

System Proposal

YaST2 startup concept SUSE LINUX AG

Marcus Schaefer, Marcus.Schaefer@suse.com *



Author: Marcus Schaefer
Date: March 8, 2006

*SUSE LINUX AG, Maxfeldstrasse 5, D-90409 Nuremberg

Contents

1	Introduction	5
1.1	CVS directory structure	5
1.2	Log-File y2start.log	6
1.3	Special file /etc/install.inf	6
1.4	Level numbering	6
1.5	Basic workflow levels	7
2	YaST2.First-Stage	9
2.1	First-Stage Hooks	9
2.2	First-Stage main script	9
2.3	First-Stage level scripts	10
3	YaST2.Second-Stage	11
3.1	Second-Stage Hooks	11
3.2	Second-Stage main script	11
3.3	Second-Stage level scripts	12
4	YaST2.call	15
4.1	Call-Stage Hooks	16
4.2	Medium prequalification	16
4.3	Medium selection check and fallback	17
5	Possible installation methods	19

1 Introduction

YaST provides many different installation methods and there may be some more special cases of installation methods in the future. This draft is describing a concept of how to start/restart the installation within the different environments. The current implementation provides a clean solution working on single stages which handles a predefined set of tasks. If there are more tasks to perform one need to verify first which stage is responsible to hold the implementation and should update the documentation before implementing the code.

1.1 CVS directory structure

The main directory storing all currently used scripts within the CVS tree is at **source/installation/scripts/**. Below this directory the following structure will be available:

- **startup**
Root directory for all startup related information and scripts
- **startup/common/**
Contains all common used code as functions.
- **startup/arch/<architecture>/**
Contains all architecture dependent code. I would strongly recommend to have this code as functions too.
- **startup/hooks/<hookdirs>**
Contains all hook scripts which are called on demand. The scripts are stored in one of the directories named: *preFirstStage*, *postFirstStage*, *preFirstCall*, *postFirstCall*, *preSecondStage*, *postSecondStage*, *preSecondCall*, *postSecondCall*
- **startup/First-Stage/**
Contains all first stage level scripts including the prefix **FXX-**
- **startup/Second-Stage/**
Contains all second stage level scripts including the prefix **SXX-**

1.2 Log-File y2start.log

Any message which could be written for later interpretation has to be saved to one single file named **var/log/YaST2/y2start.log**. This file contains information about the startup process only. The format of the file is as follows:

```
Stage [<Level-ID>] <comment>
                |– <subcomment>
                |– <subcomment>
...

```

1.2.1 Renaming of files

Some files should be renamed to reflect the workflow we want to follow:

- **YaST2.start**
Renamed to **YaST2.First-Stage**
- **YaST2.firstboot**
Renamed to **YaST2.Second-Stage**
- **YaST2**
Renamed to **YaST2.call**

1.3 Special file /etc/install.inf

The file */etc/install.inf* should contain all information about the installation environment and all information needed to continue the installation if it has been stopped for some reasons. There should be no other file for saving information handled within one of the installation scripts.

1.4 Level numbering

I would suggest every implementation following this concept to include a descriptive comment with a level ID referring this documentation. The basic startup scripts called **YaST2.First-Stage** and **YaST2.Second-Stage** are responsible for calling the single stage scripts which are saved below the directories **First-Stage** and **Second-Stage**. The naming of the stage scripts consists of a prefix letter followed by a number and a short name:

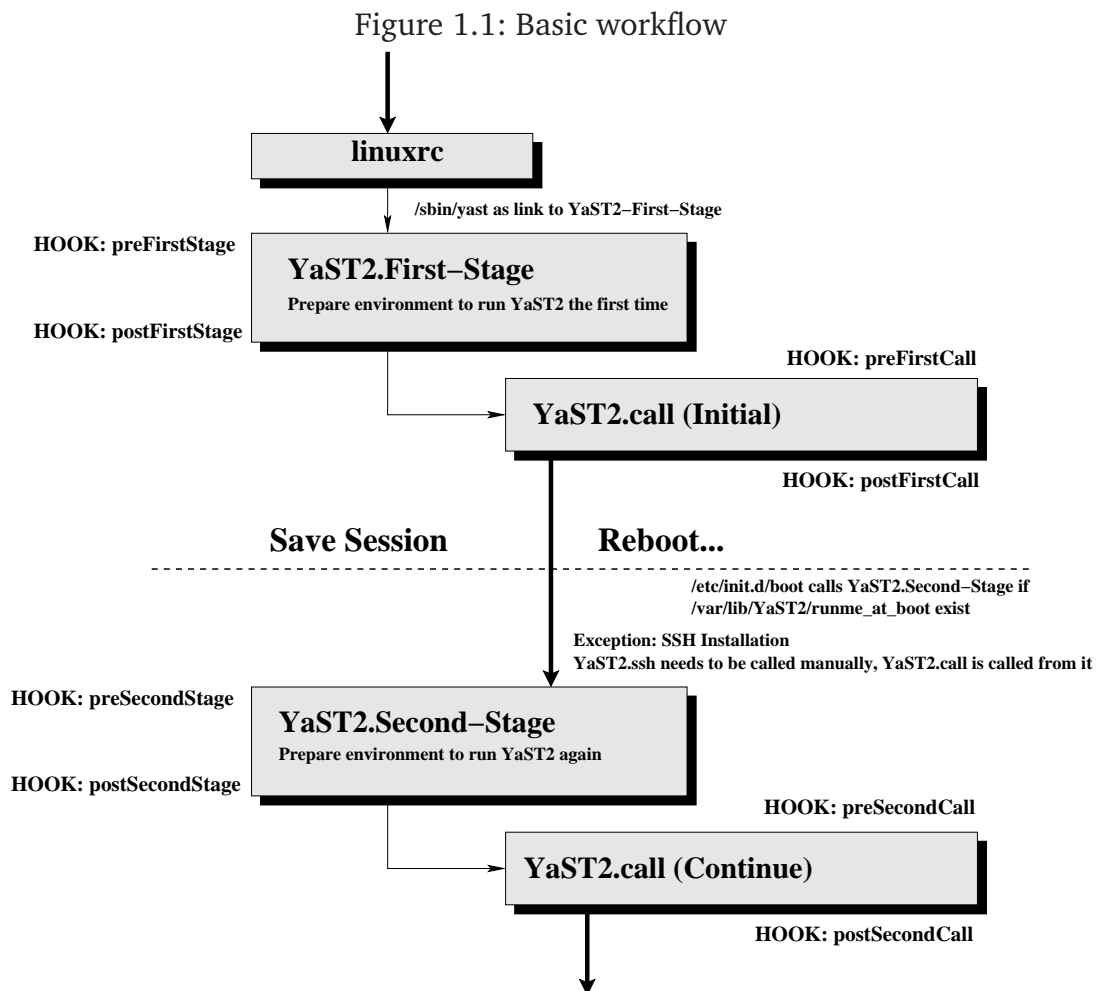
First/Second-Stage script naming convention

F<number>-**name**
S<number>-**name**

Hook scripts naming convention

%<number>-**name**

1.5 Basic workflow levels



2 YaST2.First-Stage

The YaST2.First-Stage script is the first script called for any installation process. `linuxrc` will call `/sbin/yast` which is a link to the YaST2.First-Stage script. Usually the system will reboot after this script has finished.

2.1 First-Stage Hooks

The following First-Stage hook directories are checked:

- **preFirstStage**
Within the `ins-sys` each script stored in the `preFirstStage` directory is called directly in front of the YaST2.First-Stage script
- **postFirstStage**
Within the `ins-sys` each script stored in the `postFirstStage` directory is called directly after the YaST2.First-Stage script has been finished

2.2 First-Stage main script

The following tasks are handled in the YaST2.First-Stage script

1. Include common used and stage functions
2. Source `/etc/install.inf`
3. Setup `ARCH_*` variables
4. Export important variables for use in subshells
5. Initialize `y2start.log`
6. Create stage list and call level scripts

2.3 First-Stage level scripts

The following tasks are handled in the specified level scripts

2.3.1 F02-hwinfo

1. Start hardware detection

2.3.2 F03-language

2. Set Language environment
3. Start Unicode mode

2.3.3 F04-cmdline

4. Handle kernel parameters

2.3.4 F05-terminal

5. Init virtual consoles (initviocons)

2.3.5 F06-config

6. Create mtab according to /proc/mounts

2.3.6 F07-logging

7. Set logging size

2.3.7 F08-start

8. Call YaST2.call (initial)
9. Save log file y2start.log to installed system

2.3.8 F09-cleanup

10. Clean sweep (umount, etc...)
11. Write YaST2 exit code to /tmp/YaST2-First-Stage-Exit-Code

3 YaST2.Second-Stage

The YaST2.Second-Stage script is called via `/etc/init.d/boot` and is used to finish the installation process. It will call YaST2 in the so called *continue* mode. The script itself is called from a installed system environment.

3.1 Second-Stage Hooks

The following Second-Stage hook directories are checked:

- **preSecondStage**
Within the installed system each script stored in the `preSecondStage` directory is called directly in front of the YaST2.Second-Stage script
- **postSecondStage**
Within the installed system each script stored in the `postSecondStage` directory is called directly after the YaST2.Second-Stage script has been finished

3.2 Second-Stage main script

The following tasks are handled in the YaST2.Second-Stage script

1. Include common used and stage functions
2. Source `/etc/install.inf`
3. Setup `ARCH_*` variables
4. Prepare reset of manpages,info dirs
5. Continue logging to `y2start.log`
6. Create stage list and call level scripts

3.3 Second-Stage level scripts

The following tasks are handled in the specified level scripts

3.3.1 S02-logging

1. Start syslog daemon and set debug and logging options

3.3.2 S03-autoconf

2. If exists, handle autoconf.xml and **call YaST2.call (continue)**

3.3.3 S04-language

3. Set Language environment
4. Start Unicode mode

3.3.4 S05-config

5. Provide configurations for the subsystems: hotplug, USB, pcmcia

3.3.5 S06-services

6. Start subsystems if needed: hotplug, pcmcia

3.3.6 S07-medium

7. check installation medium related install.inf variables
 - Start shell on second virtual console. Related to: NoShell
 - Activate network and sshd for SSH or VNC installations. Related to: Display_IP VNC UseSSH
 - Prepare for VNC installation. Related to: VNC
 - Prepare for SSH installation. Related to: UseSSH

3.3.7 S08-start

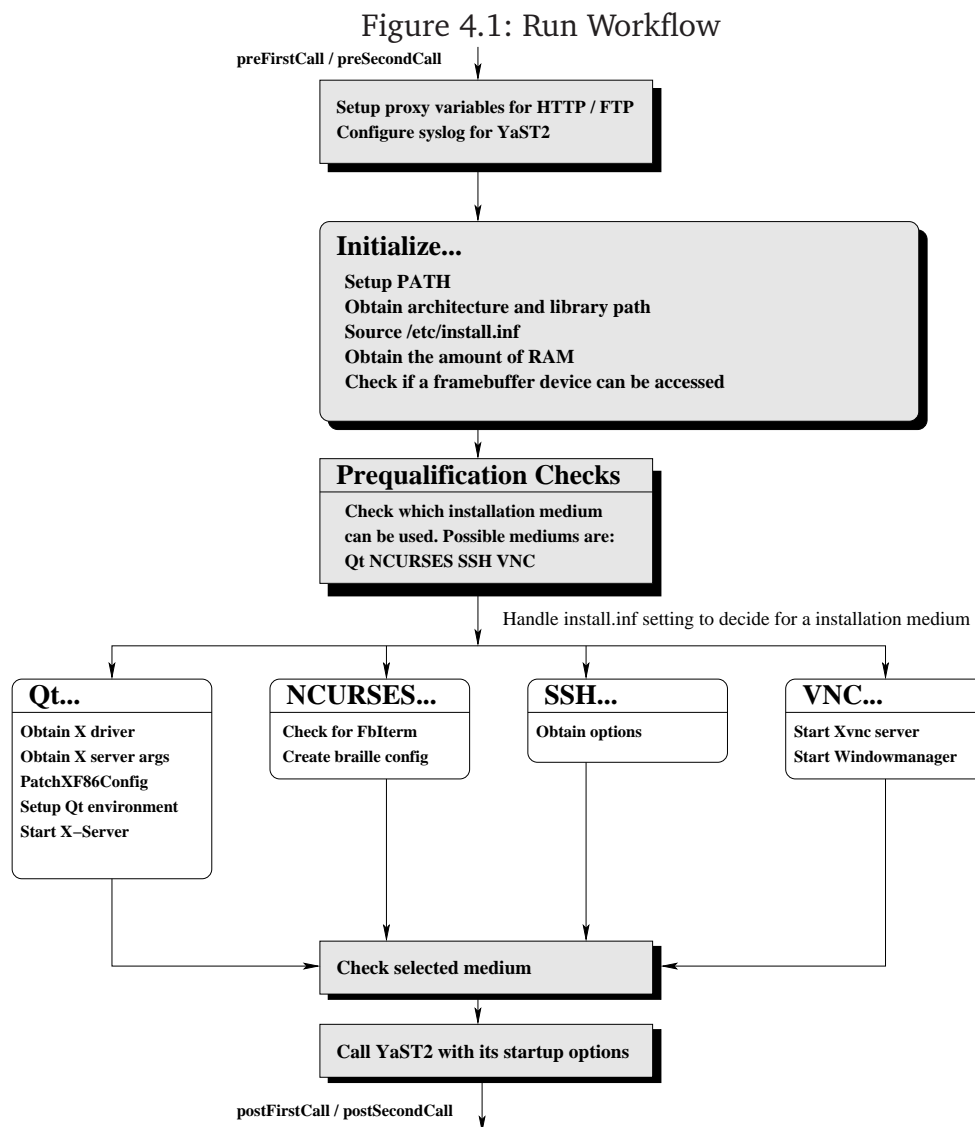
8. **Call YaST2.call (continue)** or wait for manual startup if SSH installation was requested

3.3.8 S09-cleanup

9. Clean sweep, kill X-Server processes, stop services stop unicode mode

4 YaST2.call

YaST2.call works in the First-Stage environment as well as in the Second-Stage environment. The YaST2.call script includes all the tasks which needs to be handled independent of the environment. YaST2.call requires a correct prepared environment done within the YaST2-First-Stage or YaST2-Second-Stage scripts. Its major task is to start YaST2. The following workflow shows how YaST2.call works.



4.1 Call-Stage Hooks

The following Call-Stage hook directories are checked:

- **preFirstCall**
Within the inst-sys each script stored in the preFirstCall directory is called directly in front of the YaST2.call script
- **postFirstCall**
Within the ins-sys each script stored in the postFirstCall directory is called directly after the YaST2.call script has been finished
- **preSecondCall**
Within the installed system each script stored in the preSecondCall directory is called directly in front of the the YaST2.call script
- **postSecondCall**
Within the installed system each script stored in the postSecondCall directory is called directly after the YaST2.call script has been finished

4.2 Medium prequalification

There are four different mediums to use for an installation. Each of them have a view needs which should be checked first to know about the possible mediums to use.

- Qt
 1. Qt plugins are needed
 2. Appropriate X driver module must be found
 3. Memory requirements must be fulfilled
 4. X-Server needs to be started if there is no DISPLAY to access either
 5. /etc/X11/xorg.conf must exist
 6. /usr/X11R6/share/fvwm/fvwmrc.yast2 must exist
- NCURSES
There are no prerequisites for ncurses mode
- SSH
 1. sshd must be running
- VNC
 1. /usr/X11R6/share/fvwm/fvwmrc.yast2 must exist

4.3 Medium selection check and fallback

The default medium is Qt but the medium can be changed with options given to the bootmanager. The options are passed to the kernel, linuxrc will handle it and provide the options not handled within the file `/etc/install.inf`. Referring this information one of the possible mediums is selected. If the medium cannot be selected because of a missing prerequisite the NCURSES fallback is used.

If the prequirements for the selected medium is ok we will handle the installation medium according to the workflow above. After the medium has been prepared we need to check the medium again:

- Qt
 1. If an X-Server must be started, check if the server is running and accessable
- NCURSES

There are no checks for the ncurses mode
- SSH
 1. The network interface has to be reachable
- VNC
 1. The Xvnc server must be running

If one of the medium selection checks failed this should be handled as a fatal error and should be shown in a descriptive error message.

5 Possible installation methods

- **Standard Fbdev based, UI:Qt**
Insert bootable CD and simply start the installation without any options set
- **Standard vmware based, UI:Qt**
Install system within a virtual machine. vmware boots up from CD and the installation is started without any further options. The vmware graphics driver will be used.
- **Extended vesa driver based, UI:Qt**
Install with special graphics hardware which doesn't support a frame-buffer device, for example i810, glint. In this case the vesa driver is used. To test this mode just use **vga=normal** on the kernel command-line
- **Extended accelerated X-driver based, UI:Qt**
Set the boot paramater **acceleratedx=1**. In this case an accelerated X11 driver will be used if one of the following drivers can handle the card:
 - ati
 - r128
 - radeon
 - nv

If the card is not supported from one of the listed drivers the default framebuffer or vesa configuration is used.
- **Textmode with fbdev active, UI:ncurses**
Set the boot paramater **textmode=1**. In this case the YaST2 GUI will start in ncurses mode. The kernel framebuffer is active if supported.
- **Textmode with vga active, UI:ncurses**
Select **F2 -> TextMode**. In this case the kernel framebuffer is switched off and the GUI will start in ncurses mode.
- **Remote via VNC, UI:Qt**
Set the boot paramater **vnc=1 vncpassword=12345678**. Use the vncviewer on the remote side to connect to the VNC server. Additional information how to connect will be printed to the textconsole.

- **Remote via SSH, UI:ncurses**
Set the boot parameter **usessh=1 sshpassword=12345**. After first re-boot you need to login again and call YaST2.ssh which calls YaST2.call in continue mode to finish the installation.
- **Serielle Konsole, UI:ncurses**
Set the boot parameter **console=ttyS0,115200** and plug in a serial cable between the serial interfaces of the two computers. Now call **screen /dev/ttyS0 115200** on the remote side and choose the appropriate terminal type from the list (normally type 7).
- **Remote Display access, UI:Qt**
Set the boot parameter **Display_IP=[IP or Name]** to an IP address or hostname whereas the corresponding machine has to accept X11 connections via port 6000