KIWI Quick Start

Marcus Schäfer
Thomas Schraitle

$Id: MAIN.quickstart.kiwi.xml 3036 2008-01-23 15:14:58Z thomas-schraitle $

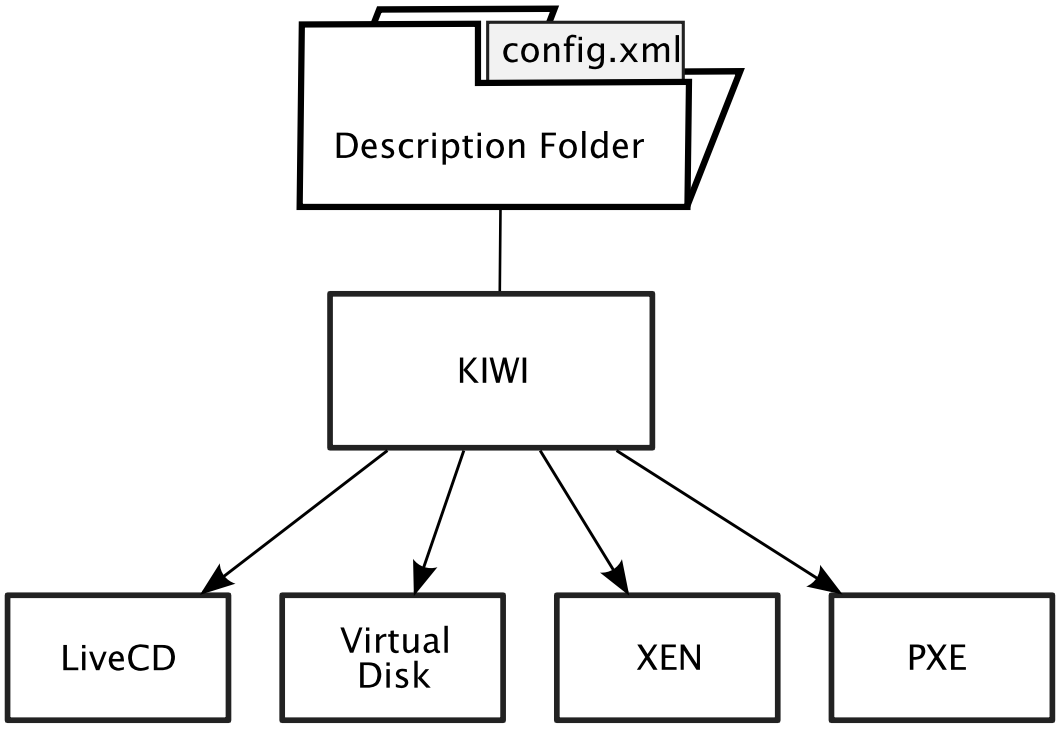
Created: 01/23/2008

Table of Contents

1. What is KIWI?

KIWI is a system for creating operating system images. An image is a directory with a file that represents the operating system, its applications and configurations and the filesystem structure of the OS and includes possible additional metadata. Depending on the image type, also disk geometry and partition table data can be part of the image. The image needs to be deployed at the destination system in order to get activated. Figure 1 shows you the general procedure that can be done with KIWI [kiwi].

Figure 1. KIWI Image Types



The image types in the above figure are:

LiveCD, also DVD, USB Stick
: deploy to media with KIWI and start your image.

Virtual Disk
: play full virtual systems with VMware, for exmaple.

XEN (Paravirtual Image for Hyphervisor)
: create guest configuration with KIWI.

PXE (Network Boot)
: provides boot environment with KIWI.

With all this in mind, KIWI provides the following advantages:

- Create your own Live CD/DVD which contains only those packages that you really need.
- Debug a new distribution by creating all necessary packages.
- Build your own installable ISO images to customize available packages, patterns, services, etc.
- Test new hardware with a predefinied image.
- Create KIWI images that contain software that is not included in the official distribution.
- Build KIWI images that can be directly used by virtualization systems like Xen or VMware.
- Generate KIWI images for different processor architectures only by switching to a different repository.
- Create smaller images than with a normal installation.
- Build images for network systems. KIWI supports you by establishing the infrastrucutre, for example PXE.

2. Background Information for “Fruit Lovers”

In order to create an image with KIWI it is required to create an image description first. A KIWI image description is a directory with one mandatory `config.xml` file and optional other files like configuration files or scripts.

Example descriptions can be found under `/usr/share/doc/packages/kiwi/examples/`. For detailed information about the content of an image description, look into the *System Design* document, see [kiwi.design].

3. What do I need?

To build images with KIWI, you need the following preconditions:

1. Free space: the more the better.
2. KIWI is split into several packages, targeted to different image types. In every case you need the base package, kiwi. Depending on your target image, you need the following additional packages:

Installation media	kiwi-desc-oemboot, kiwi-desc-vmxboot
Virtualization	kiwi-desc-vmxboot, kiwi-desc-xenboot
USB sticks	kiwi-desc-usbboot
Network client	kiwi-desc-netboot

3. Recommended distribution is openSUSE 10.3 or later.

4. Example files from the kiwi package:

   `/usr/share/doc/packages/kiwi/examples/test1`
   Is an openSUSE 10.3 KDE system, based on patterns.

   `/usr/share/doc/packages/kiwi/examples/test2`
   This is an openSUSE 10.3 console system, based on a package list.

   Both examples have the users `root` and `linux` with the password `linux`.

5. The documentation of the KIWI RELAX NG Schema. The KIWI configuration file is based on XML and the structure is documented in the package kiwi under `/usr/share/doc/packages/kiwi/kiwi.html`. The information therein is needed to know the correct elements and attributes to create a valid XML file.

4. Knowing KIWI's Build Process

The building process of KIWI is separated into three steps:

1. **Physical Extend.** This stage prepares the content of your new filesystem. During this stage, you determine which packages are installed on your image and which configuration files are included.

2. **Logical Extend.** This stage creates the operating system image based on the first step.

3. **Deployment.** The resulting ISO image can be deployed with different methods like installed on a hard disk or played by a virtualization system (VMware, Qemu, VirtualBox).

Detailed background can be found in the *System Design* paper, see `/usr/share/doc/packages/kiwi/kiwi.pdf`.

5. Creating a Graphical Live ISO CD

To create an ISO image, proceed as follows:

1. Install the package kiwi and resolve any dependencies.

2. Open a shell and become root.

3. Run KIWI with the following command to prepare the first phase ("physical extend"):

   ```
   kiwi --prepare /usr/share/doc/packages/kiwi/examples/test1 \
       --root /tmp/kiwi-test1
   ```

   If everything is correct, you see the message KIWI exited sucessfully.

4. Build the ISO image:

   ```
   kiwi --create /tmp/kiwi-test1 -d /tmp --type iso
   ```

   KIWI builds the ISO image. If everything is correct, it prints the message KIWI exited sucessfully.

5. Test the ISO image with Qemu, VMware or VirtualBox, for example:

   ```
   qemu -boot d -cdrom ISO_FILE -m 512
   ```

6. Creating a Minimal Image

If you want to start with a minimal example and extend it later, proceed as follows:

1. Copy the example directory from `test2` from your KIWI package:

```
cp -a /usr/share/doc/packages/kiwi/examples/test2/ .
```

2. Open the configuration file `test1/config.xml` and locate the XML element `packages`:

```
<packages>
  <package name="aaa_base"/>
  <package name="aaa_skel"/>
  <package name="acl"/>
  <package name="acpid"/>
</packages>
```

Essential Packages

The following lines are essential and *must* always be present:

```
<packages type="boot">
  <package name="filesystem"/>
  <package name="glibc-locale"/>
</packages>
```

3. Change the list of packages to your needs, but be aware to not remove essential ones like aaa_base, aaa_skel and others.

4. Save the configuration file.

5. Run the first phase (physical extend):

```
kiwi --prepare test1 --root /tmp/kiwi-test2
```

6. Run the second phase (logical extend):

```
kiwi --create /tmp/kiwi-test2 -d /tmp --type iso
```

7. Test the ISO image.

7. Creating Bootable USB Sticks

An USB stick with your Linux distribution is very convenient: you can carry it with you and it is faster than a hard disk solution. This section shows you how you can install an openSUSE system on your USB stick. For the installation the stick needs a miminum amount of 1 GByte.

Bootable USB Sticks and BIOS

You can only boot from the stick if the BIOS accepts your USB stick as bootable medium. Unfortunately, some BIOS types and USB sticks do not work together. It is nearly impossible to predict which combinations work, so if your USB stick cannot be booted, try another one from a different vendor.

Proceed as follows:

1. Install the packages kiwi-desc-usbboot and kiwi-desc-livesystem and resolve any dependecies.

2. Open a shell and become root.

3. Copy the directory:

```
cp /usr/share/kiwi/image/kwliveCD-suse-10.3  usb-live/
```

4. Open the configuration file `usb-live/config.xml` with your favorite editor and make the following changes:

a. Search for the first element `repository` beginning with:

```
<repository type="yast2">
     <source path="/mounts/dist/install/..."/>
</repository>
```

Disable it with a XML comment:

```
<!-- <repository type="yast2">
     <source path="/mounts/dist/install/..."/>
</repository> -->
```

b. Insert the following code on a new line after the disabled `</repository>` element:

```
<repository type="yast2">
  <source path="http://download.opensuse.org/distribution/10.3/repo/oss"/>
</repository>
<repository type="yast2">
  <source path="http://download.opensuse.org/distribution/10.3/repo/non-oss/"/>
</repository>
```

5. Run KIWI with the following command to prepare the first phase:

```
kiwi --prepare usb-live/ --root /tmp/kiwi-usblive \
   --add-profile KDE --logfile log-prepare.txt
```

This takes some time. If you want to see what is going on, remove the option `--logfile` or open a new shell, become root, change to your respective directory and type `tailf log-prepare.txt`.

6. Build the image:

```
mkdir kiwi-image
kiwi --type usb --create /tmp/kiwi-usblive/ -d kiwi-image --logfile log-build.txt
```

7. Install the image:

```
kiwi --bootstick kiwi-image/initrd-usbboot-suse-OSVERSION.ARCH-VERSION.gz \
  --bootstick-system kiwi-image/openSUSE-OSVERSION.ARCH-VERSION
```

Replace the placeholders with the appropriate values.

8. Creating Customized Images

This section describes different scenarios.

Creating a Developer Image

If you need an image for developing programs, you can add patterns in your configuration file. In this case, proceed as follows:

1. Copy the example directory from `test2` from your KIWI package:

```
cp -a /usr/share/doc/packages/kiwi/examples/test2/  devel/
```

2. Open the configuration file `devel/config.xml` and change the attribute `name` in element `image` to `devel`.

3. Add your patterns into the `packages` element but before `</packages>`:

```
<packages type="...">
```

```
  <packages name="aaa_base"/>
  <!-- More package elements follows -->
  <opensusePattern name="devel_basis"/>
  <!-- Add more patterns... -->
</packages>
```

Getting Pattern Names

You can get the names of patterns in your installed system under `/var/cache/zypp/raw/REPO/suse/setup/descr/` (replace *REPO* with the name of your system, for example `openSUSE-10.3-retail 10.3`. Watch for the space!) or look into the directory `suse/setup/descr/` on your DVD or first CD.

Depending what your interests in development are, you have to add more patterns. Table 1 contains an overview of some common pattern names.

Table 1. Examples of Pattern Names for Development

Pattern	Explanation
devel_C_C++	Tools and libraries for software development using C/C++ and other derivative of the C programming language
devel_basis	Minimal set of tools for compiling and linking applications
devel_ide	Integrated Development Environments
devel_java	Tools and libraries for software development using the Java programming language
devel_kernel	Tools for Linux kernel development
devel_python	Tools and libraries for software development using the Python programming language
devel_web	Tools and libraries for Web application development

4. Save the configuration file.
5. Run the first phase (physical extend):

   ```
   kiwi --prepare devel --root /tmp/kiwi-devel
   ```

6. Run the second phase (logical extend):

   ```
   kiwi --create /tmp/kiwi-devel -d /tmp --type iso
   ```

7. Test the ISO image.

Creating a Rescue System

A rescue system lets you debug your system. It contains tools that are required for these tasks. Mainly, the overall procedure is very similar to the last one. Proceed as follows:

1. Install on your system the packages aufs and squashfs.
2. Copy the example directory from `test2` from your KIWI package:

   ```
   cp -a /usr/share/doc/packages/kiwi/examples/test2/ rescue/
   ```

3. Open the configuration file `devel/config.xml` and change the attribute `name` in element `image` to `rescue`.

4. Add your packages into `config.xml`. Insert before the `</packages>` end tag the following packages:

Table 2. Examples of Pattern Names for Development

Pattern	Explanation
dar	Backup and Restore Application
ddrescue	Data Copying in the Presence of I/O Errors
dmraid	A Device-Mapper Software RAID Support Tool
fuse	User space File System
fuseiso	FUSE module to mount CD-ROM images with ISO9660 filesystems in them
fusesmb	SMB for FUSE
hfsutils	Tools Used for the Macintosh's Hierarchical File System
jfsutils	IBM JFS Utility Programs
mtools	floppy daemon for remote access to floppy drive
ntfsprogs	NTFS filesystem libraries and utilities
parted	GNU partitioner
scpm	System Configuration Profile Management
squashfs	A Read-Only File System with Efficient Compression
sshfs	Filesystem client based on SSH file transfer protocol
sudo	Execute some commands as root
udftools	UDF filesystem tools
xfsdump	Administrative Utilities for the XFS File System
xfsprogs	Utilities for managing the XFS file system

5. Save the configuration file.

6. Run the first phase (physical extend):

```
kiwi --prepare devel --root /tmp/kiwi-devel
```

7. Run the second phase (logical extend):

```
kiwi --create /tmp/kiwi-devel -d /tmp --type iso
```

8. Test the ISO image.

9. Additional Information

[kiwi] *KIWI Image System*. Marcus Schäfer. WWW: `http://developer.berlios.de/projects/kiwi/`.

[kiwi.design] *KIWI Image System*. System Design. Marcus Schäfer. file: `file:///usr/share/doc/packages/kiwi/kiwi.pdf`.

[packages] *KIWI Packages*. WWW: `http://download.opensuse.org/repositories/openSUSE:/Tools/`.

[xen] *Xen Source*. WWW: `http://www.xensource.com`.

[pxe] *Preboot Execution Environment*. WWW: `http://de.wikipedia.org/wiki/Preboot_Execution_Environment`.