

LinuxCOE SystemDesigner

Table of Contents

<u>LinuxCOE SystemDesigner</u>	1
<u>Administration Guide</u>	1
<u>Hewlett-Packard</u>	1
<u>Bryan Gartner</u>	1
<u>Lee Mayes</u>	1
<u>Craig Lamparter</u>	1
<u>Preface</u>	3
<u>Installation of LinuxCOE SystemDesigner</u>	4
<u>Prerequisites</u>	4
<u>Installation Methods</u>	5
<u>Snapshots</u>	5
<u>Note</u>	5
<u>Note</u>	6
<u>Packages</u>	6
<u>Configuration</u>	7
<u>Configuration Standards</u>	7
<u>Note</u>	8
<u>Global Configuration File</u>	8
<u>Common Administration Tasks</u>	10
<u>Information Interface</u>	10
<u>Theme Customization</u>	13
<u>Note</u>	13
<u>Note</u>	14
<u>Configuration Overlay</u>	14
<u>Add EULA or notification</u>	15
<u>Assigned Choices</u>	15
<u>Advanced Administration Tasks</u>	17
<u>Add Distro, Ver, and Arch Option</u>	17
<u>Note</u>	17
<u>Add Install Method</u>	18
<u>Forced Components</u>	18
<u>Note</u>	18
<u>Note</u>	19
<u>Value-Add Modules</u>	19
<u>Note</u>	20
<u>Summary</u>	21
<u>Colophon</u>	22

LinuxCOE SystemDesigner

Administration Guide

Hewlett-Packard

Bryan Gartner

Lee Mayes

Craig Lamparter

Copyright © 2000-2009 Hewlett-Packard Development Company, L.P.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

\$Date: 2009/02/20 14:23:05 \$

Table of Contents

Preface

Installation of LinuxCOE SystemDesigner

Prerequisites

Installation Methods

Snapshots

Packages

Configuration

Configuration Standards

Global Configuration File

Common Administration Tasks

Information Interface

Theme Customization

Configuration Overlay

Add EULA or notification

Assigned Choices

Advanced Administration Tasks

Add Distro, Ver, and Arch Option

Add Install Method

Forced Components

Value-Add Modules

Summary

Colophon

Preface

One of the benefits of FOSS is the amount of choice afforded to the followers and adopters. As always though, this comes with an associated cost, and in particular during deployment and life cycle management. There are many degrees of freedom when one considers the vast amount of choice in Linux distributions, and their respective versions and supported architectures. The question then becomes can users easily deploy their selection, and further, how can such a deployment be scaled to an organizational scale. One solution to this is the **LinuxCOE SystemDesigner** (<http://linuxcoe.sf.net>)

Released to the FOSS community under the **GPL** in 2005 by Hewlett-Packard, this package provides a web interface to easily deploy a large number of the popular Linux distributions. Used within that corporation for many years, it caters to simple, one-off deployments but can just as easily store a profile of installation selections for mass deployments. The **LinuxCOE SystemDesigner** web interface allows users to design or augment the properties for a Linux system installation by exposing the various attributes from the specific Linux distribution installation process, yet gives all of them a common look and feel. In this way, users of all expertise levels are accommodated in its usage model. It ultimately utilizes the Linux distribution's native back-end technology for the actual installation, so as to more easily integrate with any IT staff processes, training, and education.

In all of its modes, **LinuxCOE SystemDesigner** can deploy systems that are completely stock from the distribution perspective. Additionally though, it can easily be extended to offer any number of value-add modules, delivering bundles of software relevant to a particular task or organization. This can include FOSS, proprietary, or localized customization offerings. It is this lack of assumptions about how one runs their infrastructure that ensures compatibility for the use of **LinuxCOE SystemDesigner** and the preservation of the value proposition of choice in FOSS.

Installation of LinuxCOE SystemDesigner

Table of Contents

[Prerequisites](#)

[Installation Methods](#)

[Snapshots](#)

[Packages](#)

The **LinuxCOE SystemDesigner** provides a unifying web interface to a large number of Linux distributions. It allows users and administrators to design Linux system characteristics and create hands free Linux system installations. The installation and configuration of the **LinuxCOE SystemDesigner** is carried out in a modular fashion.

Example use case:

A Linux system administrator would like to setup a process for his peers and user base to install Linux in a repeatable and consistent fashion, without having to continually manage the installation media. Using the **LinuxCOE SystemDesigner** and a set of network repositories, either local or public mirrors, the users can reference a specific profile, or install recipe, or can create ad-hoc configurations upon demand.

Prerequisites

The **LinuxCOE SystemDesigner** requires just a few key services for the intended system to host a valid instance.

- The hardware requirements to host the **LinuxCOE SystemDesigner** instance are quite modest. Basically a working setup will comprise any typical computer, or virtual machine instance, with the following attributes:
 - either an i386 or x86_64 system (others may work, but have not yet been tested), or even a virtual machine instance
 - capable of running Linux, due to required loopback mount capabilities
 - has reasonable quantities of RAM for the given Linux distribution's base installation,
 - has enough spare disk space, the **LinuxCOE SystemDesigner** application is less than 5MB, however, installing all of the distribution modules will still require less than 20MB of space, but you will still need to provide space for all of the seed bootable images, which varies according to the distributions you will vend, plus enough free space to manipulate and serve the generated boot images (although these are automatically cleaned up by timestamp).
- The operating system should be Linux, you should have access to this networked system, and you have privileges to install software on that host.
- A working [Perl](#) environment, as the CGI interface is completely implemented with this programming language.
- A web server (like [Apache](#)) is installed and configured to serve files, along with knowing the credentials of the designated user and group that runs web server processes.
- The [sudo](#) package is installed and usable. This will be needed to allow the web server user to run specific commands (like mount and umount) to manipulate the Linux boot images that the **LinuxCOE SystemDesigner** vends.

- The [genisoimage](#) or [mkisofs](#) package is installed and usable. This will be needed to allow the web server user the ability to create bootimage ISO files

Installation Methods

By design, the **LinuxCOE SystemDesigner** is very modular. As such, the installation will be comprised of installing a base, and then a succession of optional modules.

Snapshots

As you may want to access the latest artifacts, one supported method of installation, is to install directly from [CVS](#) or from the [project snapshots](#) which are available as *gzip'd tar files*

Since the namespace has evolved over time, you may refer to the following list to help map and download the appropriate [CVS](#) modules or [snapshots](#) or [package snapshots](#) , respectively.

- CVS modules : snapshots : package snapshots
 - SystemDesigner : linuxcoe-sd : linuxcoe-sd-base
 - docs : linuxcoe-sd-docs : linuxcoe-sd-docs
 - SystemDesigner-DISTRO : linuxcoe-sd-data-DISTRO : linuxcoe-sd-data-DISTRO
- First, download the latest files from the **LinuxCOE SystemDesigner** project. To get started, you will need the "*SystemDesigner*" module from [CVS](#) or it's corresponding "*linuxcoe-sd*" snapshot file. Complete this installation first! All of the other modules, reference decisions from this base module's installation preferences.

Note

In case you would like to generate a fresh copy of the snapshots from the project version control repository, you can perform the following actions:

- checkout (or export) a copy into a local sandbox from the [repository](#) via [anonymous CVS](#). Of course you can select any particular version, including HEAD of stream, by using the appropriate CVS option.
- Perform the following commands:

```
cd ModuleName
./autogen.sh
./configure
make dist
```

- you should now have a local copy of the module's snapshot in *gzip'd tar* format
- Next, pick a suitable location and unpack the module with

```
tar -zxvf linuxcoe-sd-@PACKAGE_VERSION@.tar.gz
```

and change directories into the newly created distribution directory.

- Next, please review the included *README* and *INSTALL* files for the steps to take to finish the installation of this module.

Note

All of the other additional modules have a dependency on the "*SystemDesigner*" CVS module or "*linuxcoe-sd*" snapshot so it should be completely installed first.

- Finally, you will probably also want to obtain the "*SystemDesigner-docs*" CVS module or "*linuxcoe-sd-docs*" snapshot as well. Additionally, you will also want to download at least one or more of the "*SystemDesigner-DISTRO*" CVS modules or the "*linuxcoe-sd-data-DISTRO*" snapshot corresponding to the particular Linux distribution you want this instance to vend. Download these, unpack them, and follow the instructions in their *README* and *INSTALL* files, which contain the specific instructions to install each respective module.

Packages

At this point in time, basic pre-packaged versions in standard ".deb" or ".rpm" formats are being automatically generated. Download the package snapshots for your distribution, version, and architecture and please provide any feedback you may have.

Configuration

Table of Contents

[Configuration Standards](#)

[Global Configuration File](#)

Configuration Standards

By design, the **LinuxCOE SystemDesigner** is extremely configurable. The look and feel of much of the web interface is customizable, much of the functionality of the interface is controlled by configuration settings, and nearly all the user-selectable information displayed during the usage of the **LinuxCOE SystemDesigner** is also configurable. From a single code base, it is possible to host instances that behave quite differently, offering either a superset or subset of choices to the users.

The **LinuxCOE SystemDesigner** application attempts to adhere to the following convention regarding the acquisition and precedence of configuration artifacts:

- If the directive "*ALT_DATA*" is defined in the [Global Configuration File](#), use configuration values from files located in that directory.
- If it exists, use configuration values from files located in

@prefix@/etc

directory.

- If nothing is available from above, use configuration values from files located in the

@prefix@/

directory.

- If still nothing available from either of above locations, use configuration values that are coded within the **LinuxCOE SystemDesigner** application itself as defaults for any items which must have some runtime value.

The **LinuxCOE SystemDesigner** application attempts to adhere to the following convention regarding the naming of configuration artifacts:

- If it exists, use configuration values from filenames containing the string "*Distribution-Version-Architecture*" which matches the particular tuple the user selected.
- If there is not a specific match from above, use configuration filenames with the string "*Distribution-Version*", basically wildcarding the Architecture portion.
- If there is still not a match from the preceding two cases, use configuration filenames with the string "*Distribution*", basically wildcarding both the Version and Architecture portions.

As such, the installation process of the **LinuxCOE SystemDesigner** places nearly all of the configuration items in the directories under

@prefix@/

Conveniently, this means that you can easily affect change by simply copying the respective configuration file from

```
@prefix@/
```

to it's corresponding location in

```
@prefix@/etc
```

and making changes in this latter location. This will preserve the updates even during subsequent installations or upgrades of the **LinuxCOE SystemDesigner**.

The following list serves to identify location of and function for the major categories of configuration files (as installed in their default locations. You are also encouraged to view the respective README file in each of these locations.

Note

These links will only be usable if viewing this document on a system that has a running instance of the **LinuxCOE SystemDesigner**.

- [@prefix@/linuxcoe.rc](#) -- global configuration file
- [@prefix@/addons](#) -- delivery of custom, value-add modules
- [@prefix@/base](#) -- specific forced PRE/MID/POST packages
- [@prefix@/boot](#) -- collection of files embedded in boot image
- [@prefix@/data](#) -- distribution specific selections
- [@prefix@/depots](#) -- sets of package repository depot specifications
- [@prefix@/html](#) -- static elements of web interface
- [@prefix@/html/bundles](#) -- HTML formatted listings of distribution bundles
- [@prefix@/html/eulas](#) -- optional EULA or notifications regarding usage
- [@prefix@/html/figures](#) -- collection of figures for documentation guides
- [@prefix@/html/guides](#) -- collection of docbook-format guides
- [@prefix@/html/images](#) -- collection of web interface images
- [@prefix@/html/themes](#) -- default web interface theme
- [@prefix@/images](#) -- initial seed boot image for particular distribution
- [@prefix@/osvend.d](#) -- collection of OSVEND file fragments

Global Configuration File

Much of the personality and functionality of the **LinuxCOE SystemDesigner** application is dictated by a single configuration file. While packaged with sensible defaults for a basic installation, the

```
@prefix@/linuxcoe.rc
```

file contains many configuration directives to control the overall application. As an example, this file is used to configure:

LinuxCOE SystemDesigner

- path information for log files, database access (if used), and downloads (and transports used)
- web interface theme location, and paths (plus transports) for interface itself
- options and default settings for web interface pull-down menus and checkboxes

All of the options and settings are commented within the file itself. You are encouraged to read through this file to get a feel for the vast range of configurations possible. And like the file itself says, "*If there's something that is not configurable that you think should be ...*" just let the development team know. Many of these configurations parameters have been suggested by other users and been implemented for all of us to use.

When needed, you may make changes to the global configuration for **LinuxCOE SystemDesigner** by adjusting the values contained in this file. The default copy of this file is located at:

```
@prefix@/linuxcoe.rc
```

and you are free to to modify this installed version. The installation tools should preserve your changes through subsequent upgrades to the **LinuxCOE SystemDesigner**. Specific configuration changes that are often asked about are highlighted in the Common Administration Tasks section.

Common Administration Tasks

Table of Contents

[Information Interface](#)

[Theme Customization](#)

[Configuration Overlay](#)

[Add EULA or notification](#)

[Assigned Choices](#)

Each of the following sections strives to be a mini-HowTo for those wishing to change a particular aspect of the default behavior and configuration of the **LinuxCOE SystemDesigner**. Most of these only involve changes in a single configuration file or location.

Information_Interface

Use Case:

With all of the configuration files and settings, it is sometimes difficult to grasp what can be provisioned from the **LinuxCOE SystemDesigner** web interface.

A very simple interface is available for the **LinuxCOE SystemDesigner** web service allows one to query the following items:

1. a list of distribution, version, architecture tuples via:

```
/@webalias@-cgi-bin/coe_info/osvend
```

```
Ubuntu Gutsy i386
Ubuntu Gutsy x86_64
Ubuntu Hardy i386
Ubuntu Hardy x86_64
Ubuntu Intrepid i386
Ubuntu Intrepid x86_64
END_OF_LINUXCOE_BEAM_DATA
```

LinuxCOE SystemDesigner

2. a list of profiles available for each tuple, via:

```
/@webalias@-cgi-bin/coe_info/list/distribution/version/architecture
```

with appropriate substitutions for the desired distribution, version, and architecture from the previous listing.

3. and a view the actual content of a given profile for its respective tuple

```
/@webalias@-cgi-bin/coe_info/beam/distribution/version/architecture/profile
```

with appropriate substitutions for the desired distribution, version, architecture, and profile name from the previous listing.

As you might have guessed, this interface is very useful to detect when certain attributes of your **LinuxCOE SystemDesigner** instance are not delivering what is expected, or even as a way to archive or pass profiles between one instance and another.

Theme Customization

Use Case:

You wish to customize the look and feel of the theme for the **LinuxCOE SystemDesigner** web interface to correspond to branding requirements or to your internal web standards.

The web interface has a very simple theme methodology. By default, the **LinuxCOE SystemDesigner** will prepend header content, then embed the coded output, and finally append footer content. For instance, the global configuration file *linuxcoe.rc* contains the following relevant directives:

```
# BRANDING
# this will replace 'LinuxCOE Version' in all of the cgi's output
# It defaults to 'LinuxCOE Version' if unset.
# BRANDING LinuxCOE @PACKAGE_VERSION@

# COETHEME - Wrap SysDes in this theme.
# will read this directory, tossing all header*.html files at the browser
# before its data, and all footer*.html files afterwards
COETHEME @prefix@/html/themes/default
```

With this default setting, the **LinuxCOE SystemDesigner** will look in the "*COETHEME*" directory for any files called "*header**" and "*footer**" distinguished by their typical lexographic sort order. So, the following ordered set of content (given the above configuration directive and the default delivered filenames) is delivered for the browser to render:

1. header1.html
2. header2.html
3. header3.html
4. header4.html
5. header5.html
6. **LinuxCOE SystemDesigner** coded output
7. footer1.html

Note

Unfortunately, all of this theme customization only applies to the web interface that is dynamically generated. For the more static home page, one would need to manually edit the include statements to reflect your new theme locations.

Feel free to use your web browser to view the page source of your newly installed **LinuxCOE SystemDesigner** instance to examine this content ordering and each file's contribution. Each of the header and footer sections are annotated to show their contribution to the final rendered page.

So, you have a couple of options to change this default theme to fit your organizational or branding preferences.

1. The quick approach is to simply modify the files (any combination of *header** and *footer**) delivered during installation in the directory

```
@prefix@/html/themes/default
```

to match your preferences of your new theme.

Note

The caveat is that a subsequent re-installation will likely obliterate **all** of your changes, reverting back to the default theme.

2. To avoid potential future conflicts with the **LinuxCOE SystemDesigner** installation process, a better approach is to change the *COETHEME* directive in the global configuration file to point to any location of your choosing which contains the desired *header** and *footer** files.

Configuration Overlay

Use Case:

You wish to change only a few configuration values from the Global Configuration File but really only want these changed configurations visible some of the time or for a particular audience of **LinuxCOE SystemDesigner** users.

While the global configuration file follows the Configuration Standards it also has some rather unique qualities as well regarding the precedence of configuration values. In essence, a configuration value can be effectively overwritten by using a slightly different invocation of the **LinuxCOE SystemDesigner** application.

Given the above use case, you can perform this "overlay" of configuration values by:

1. Determine which configuration values from the Global Configuration File that are of interest.
2. Create a file snippet of these name and value configuration settings, and save it to an appropriately named file in the directory:

```
@prefix@/etc
```

Using the standard web interface of the **LinuxCOE SystemDesigner**, invoke any of the "Boot Image", "Profiles", or "Retrofit" modes. Once the browser has rendered the page, append the following string to the navigational URL:

```
?defs=FileSnippetName
```

This will essentially source the Global Configuration File and then source your new file snippet's contents in order, utilizing the last name and value directive sourced.

As you now navigate this particular **LinuxCOE SystemDesigner** mode, you should be able to notice the changed behavior of the configuration values contained in the file snippet.

You can repeat this step as often as necessary, creating as many "overlay" configuration sets as needed.

3. And for bonus points, you could modify any calling page by deep-linking to the newly parameterized **LinuxCOE SystemDesigner** mode (with the appended)

```
?defs=FileSnippetName
```


to affect a differing view for that audience.

Add EULA or notification

Use Case:

Either due to legal requirements or for local support reasons, you wish to notify users of your **LinuxCOE SystemDesigner** web interface that a particular offering has some important ramifications before they proceed.

For each of the "*Distro, Ver, and Arch*" tuple offering, you can place a "*click-through*" acceptance step in the *Design a system and create a network installation boot image* process. This is quite simple to implement.

- Navigate to the

```
@prefix@/html/eulas
```

directory. Simply put an HTML file snippet in that location with any suitable markup to describe the situation. Also ensure the file contains (usually at the end), the explicit text

```
AGREE BUTTON
```

with any optional text to follow the actual continuation button. Save the file as

```
Distro-Version-Architecture.html
```

substituting the respective values for your desired target tuple.

- Then test this by selecting the same tuple and your "*Boot Image*" process will now have a Step 1b to click-through.

Assigned Choices

Use Case:

To simplify the **LinuxCOE SystemDesigner** web interface for first-time or novice users, you wish to make one or more of the available values show up as defaults or in fact eliminate choices.

Given the above use case, you can set many of the choices to have a default, or preselected, value. If you want them set in the primary web interface of the **LinuxCOE SystemDesigner**, you would directly modify the Global Configuration File and simply search for the name and value pairs that are referenced by the "*DEFAULT*" keywords (likely in the associated comments). Then just modify the value to be your default answer. Users may still browse the available selections, and even pick others, but at least you have suggested an appropriate, default value.

While you are modifying the Global Configuration File you may also notice a set of name and value pairs with the "*ONLY*" keyword. If you want to force users toward a prescribed choice, you may modify these values. If done in the Global Configuration File this will result in a restricted, or simplified user experience.

LinuxCOE SystemDesigner

Of course, either the "*DEFAULT*" or the Configuration Overlay methodology to create any number of combinations, where parameters are either limited, or have preselected, default values. It really is up to you what you expose for the web interface.

Advanced Administration Tasks

Table of Contents

[Add Distro, Ver, and Arch Option](#)

[Add Install Method](#)

[Forced Components](#)

[Value-Add Modules](#)

Each of the following sections strives to be a mini-HowTo to change a particular aspect of the default behavior and configuration of the **LinuxCOE SystemDesigner** for more complex changes that may require many changes in multiple locations.

Add Distro, Ver, and Arch Option

Use Case:

You wish to add a specific selection to the **LinuxCOE SystemDesigner** web interface for a favorite tuple of distribution, version, and architecture.

Each of the **LinuxCOE SystemDesigner** web interface nodes keys the remaining steps upon the "*Distro, Ver, and Arch*" tuple. However, when you visit your instance, the combination of interest is missing. Adding this may range from being very easy to moderately difficult or even not possible at this point in time.

- The easy answer is to ensure you have the appropriate distribution overlay module installed on your instance. You can do this by referencing the [Installation of LinuxCOE SystemDesigner](#) portion of this document. If your desired selection still is not visible, and it is in the set of available distribution overlay modules, then you may need to edit the relevant file in the following directories:

```
@prefix@/etc/osvend.d
@prefix@/osvend.d
```

Perhaps, an entry is commented out (even as installed). This is often the case for those distributions which have licensing or subscription issues, and the entry cannot be completed without local, site-specific information. Please modify the respective line and retry your **LinuxCOE SystemDesigner** session.

- For certain combinations of the distribution, version, and architecture tuple, it may be possible to extend an existing distribution overlay module to support your preference. For instance, when an update level or service pack version of a distribution is release, it may be as simple as adding the required OSVEND entries as above, and then using symlinks throughout the various configuration directories to link the earlier, and now current version.

Note

It is currently beyond the scope of this document to detail all of the changes necessary (but may become a future section of this document if interest exists). So, the best bet is to contact the developers as noted on the [project home page](#).

Likewise for new distributions, it would probably be easiest to ask the developers for help, guidance, and

suggestions, especially if the new distribution utilizes a new (to the **LinuxCOE SystemDesigner**) technology for automated installations.

Add Install Method

Use Case:

For whatever reason, users cannot utilize the install method protocol you currently have configured in your **LinuxCOE SystemDesigner** web interface and you need to add a different protocol.

By default, the **LinuxCOE SystemDesigner** is currently capable of supporting network options of FTP and HTTP plus local media options of Floppy, USB drive, or CD/DVD as install methods.

- If one of the two network protocol methods (FTP/HTTP) is not configured on your instance, you may add the desired one by modifying the relevant OSVEND entries in It is quite possible that the particular entry is commented out or missing. Generate a suitable line, save the resultant file, and retry your **LinuxCOE SystemDesigner** session.
- If one of the local media options is missing, this may be intentional. Newer distributions (with kernels of revision 2.6 and higher) will not fit on a floppy drive. Other distributions may not support USB drive options. For publically visible distributions, little or no effort has been made to enable local media installations, since multitudes of public mirrors are usually available.

Forced Components

Use Case:

In your environment, you would like some set of packages to be installed on every system. This may be for monitoring reasons or to implement some security checks, perhaps as part of your acceptable usage policy.

Given the above use case, you can configure the **LinuxCOE SystemDesigner** to deliver these packages, even if they are above and beyond the distribution. These are done in a forced manner, in that users do not have anything in the web interface to de-select them.

Note

Due to this forced nature, it would be considered good form to notify the users of these forced selections, and the reasoning behind them via documentation or other communications means.

To implement these forced components, you should review the contents of the

```
@prefix@/base/README
```

file. This will give you an overview of how to setup forced components. As noted in the [Configuration Standards](#), you will likely want to make a corresponding directory in

```
@prefix@/etc/base
```

and make your modifications there. This will preserve your changes during upgrades and re-installation of the **LinuxCOE SystemDesigner**.

In this new directory space, you can populate the respective "*PRE*"-*, "*MID*"-*, and "*POST*"-* files as needed to install the packages you wish to force.

Note

Since these are forced components, you will want to completely test the packages you wish to install, and that the "*PRE*", "*MID*", and "*POST*" dependencies and ordering is acceptable to you and to the native Linux package manager. And remember, users may select only very base or minimal installations.

During the actual installation on the target system, these forced components will be installed after the main distribution (as prescribed with the user selected software packages). This will be done in phases corresponding to the "*PRE*", "*MID*", and "*POST*" ordering.

Value-Add Modules

Use Case:

For users of your environment, you would like some set of optional packages to be available for installation (and even later retrofit). These may be to layer on additional functionality or to accommodate user preferences.

Given the above use case, you can configure the **LinuxCOE SystemDesigner** to present an interface to these packages, delivering these packages during installation and even for later delivery via the retrofit function. Depending on how you setup the Value-Add module, updates to these packages may also be visible to target systems as they are made available. Value-Add modules are completely optional, and provided as a means to further the value proposition of choice.

To help set the stage, let's start off with some clarifying terminology:

- The term "*Add-on*" is a configuration reference to a "Value-Add Module", enabling the **LinuxCOE SystemDesigner** to offer this optional functionality.
- A "*Value-Add Module*" references a collection of one or more "Software Bundles" grouped together, usually to address a specific audience. It consists of a pair of configuration files, and references an associated repository containing the "Software Bundles".
- A "*Software Bundle*" is one or more software packages bundles defined together, usually to address a specific function. These are often housed in a directory structure, along with some other files, to become a package repository.

Now that we have those terms in place, we can go about configuring the **LinuxCOE SystemDesigner** to provide this functionality.

To implement these value-add modules, several steps are required:

1. To enable the **LinuxCOE SystemDesigner** to include an "*Add-on*", you will need to modify the Global Configuration File directly or via the Configuration Overlay process. Look for the following directive:

```
# ADDONS - Additional Software Depots
```

LinuxCOE SystemDesigner

```
# ex: ADDONS LinuxCOE HPONLY Delivery
#ADDONS LinuxCOE
```

and modify to have the ADDONS line reference a filename of your choosing that reflects this particular "Add-on".

2. Next, create a filename (matching the ADDONS directive) in any of the locations referenced by the Configuration standards. You may want to copy the content of the supplied example

```
@prefix@/addons/LinuxCOE
```

as a starting point. Your new file will need significant modifications to match your new environment and to correspond to your "*Software Bundle*" repository. The comments of the example file provide a guideline as to what is expected.

Of particular interest is the directive:

```
# CONFIG -> Absolute filesystem path to the bundle configuration file
CONFIG /SomeAbsolutePath/some_bundles.cfg
```

This file referenced by the "CONFIG" directive, must also reside on your local **LinuxCOE SystemDesigner** instance. You may want to copy the content of the supplied example

```
@prefix@/addons/LinuxCOE.cfg
```

as a starting point. The comments of the example file provide a guideline as to what is expected.

Note

Due to modern package managers, you typically don't need to specify any dependency packages, except those which may reside outside the base repository for the associated Distribution-Version-Architecture. However, you should thoroughly test the installation, since users may select only very base or minimal installations.

3. Next, you'll need use some force components that the **LinuxCOE SystemDesigner** team has created to manage the target system's apt or yum repository configurations. These are currently available in the postproc CVS module, and, for now, can be built using the EPM tool.
4. Finally, you would need to create the actual repository that contains the packages referenced, any associated help files, and likely any meta-package indices used by apt, yum, or the native package manager. This repository may be co-located on this **LinuxCOE SystemDesigner** instance, or on a remote machine assuming you have made the appropriate modifications to the directives in the *CONFIG* file with regard to "APT-RPM" "APT-DEB" "YUM-RPM" designations.

Summary

As an administrator, you should now have a feeling for the vast range of choices that can be offered by the **LinuxCOE SystemDesigner**. With the single code base installed, you can offer users many differing looks, feels, and options. These can range from a subset of functionality, to a preview of upcoming offerings. Like any user, you can provide stock profiles that are easily selectable. In addition, you can offer users bare bone Linux distributions installations, installs with certain components automatically included, or even some optional value-add bundles to be installed. Whether you want to limit your scope to a captive audience or to offer the service to the Internet at large, your typical use cases should be satisfied.

Colophon

The authors collectively are, or have been, employed at Hewlett-Packard company for nearly 60 years of total experience. Each of them has been actively using Linux for the better part of ten years, and helped foster significant Linux adoption within Hewlett-Packard due in large part to their collective contribution to **LinuxCOE SystemDesigner**. They function and excel as part of a thriving FOSS group nurtured within Hewlett-Packard, and actively promote and leverage contributions from the overall community at large.

The look of the documentation is entirely credited to the foresight of the authors of DocBook and the surrounding ecosystem of tools from that community.

Ever since inception, the **LinuxCOE SystemDesigner**. has used penguin species as it's mascot, starting with the genus name of Spheniscus, and working it's way through the list in alphabetical order for each major release. Currently the mascot is the *Emperor Penguin* which we encourage you to look up and learn about in [Wikipedia](#), or some other on-line resource.