

User's Manual of OBS Light

For information purpose only

Document status: Draft 0.1

Authors :

Name	E-Mail	Date	Operation on the document	Version
Gustav Hellmann	vladimir@fridu.net	09/06/11	Creation	0.1

Diffusion :

Name	E-mail	Date	Version
Dominique Le Foll	dominig.arfoll@fridu.net	10/06/11	0.1

Pending tasks:

Task	Assigned to	E-Mail
To validate	Dominique Le Foll	dominig.arfoll@fridu.net

1.Acknowledgment

All the development of the OBS Light is done under the auspices of the Open Technology Center (OTC) of Intel and with the financial support of Intel and of its subsidiary Wind River. Intel possesses the full intellectual property rights on any commercial or non-commercial use of the OBS Light or associated services such as maintenance or users' training as well as the possibility to review or to revoke the GPL License under which the first version of OBS Light is currently distributed. All persons who participate in the OBS Light project explicitly acknowledge this fact and thank Intel for its support.

2.Installation

An installation launcher for OBS Light is provided on the DVD. Alternatively, the user can download the installation rpm from a server: <http://obslight/releases/latest-release/installation> (not available now).

3.Introduction

OBS Light is a fully self-sufficient local build tool. It allows:

- Local administration.
- Auto-patching.
- Local cross-compilation of RPM packages.
- Creation of MIC2 images.
- Simulation of deployment of a created image.

The development of OBS Light has been launched in order to better respond to the growing demands of MeeGo's industrial partners and developers for a concise local build tool as an alternative to OBS. OBS Light possesses the following characteristics:

- It offers two operational modes: the UI mode and the commandline mode. An augmented version of OSC that contains new functions is used.
- It allows the user to work locally and completely detached from the remote OBS.
- The user can add a new RPM package to the existing local target repository or modify an existing RPM package within it.
- The user can launch a local build of a rpm package for either i586 or armv7 architecture.
- The MIC2 image creation is fully integrated. As for OSC, an augmented version of MIC2 that contains new functions is used.
- A synchronization between the OBS Light and the remote OBS is possible.

In the current form, OBS Light can be used for any embedded MeeGo project. Developed originally in the MeeGo context, OBS Light aims in the long term to become a popular universal local cross-compilation tool. The development of OBS Light is governed by best open source practices:

- OBS Light is distributed under the GPLv2 agreement and respects the Open Source license agreements with all of its dependencies by providing explicit source code links.
- A mailing list is available which informs the user about the newest updates of the application.
- A bug tool is available where the user can submit bugs: <http://obslight/bugzilla>

The UI of OBS Light provides contextual help topics and the user is explicitly encouraged to consult them. Alternatively, the user can obtain help in the commandline mode by the means of '--help' option.

4.The basic functions of the UI of OBS Light

4.1 The management of target repositories for the build

The user can choose in the list of locally available target repositories the repository he wants to build against. If necessary, he can synchronize the specified local target repository with the remote OBS repository. The user can add a new target repository to the list of existing repositories by entering its URL or delete an existing repository. In the window with the list of existing target repositories the user can see information about the update status of each target repository.

4.2 The management of the user's projects

Via a radio button the user has first to make a choice between 2 operational modes within the OBS Light:

- The Offline mode: the default OBS Light mode which is local and completely independent from OBS.
- The OBS mode: the user works with OBS, that is either locally installed or remote.

For beginner: Work in the Offline mode.

In order to be able to access his projects, the user has now to log into the OBS Light project management system by typing his Username and Password. After a successful login the user can select a project he wants to open in the combo box containing the list of available projects. If working in the Offline mode of OBS Light, the user can create a new project or delete an existing one via buttons available on the UI.

For beginner: You can find an existing project on the server or on the DVD.

Remark: A chroot is linked to the project.

4.3 The management of the RPM and of its packages

4.3.1 RPM source file choice

4.3.1.1 Source management of an existing RPM and its patching

After expanding a selected RPM source file the user gets access to its packages. Following the selection of a package by the user, its source files appear in a window. After providing the links to the modified source files, the user can launch the generation of a patch for the selected package either via OBS Light commandline or via OBS Light UI. The patch is generated by OBS Light fully automatically within the chroot. The user can launch a build in order to obtain the patched RPM source file and a new version of the RPM bin file.

For beginner: Work in the chroot directory, try to change a header file and update and to observe the result.

4.3.1.2 Adding a new RPM

The user creates a new package by the means of patch generator of OBS Light after providing the links to the new source files. The new package is generated by OBS Light fully automatically within the chroot. The user can launch a build in order to obtain a new RPM source and new RPM bin file. The user has to specify the name of the new RPM in a text entry box before launching a build.

For beginner: Work in the chroot directory.

4.3.2 Available actions on packages

- Add a new package.
- Delete an existing package.
- Add a new source file to a package.
- Delete a source file from a package.
- Update a source file via a link.
- Launch the build of a package or launch the build of all packages.

4.3.3 Build parameters

The user must specify a target repository and a build architecture before launching a build. The combo box for the target repositories contains the repositories as specified above in 4.1. The acceptable build architectures are i586 or armv7.

4.4 The MIC2 image creation

4.4.1 RPM views and selection for the image creation

The user can see all the RPM files of a project together with their check boxes. The check box status (checked/unchecked) indicates whether the RPM file will be included during the MIC2 image creation. By default, all the check boxes are checked. Besides, the user has the possibility to specify a filter within the complete RPM list view, or see the RPM list by group. The user can see all the selected RPM in a separate view.

4.4.2 Available actions

The user can save and load his RPM configuration, i.e. the specified ensemble of the selected RPM and RPM groups, for future image creation purposes. Once the RPM configuration is specified, the user can launch the image creation. The user has the possibility to export the ks file generated for mic2 image creation to a local directory.

4.4.3 Image creation parameters

Before a launch of an image creation, the user has to specify via combo box a valid image type (fs, liveCD, liveUSB, loop, raw/KVM/QEMU, vmdk/VMWare, ubi, vdi/VirtualBox, mrstnand/Moorestown, nand, jffs2) and enter an image name in the text entry box.

4.4.4 The simulation of deployment of a created MIC2 image

The user can select an image in the list of available images and launch the image in a virtual machine.

4.5 The log files

OBS Light generates a log file of the RPM build and a log file of the image creation. The log files will provide the user with valuable information about the fail location of the build respectively of the image creation process. When submitting a bug for further analysis to the OBS Light bug list

<http://obslight/bugzilla>,

the user should always provide the log file in the attachment.

