

# Building Images With KIWI

**Robert Schweikert**

SUSE-IBM Software Integration Center Tech Lead

[rjschwei@suse.com](mailto:rjschwei@suse.com)



# Outline

- Preliminaries
  - What is it
  - Documentation resources
- Using kiwi
  - High-level overview
  - The KIWI configuration tree
  - Exploring image types
  - What is an appliance
- KIWI Usage and Support
- Summary
- Questions, Thoughts, Comments

# Preliminaries

# What is KIWI



- It's a fruit
- It's a bird
- Someone from New Zealand
- It's a Linux image build system
  - KIWI is an open source project hosted on github  
<https://github.com/openSUSE/kiwi>
  - Licensed under GPL v2
  - Mostly written in Perl
  - Sponsored by SUSE®
  - Backend of SUSE Studio (<http://www.susestudio.com>)



# Setting up kiwi

- Two versions you might consider
  - Version released with your distribution
  - Weekly releases
- Distribution version
  - SUSE<sup>®</sup> Linux Enterprise – you will find KIWI in the SDK
  - openSUSE – KIWI is part of the standard distribution
- Weekly release
  - openSUSE Build Service (OBS)
    - Virtualization:Appliances
- In YaST or via zypper search for **kiwi**, install packages

# KIWI Build Environment

- kiwi is a command line tool
  - SUSE® Studio (Online or Onsite) is web app provides GUI
- Most kiwi commands require root privileges
  - Need access to package management
  - Need access to user management
  - Need to create system configuration
- **Be cautious with kiwi commands you are root**
  - Dedicated build system or VM



# Documentation Resources and Help

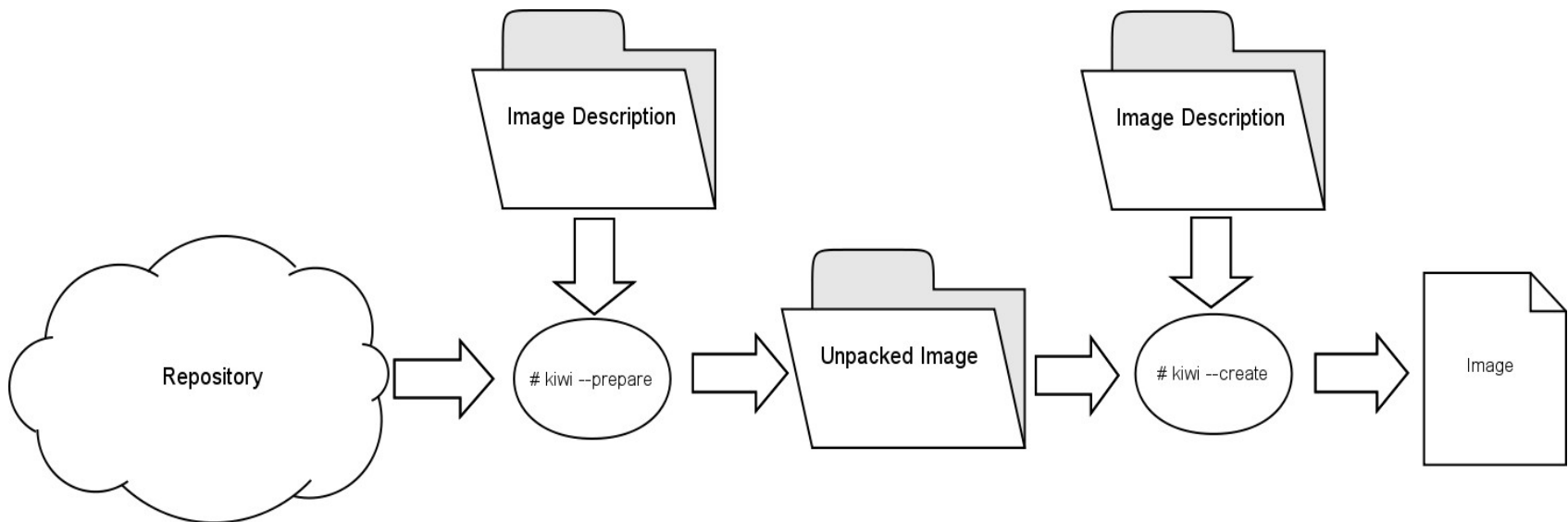
- KIWI User's Manual a.k.a Cookbook
  - <http://doc.opensuse.org/projects/kiwi/doc/>
  - /usr/share/doc/packages/kiwi/kiwi.pdf (kiwi-doc package)
- KIWI Examples
  - <http://en.opensuse.org/Portal:KIWI>
- Configuration File Schema
  - <http://doc.opensuse.org/projects/kiwi/schema-doc/>
- Mailing List
  - <http://groups.google.com/group/kiwi-images>
- IRC (irc.freenode.net)
  - #opensuse-kiwi

Using kiwi



# High-level Overview

- Two step process
  - Prepare step
    - Output is the unpacked image tree (directory)
  - Create step
    - Output is the desired image



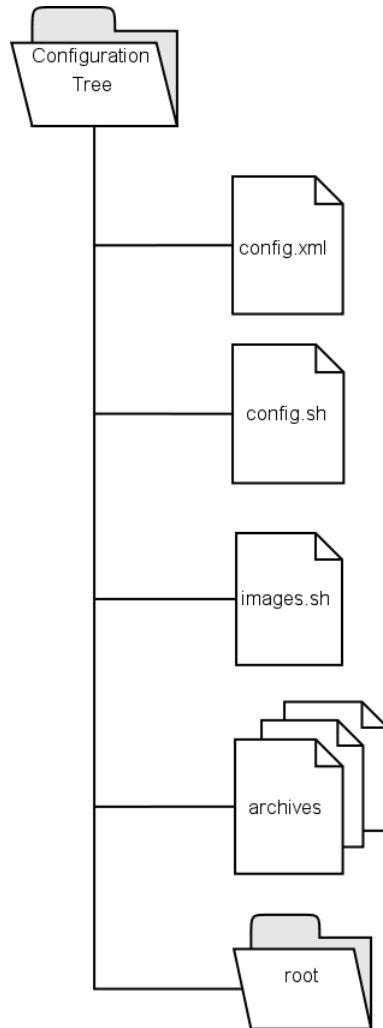
# High-level Overview – Prepare Step

- Read the configuration
  - Initialize repositories
  - Install packages into unpacked image tree
  - Apply overlay files
  - Run a customizable script (config.sh)
- 
- Generic command
    - `kiwi --prepare PATH_TO_CONFIG --root PATH_TO_DEST`

# High-level Overview – Create Step

- Read configuration information (from unpacked img)
- Run customizable script (images.sh)
- Create the initrd
  - initrd is another image description supplied by KIWI packages
  - kiwi-desc-\*-boot packages
- Build image in specified type using unpacked image tree and the created initrd

# The Configuration Tree



- Configuration is a directory structure (tree)

- config.xml → required
- config.sh → optional
- images.sh → optional
- archives → optional
- root → optional

# Configuration Details

- **config.xml** – required
  - Configuration file for image attributes, type, users, packages
- **config.sh** – optional
  - Shell script executed at root level of unpacked tree at the end of prepare. Customize set up, fiddle with files
- **images.sh** – optional
  - Shell script executed at root level of unpacked tree at the beginning of the create step. Remove drivers, modify based on image type



# Configuration Details

- **archives** – optional
  - One or more archives in known format, tar, tar.gz, tar.bz2 etc.
  - Name specified in config.xml with <archive>
  - Extracted at root level of unpacked image tree prior to execution of config.sh
- **root** – optional directory
  - Root file system representation. This directory is referred to as overlay tree, all files and directories copied to unpacked image tree in the location found in this directory.

# config.xml

- Contains the image description
  - XML validated against schema
  - Checked for consistency of data
  - When we update the schema we provide XSLT based upgrade functionality
  - Upgrade is automatic

# config.xml – The Guts

```
<?xml version="1.0" encoding="utf-8"?>
<image schemaversion="" name="">
  <description type="system">
    <author></author>
    <contact></contact>
    <specification></specification>
  </description>
  <preferences>
    <type image="" primary="true" filesystem="" boot="" format=""/>
    <version>1.0.0</version>
    <packagemanager>zypper</packagemanager>
    <rpm-check-signatures>>false</rpm-check-signatures>
    <rpm-force>>true</rpm-force>
    <locale>en_US</locale>
    <keytable>us.map.gz</keytable>
  </preferences>
```

# config.xml – The Guts

```
<users group="root">
  <user pwd="" home="/root" name="root"/>
</users>
<repository type="yast2">
  <source path=""/>
</repository>
<packages type="image">
  <package name=""/>
</packages>
<packages type="bootstrap">
  <package name="filesystem"/>
  <package name="glibc-locale"/>
</packages>
</image>
```

# XML Schema

- Implemented in RELAX NG (relaxing) compact notation
  - <http://www.relaxng.org>
  - <http://www.relaxng.org/tutorial-20011203.html>
  - [http://en.wikipedia.org/wiki/RELAX\\_NG](http://en.wikipedia.org/wiki/RELAX_NG)
- Schema located in
  - /usr/share/kiwi/modules/KIWISchema.rnc
  - Installed as part of the kiwi package
- Parser messages are not always the most useful
  - Jing is helpful in getting better error messages



# Image Types

- Specified by value of “image” attribute of `<type>` element in config.xml
- Amazon EC2 (ec2)
  - Support both S3 and EBS backed images
  - Need to also specify settings with `<ec2config>` element
  - Requires install of Amazon tools
  - For S3 result is an AMI that can be uploaded
  - For EBS result is a disk image that can be dumped onto a volume in EC2 via dd\_rescue

# Image Types

- Self installing pre-load (oem)
  - Image within an image
    - Inner image is the one configured
    - Outer image is a “live system” that automatically dumps image to target storage
  - Use <oemconfig> child to configure various aspects
  - Partitioning via LVM use <systemdisk> element
  - Create install media for USB stick or optical (CD/DVD)

# Image Types

- Network install (pxe)
  - Boot image and system image are separated
  - Use requires pxe infrastructure
    - DHCP server
    - atftp server
  - Client configuration via config.<MAC\_ADDRESS>

# Image Types

- Live image (iso or USB)
  - Iso is separate type specified in config.xml
    - Result is a .iso file than can be burned to optical (CD/DVD)
  - USB
    - Special build, i.e. config.xml specified use “kiwi –bootstick” to deploy
    - Use regular disk image and dump to stick with “dd” command
      - Do not forget “bs=32k” on dd command or the stick will not boot

# Image Types

- Virtual images (vmx)
  - ec2 → Amazon AMI
  - ovf → Open Virtualization Format
  - ova → Modified ovf
  - qcow2 → Native qemu format
  - vmdk → Native VMWare format also generate .vmx file
  - vhd → Hyper-V format
  - vhd-fixed → Specialized Hyper-V format
  - raw → raw disk image
    - Kiwi always produces a raw disk image
    - No configuration needed

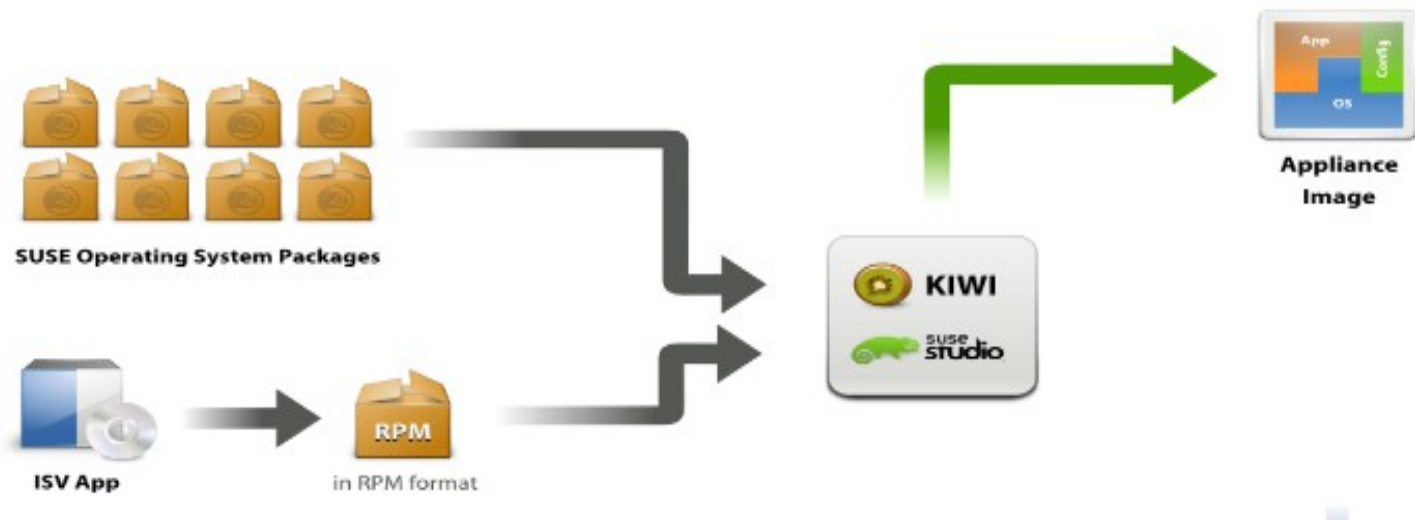


# Image Types

- Xen (xen)
  - Build guest or host
    - Host
      - Set “**bootprofile**” attribute to “**xen**”
      - Set “**bootkernel**” attribute to “**xenk**”
      - Set “**domain**” attribute <**xenconfig**> to “**dom0**”
    - Need to install the kernel-xen and other xen packages
    - Generates .xenconfig file

# What is an Appliance

- An appliance is the combination of the OS, an application, and any middle ware plus the appropriate configuration and optimization settings in one package



# What is an Appliance

- Another delivery method for an application
  - Saves install effort at end customer (pre-installed)
  - Saves configuration and tuning effort at customer
- 3 form factors
  - Hardware appliance
  - Software appliance
  - Virtual appliance

# KIWI Usage and Support

# KIWI Usage and Support

- Usage
  - KIWI is the backend of SUSE® Studio (Online and Onsite)
  - KIWI is the backend of SUSE Linux Enterprise Point of Service Image builder
  - KIWI is integrated with OBS to build images
- Support
  - Best effort community support on the mailing list
  - SUSE Support with
    - SUSE Studio Onsite
    - SUSE Linux Enterprise Point of Service



# Summary

# Summary

- Kiwi is a very flexible image build tools
- Command line driven
  - Easy to integrate into existing build processes
- Support many image formats
  - Can build multiple image formats from one configuration
- Open source licensed under GPL v2
- Mailing list:  
<http://groups.google.com/group/kiwi-images>
- IRC: #opensuse-kiwi on irc.freenode.net

Questions, Thoughts, Comments





## **Unpublished Work of SUSE. All Rights Reserved.**

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE.

Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE.

Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

## **General Disclaimer**

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

