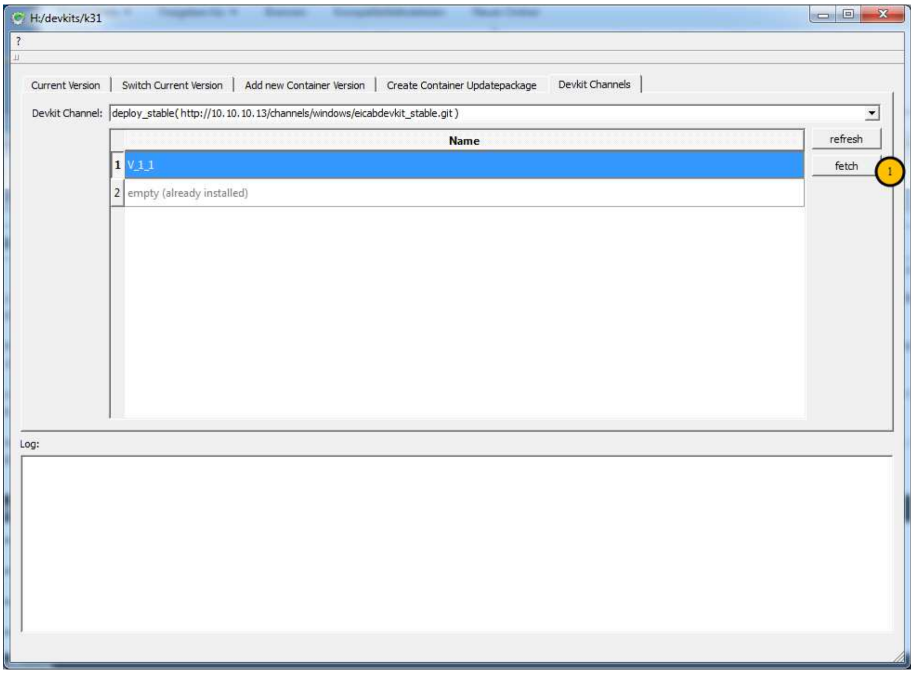
New Files: 23 Modified Files: 1 Removed: 2 Index: 0

```
INFO: Fr 5. Sep 12:59:57 2014: Start verification depending on the internal state this can last serveral minutes. to optimize database run garbage collector
WARNING Fr 5. Sep 13:00:07 2014: Verification DONE Issues found
```

Figure 1.8: Verification Result

1.5 Fetch DevkitContainer over Network

After a successful configuration of a DevkitChannel as described in Configure a DevkitChannel it appears in the 'select'-box of the 'Channel'-tab. By selecting a DevkitChannelId all available containers are displayed in the table below. Choose the desired version and press (1) to fetch the data. After the transfer the working directory of the file-system stays untouched. In order to set the working directory in the file-system to the new version it has to be selected in the 'Switch Current Verison'-tab as described in Switch between DevkitContainer Versions.



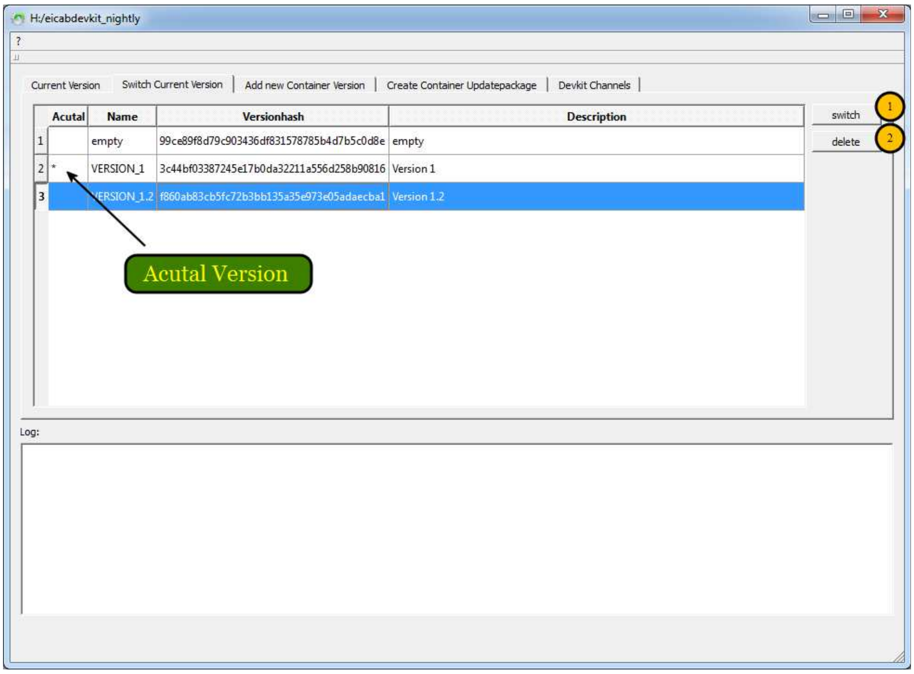
Devkit Channel: deploy_stable(http://10.10.10.13/channels/windows/eicabdevkit_stable.git)

	Name
1	V_1_1
2	empty (already installed)

Figure 1.9: DevkitContainer over Network

1.6 Switch between DevkitContainer Versions

This tab is used to set the current working directory of a DevkitContainer to a specific version. Just select the desired version and press (1). **CAUTION also the user-specific configuration gets reset!** . If a specific version is not needed any more press (2) to remove it. For a complete removal which also lowers the disk usage the garbage collector has to be executed from the MainView (MainView (Current Version)).

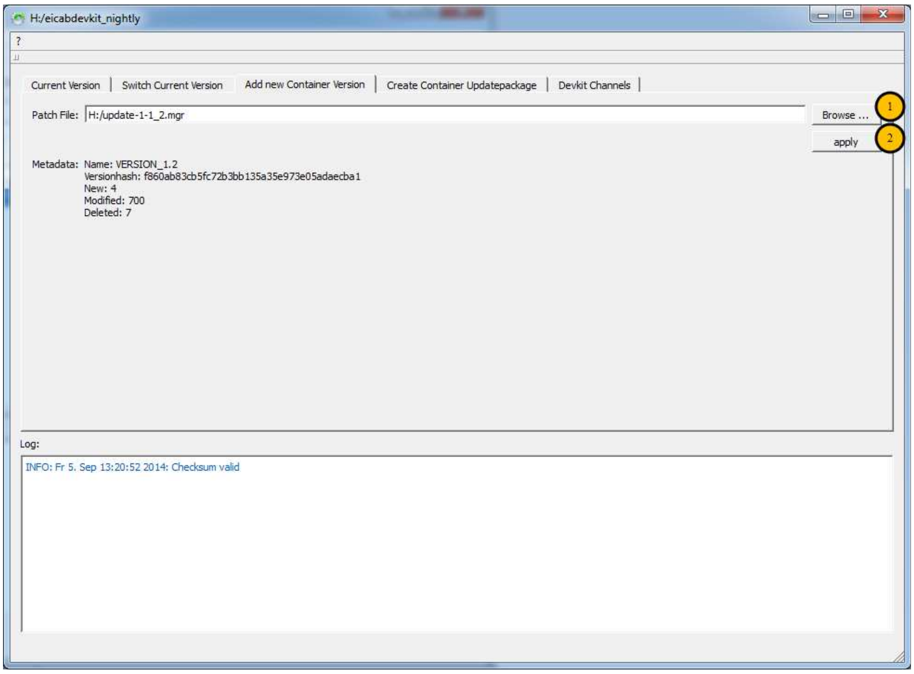


	Acutal	Name	Versionhash	Description
1		empty	99ce89f8d79c903436df831578785b4d7b5c0d8e	empty
2	*	VERSION_1	3c44bf03387245e17b0da32211a556d258b90816	Version 1
3		VERSION_1.2	f860ab83cb5fc72b3bb135a35e973e05adaecba1	Version 1.2

Figure 1.10: Switch between DevkitContainer Versions

1.7 Apply an Update-Package

New versions of a DevkitContainer can be distributed with an update-package. An update-package consists of a single file with the extension ".mgr". To insert a new version to a DevkitContainer select the update-package (1). If the integrity of the package is valid the new version can be added by pressing (2). In order to set the working directory in the file-system to the new version it has to be selected in the 'Switch Current Verison'-tab as described in Switch between DevkitContainer Versions.



Patch File: H:/update-1-1_2.mgr

```
Metadata: Name: VERSION_1.2
          Versionhash: f860ab83cb5fc72b3bb135a35e973e05adaecba1
          New: 4
          Modified: 700
          Deleted: 7
```

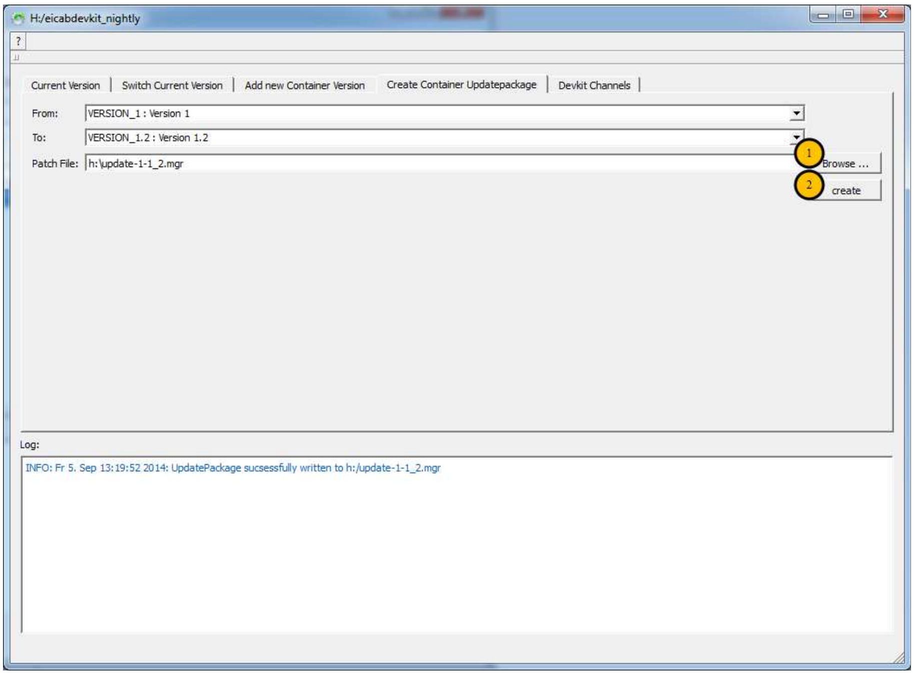
Log:

```
INFO: Fr 5. Sep 13:20:52 2014: Checksum valid
```

Figure 1.11: Apply Update-Package

1.8 Create an Update-Package (Offline-Worklow)

DevkitManager can be used to generate an Update-Package. This packages contain only the differences between two versions. Therefore a "from"- and "to"-version has to be selected and the ’from’-version has to be available in the target DevkitContainer. After the selection of a output-file. The update-package is written by clicking (2). This file can be applied on the target DevkitContainer as described in Apply an Update-Package.



From: VERSION_1 : Version 1

To: VERSION_1.2 : Version 1.2

Patch File: h:/update-1-1_2.mgr

Log:

```
INFO: Fr 5. Sep 13:19:52 2014: UpdatePackage sucsessfully written to h:/update-1-1_2.mgr
```

Figure 1.12: Create an Update-Package