18 Deploying Customized Preinstallations

Rolling out customized preinstallations of SUSE Linux Enterprise Server to many identical machines spares you from installing each one of them separately and provides a standardized installation for the end users. With YaST Firstboot, create customized preinstallation images and determine the workflow for the final personalizing steps that involve end user interaction (as opposed to AutoYaST, which allows completely automated installations; for more information, see *Chapter 19, Automated Installation*).

Creating a custom installation, rolling it out to your hardware, and personalizing the final product involves the following steps:

1. Prepare the master machine whose disk needs to be cloned to the client machines. For more information, refer to *Section 18.1, "Preparing the Master Machine"*.

2. Customize the firstboot workflow. For more information, refer to *Section 18.2, "Customizing the Firstboot Installation"*.

3. Clone the master machine's disk and roll this image out to the clients' disks. For more information, refer to *Section 18.3, "Cloning the Master Installation"*.

4. Have the end user personalize the instance of SUSE Linux Enterprise Server. For more information, refer to *Section 18.4, "Personalizing the Installation"*.

18.1 Preparing the Master Machine

To prepare a master machine for a firstboot workflow, proceed as follows:

1. Insert the installation media into the master machine.

2. Boot the machine.

3. Perform a normal installation including all necessary configuration steps and wait for the installed machine to boot. Also install the `yast2-firstboot` package.

4. To define your own workflow of YaST configuration steps for the end user or to add your own YaST modules to this workflow, proceed to *Section 18.2, "Customizing the Firstboot Installation"*. Otherwise proceed directly to *Step 5*.

5. Enable firstboot as `root`:

   Create an empty file `/var/lib/YaST2/reconfig_system` to trigger firstboot's execution. This file will be deleted after the firstboot configuration has been successfully accomplished. Create this file using the following command:

   ```
   touch /var/lib/YaST2/reconfig_system
   ```

6. Proceed to *Section 18.3, "Cloning the Master Installation"*.

18.2 Customizing the Firstboot Installation

Customizing the firstboot installation workflow may involve several different components. Customizing them is optional. If you do not make any changes, firstboot performs the installation using the default settings. The following options are available:

- Customizing messages to the user, as described in *Section 18.2.1, "Customizing YaST Messages"*.

- Customizing licenses and license actions, as described in *Section 18.2.2, "Customizing the License Action"*.

- Customizing the release notes to display, as described in *Section 18.2.3, "Customizing the Release Notes"*.

- Customizing the order and number of components involved in the installation, as described in *Section 18.2.4, "Customizing the Workflow"*.

- Configuring additional optional scripts, as described in *Section 18.2.5, "Configuring Additional Scripts"*.

To customize any of these components, modify the following configuration files:

`/etc/sysconfig/firstboot`
> Configure various aspects of firstboot (such as release notes, scripts, and license actions).

`/etc/YaST2/firstboot.xml`
> Configure the installation workflow by enabling or disabling components or adding custom ones.
>
> Provide translations for such a customized installation workflow, as described in *Section 18.2.6, "Providing Translations of the Installation Workflow"*.

> 💡 **Tip: Alternative Location of the Control File**
>
> `/etc/YaST2/firstboot.xml` is the default path for the control file, installed by the `yast2-firstboot` package. If you need to define a different location for the control file, edit `/etc/sysconfig/firstboot`, and change the `FIRSTBOOT_CONTROL_FILE` variable to your preferred location.

If you want to customize more than the workflow components, refer to the `control.xml` documentation at http://doc.opensuse.org/projects/YaST/SLES11/tdg/inst_in_general_chap.html#product_control.

18.2.1 Customizing YaST Messages

By default, an installation of SUSE Linux Enterprise Server contains several default messages that are localized and displayed at certain stages of the installation process. These include a welcome message, a license message, and a congratulatory message at the end of installation. You can replace any of these with your own versions and include localized versions of them in the installation. To include your own welcome message, proceed as follows:

1. Log in as `root`.

2. Open the `/etc/sysconfig/firstboot` configuration file and apply the following changes:

   a. Set `FIRSTBOOT_WELCOME_DIR` to the directory path where you want to store the files containing the welcome message and the localized versions, for example:

   ```
   FIRSTBOOT_WELCOME_DIR="/usr/share/firstboot/"
   ```

   b. If your welcome message has file names other than `welcome.txt` and `welcome_locale.txt` (where `locale` matches the ISO 639 language codes such as "cs" or "de"), specify the file name pattern in `FIRSTBOOT_WELCOME_PATTERNS`. For example:

   ```
   FIRSTBOOT_WELCOME_PATTERNS="mywelcome.txt"
   ```

   If unset, the default value of `welcome.txt` is assumed.

3. Create the welcome file and the localized versions and place them in the directory specified in the `/etc/sysconfig/firstboot` configuration file.

Proceed in a similar way to configure customized license and finish messages. These variables are `FIRSTBOOT_LICENSE_DIR` and `FIRSTBOOT_FINISH_FILE`.

Change the `SHOW_Y2CC_CHECKBOX` to "yes" if the user needs to be able to start YaST directly after performing the installation.

18.2.2   Customizing the License Action

You can customize the way the installation system reacts to a user's refusal to accept the license agreement. There are three different ways which the system could react to this scenario:

halt

The firstboot installation is aborted and the entire system shuts down. This is the default setting.

continue

The firstboot installation continues.

abort

The firstboot installation is aborted, but the system attempts to boot.

Make your choice and set `LICENSE_REFUSAL_ACTION` to the appropriate value.

18.2.3   Customizing the Release Notes

Depending on if you have changed the instance of SUSE Linux Enterprise Server you are deploying with firstboot, you probably need to educate the end users about important aspects of their new operating system. A standard installation uses release notes (displayed during one of the final stages of the installation) to provide important information to the users. To have your own modified release notes displayed as part of a firstboot installation, proceed as follows:

1. Create your own release notes file. Use the RTF format as in the example file in `/usr/share/doc/release-notes` and save the result as `RELEASE-NOTES.en.rtf` (for English).

2. Store optional localized versions next to the original version and replace the `en` part of the file name with the actual ISO 639 language code, such as `de` for German.

3. Open the firstboot configuration file from `/etc/sysconfig/firstboot` and set `FIRSTBOOT_RELEASE_NOTES_PATH` to the actual directory where the release notes files are stored.

18.2.4   Customizing the Workflow

By default, a standard firstboot workflow includes the following components:

- Language Selection

- Welcome

- License Agreement

- Host Name

- Network

- Time and Date

- Desktop

- root Password

- User Authentication Method

- User Management

- Hardware Configuration

- Finish Setup

This standard layout of a firstboot installation workflow is not mandatory. You can enable or disable certain components or integrate your own modules into the workflow. To modify the firstboot workflow, manually edit the firstboot configuration file `/etc/YaST2/firstboot.xml`. This XML file is a subset of the standard `control.xml` file that is used by YaST to control the installation workflow.

For an overview about proposals, see *Example 18.1, "Configuring the Proposal Screens"*. This provides you with enough background to modify the firstboot installation workflow. The basic syntax of the firstboot configuration file (plus how the key elements are configured) is explained with this example.

EXAMPLE 18.1: CONFIGURING THE PROPOSAL SCREENS

```
    …
    <proposals config:type="list"> ❶
        <proposal> ❷
            <name>firstboot_hardware</name> ❸
            <mode>installation</mode> ❹
            <stage>firstboot</stage> ❺
            <label>Hardware Configuration</label> ❻
            <proposal_modules config:type="list"> ❼
                <proposal_module>printer</proposal_module> ❽
            </proposal_modules>
        </proposal>
        <proposal>
        …
        </proposal>
    </proposals>
```

❶ The container for all proposals that should be part of the firstboot workflow.

❷ The container for an individual proposal.

❸ The internal name of the proposal.

❹ The mode of this proposal. Do not make any changes here. For a firstboot installation, this must be set to `installation`.

❺ The stage of the installation process at which this proposal is invoked. Do not make any changes here. For a firstboot installation, this must be set to `firstboot`.

❻ The label to be displayed on the proposal.

❼ The container for all modules that are part of the proposal screen.

❽ One or more modules that are part of the proposal screen.

The next section of the firstboot configuration file consists of the workflow definition. All modules that should be part of the firstboot installation workflow must be listed here.

EXAMPLE 18.2: CONFIGURING THE WORKFLOW SECTION

```
    <workflows  config:type="list">
        <workflow>
            <defaults>
```

```
                <enable_back>yes</enable_back>
                <enable_next>yes</enable_next>
                <archs>all</archs>
            </defaults>
            <stage>firstboot</stage>
            <label>Configuration</label>
            <mode>installation</mode>
            … <!–– list of modules  ––>
            </modules>
        </workflow>
    </workflows>
    …
```

The overall structure of the `workflows` section is very similar to that of the `proposals` section. A container holds the workflow elements and the workflow elements all include stage, label and mode information (just as the proposals introduced in *Example 18.1, "Configuring the Proposal Screens"*). The most notable difference is the `defaults` section, which contains basic design information for the workflow components:

`enable_back`
> Include the *Back* button in all dialogs.

`enable_next`
> Include the *Next* button in all dialogs.

`archs`
> Specify the hardware architectures on which this workflow should be used.

EXAMPLE 18.3: CONFIGURING THE LIST OF WORKFLOW COMPONENTS

```
<modules  config:type="list">❶
    <module>❷
        <label>Language</label>❸
        <enabled config:type="boolean">false</enabled>❹
        <name>firstboot_language</name>❺
    </module>
<modules>
```

① The container for all components of the workflow.

② The module definition.

③ The label displayed with the module.

④ The switch to enable or disable this component in the workflow.

⑤ The module name. The module itself must be located under `/usr/share/YaST2/clients` and have the `.ycp` file suffix.

To make changes to the number or order of proposal screens during the firstboot installation, proceed as follows:

1. Open the firstboot configuration file at `/etc/YaST2/firstboot.xml`.

2. Delete or add proposal screens or change the order of the existing ones:

   - To delete an entire proposal, remove the `proposal` element including all its sub-elements from the `proposals` section and remove the respective `module` element (with sub-elements) from the workflow.

   - To add a new proposal, create a new `proposal` element and fill in all the required sub-elements. Make sure that the proposal exists as a YaST module in `/usr/share/YaST2/clients`.

   - To change the order of proposals, move the respective `module` elements containing the proposal screens around in the workflow. Note that there may be dependencies to other installation steps that require a certain order of proposals and workflow components.

3. Apply your changes and close the configuration file.

You can always change the workflow of the configuration steps when the default does not meet your needs. Enable or disable certain modules in the workflow (or add your own custom ones).

To toggle the status of a module in the firstboot workflow, proceed as follows:

1. Open the `/etc/YaST2/firstboot.xml` configuration file.

2. Change the value for the `enabled` element from `true` to `false` to disable the module or from `false` to `true` to enable it again.

```
<module>
```

```
    <label>Time and Date</label>
    <enabled config:type="boolean">true</enabled>
    <name>firstboot_timezone</name>
</module>
```

3. Apply your changes and close the configuration file.

To add a custom made module to the workflow, proceed as follows:

1. Create your own YaST module and store the module file *module_name*.ycp in /usr/ share/YaST2/clients.

2. Open the /etc/YaST2/firstboot.xml configuration file.

3. Determine at which point in the workflow your new module should be run. In doing so, make sure that possible dependencies to other steps in the workflow are taken into account and resolved.

4. Create a new module element inside the modules container and add the appropriate sub-elements:

```
<modules config:type="list">
    …
    <module>
        <label>my_module</label>
        <enabled config:type="boolean">true</enabled>
        <name>filename_my_module</name>
    </module>
</modules>
```

   a. Enter the label to be displayed on your module in the label element.

   b. Make sure that enabled is set to true to have your module included in the workflow.

   c. Enter the file name of your module in the name element. Omit the full path and the .ycp suffix.

5. Apply your settings and close the configuration file.

> **💡 Tip: Finding a Connected Network Interface For Auto-Configuration**
>
> If the target hardware may feature more than one network interface add the `network-autoconfig` package to the application image. `network-autoconfig` makes sure that during firstboot all available Ethernet interfaces are cycled until one is successfully configured with DHCP.

18.2.5 Configuring Additional Scripts

Firstboot can be configured to execute additional scripts after the firstboot workflow has been completed. To add additional scripts to the firstboot sequence, proceed as follows:

1. Open the `/etc/sysconfig/firstboot` configuration file and make sure that the path specified for `SCRIPT_DIR` is correct. The default value is `/usr/share/firstboot/scripts`.

2. Create your shell script, store it in the specified directory, and apply the appropriate file permissions.

18.2.6 Providing Translations of the Installation Workflow

Depending on the end user it could be desirable to offer translations of the customized workflow. Those translations could be necessary, if you customized the workflow by changing the `/etc/YaST2/firstboot.xml` file, as described in *Section 18.2.4, "Customizing the Workflow"*. This is different from the localization of customized YaST messages, which is already described in *Section 18.2.1, "Customizing YaST Messages"*.

If you have changed `/etc/YaST2/firstboot.xml` and introduced string changes, generate a new translation template file (`.pot` file) and use the `gettext` tool chain to translate and finally install the translated files in the YaST locale directories (`/usr/share/YaST2/locale`) as compiled `.mo` files. Proceed as follows:

1. Change the `textdomain` setting from:

   ```
   <textdomain>firstboot</textdomain>
   ```

   to, for example,

```
<textdomain>firstboot-oem</textdomain>
```

2. Use **xgettext** to extract the translatable strings to the translation template file (`.pot` file), for example to `firstboot-oem.pot`:

```
xgettext -L Glade -o firstboot-oem.pot /etc/YaST2/firstboot.xml
```

3. Start the translation process. Then package the translated files (`.LL_code.po` files) the same way as translations of the other projects and install the compiled `firstboot-oem.mo` files.

If you need translations for additional or changed YaST modules, provide translations within such a module itself. If you changed an existing module, make sure to change also its text-domain statement to avoid undesired side effects.

💡 Tip: For More Information

For more information about YaST development, refer to http://en.opensuse.org/openSUSE:YaST_development. Detailed information about YaST firstboot can be found at http://doc.opensuse.org/projects/YaST/SLES11/tdg/bk09ch01s02.html.

18.3 Cloning the Master Installation

Clone the master machine's disk using any of the imaging mechanisms available to you, and roll these images out to the target machines. For more information about imaging see http://doc.opensuse.org/projects/kiwi/doc/.

18.4 Personalizing the Installation

As soon as the cloned disk image is booted, firstboot starts and the installation proceeds exactly as laid out in *Section 18.2.4, "Customizing the Workflow"*. Only the components included in the firstboot workflow configuration are started. All other installation steps are skipped. The end user adjusts language, keyboard, network, and password settings to personalize the workstation. After this process is finished, a firstboot installed system behaves as any other instance of SUSE Linux Enterprise Server.