



# OpenSolaris Text-Based Installer

Final Report

The background, design, creation, and ending position of a six week long project at Sun Microsystems.

Danny Pearson, Mark Godwin, Ryan Holden, Thea Gab  
6/15/2009

# TABLE OF CONTENTS

1.0 ABSTRACT .....	1
2.0 INTRODUCTION.....	1
3.0 REQUIREMENTS .....	1
3.1 FUNCTIONAL REQUIREMENTS.....	1
3.2 NON-FUNCTIONAL REQUIREMENTS .....	2
4.0 DESIGN.....	3
4.1 USE CASES.....	3
4.2 DESIGN OVERVIEW.....	4
4.2.1 Process flow of the Installer .....	4
4.2.2 NCURSES User Interface .....	5
4.2.3 Installation/NCURSES Connection .....	6
4.3 INTERDEPENDENCIES.....	6
4.4 SUPPORTED CONFIGURATIONS.....	6
5.0 IMPLEMENTATION DETAILS AND RESULTS.....	6
5.1 DESIGN DECISIONS .....	6
5.2 RESULTS .....	7
6.0 SCOPE .....	8
7.0 FUTURE WORK.....	8
GLOSSARY .....	9
REFERENCES/OTHER DOCUMENTATION .....	10

## 1.0 ABSTRACT

Sun Microsystems only offers the ability to install their operating system, OpenSolaris, from a GUI based interface and non-interactive automated install. For users that need to install OpenSolaris from a terminal or on computers that do not contain graphics cards compatible with the operating system, an interactive text-based install interface needs to be created that could offer a user a similar experience to the GUI installer.

The text-based installer created during the six week field session uses the NCURSES library to display colors, menus, and screens on the terminal. A library from Sun, called 'liborchestrator', is used to install the operating system via simple function calls. The project is programmed in C and designed modularly so additions can be added easily. To enable the text-based installer to begin on startup, a service is used to enable it to become part of the boot process, which is managed by the Service Management Facility (SMF) in OpenSolaris.

Distribution of the text-based installer will be offered as a boot option on the current 'LiveCD' installation OpenSolaris. Documentation of the installer along with the source code can be found in the repositories where the other installers' source code resides.

## 2.0 INTRODUCTION

OpenSolaris, Sun Microsystem's open source operating system, currently offers only two installation options, a GUI install and automated install (AI). The GUI installer provides an interactive install wizard for the user presented via the display manager GNOME. The automated installer allows a preconfigured installation to be installed across a network. Sun has long planned to provide a purely text-based installer to users based on common usage scenarios, and recently it has become a more urgent priority. Examples of such scenarios include: users that need to remotely install OpenSolaris through a console, and users without supported graphics cards. The goal for the text based installer is to mirror the functionality and user experience of the GUI installer, presenting installation options in a step-by-step fashion.

## 3.0 REQUIREMENTS

### 3.1 Functional Requirements

The main requirement for this project is to provide a text-only installer for OpenSolaris. The installer needs to run on the x86 architecture and will be provided via either CD or USB installation media. The installer should provide the same options and settings as the GUI installer and performance should be equivalent to that of the GUI installer.

## 3.2 Non-Functional Requirements

There are many additional requirements for the installer not directly related to the actual implementation, but necessary for the final project. They are:

- An excellent user experience. The user experience was designed by a Human Computer Interaction (HCI) expert on the team. The installer's user experience needs to closely resemble the design given.
- Any third party tools used must not present licensing issues. Therefore any libraries or tools used need to either already be included within OpenSolaris, or offer a suitable licensing scheme, for example: GNU Public License (GPL) or Limited GNU Public License (LGPL).
- The end product needs to be thoroughly tested by Sun. Should time remain, any bugs found will be addressed and fixed.
- Documentation needs to be created for any code developed for the project. If the project is not completely finished at the end of the six week field session, the OpenSolaris install team should be able to take over the project and finish it for release later on.
- On completion of the installer, the user needs to have the option to reboot or exit to the terminal. This will permit the user to view the installation log or troubleshoot any issues that the installer may have. Further, the user needs to be able to restart the installer manually from the command line.
- Existing installation libraries need to be used to install the operating system on the computer. An internal library at Sun called 'liborchestrator' is used for the current GUI installer and automated installer. Figure 1 highlights the interaction between the current installers and underlying libraries and code. It can be observed that the two rely mainly on only the liborchestrator library and no others.

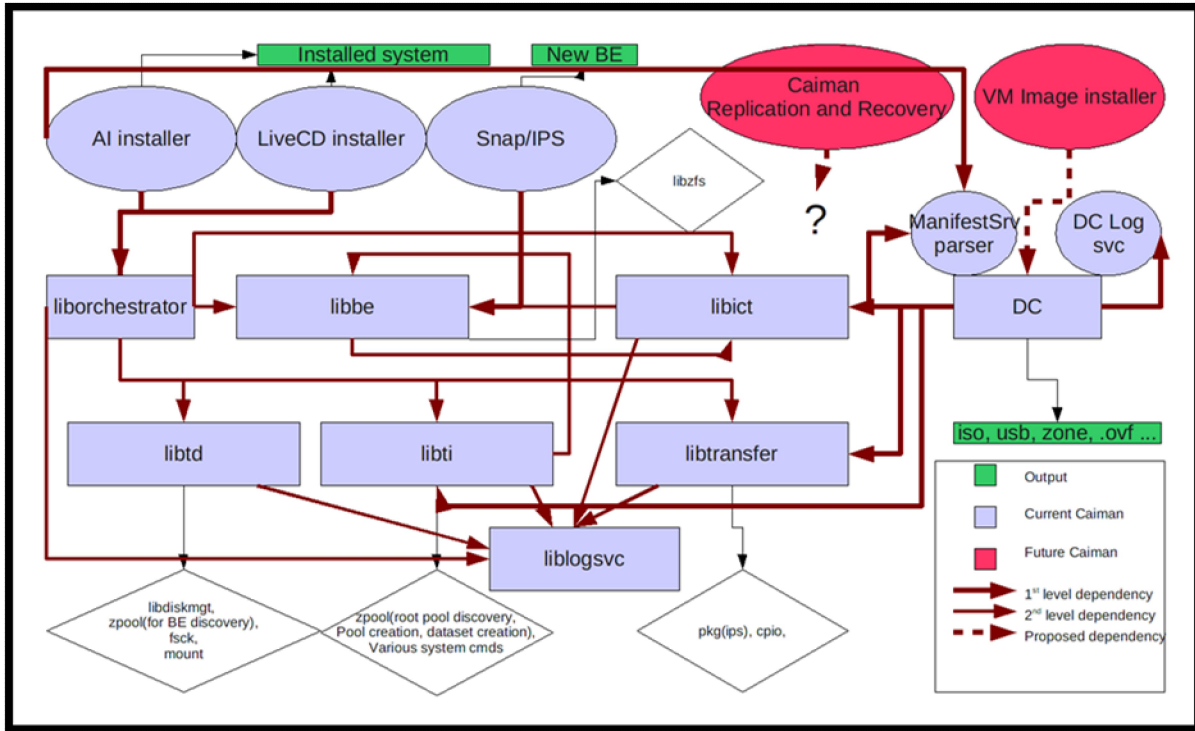


Figure 1: Current Installation Dependencies

## 4.0 DESIGN

### 4.1 Use Cases

#### Scenario One:

A user does not have enough memory or a supported graphics card for running OpenSolaris's display manager, GNOME. The user selects the 'Text-Based Installer' option from the boot menu that is presented to the user at startup from the OpenSolaris LiveCD or LiveUSB. The text-based installer installs OpenSolaris on the user's system without the user confronting the above two issues.

#### Scenario Two:

A user installing OpenSolaris has only a remote text console connection to a machine located elsewhere. The user previously inserted the CD or USB into the computer onto which they wish to install. The user selects the 'Text Based Installer' option from the boot menu that is shown to them at startup of the remote machine. They proceed to configure options and install the operating system.

#### Scenario Three:

A user installing OpenSolaris through the text-based interface encounters an error during

install. The user is given a log of the install and any appropriate error messages for troubleshooting. Further, the user is given the option to return to the terminal for command line troubleshooting.

## 4.2 Design Overview

The design of the text-based installer can be broken into three distinct parts:

1. The process flow of the installer from on the startup of the system when the text-based installer option is picked from the boot menu to what happens when the installer exits.
2. Implementation of the user interface seen throughout installation using the ncurses library in C.
3. Connecting the code written for the user interface with the install code already programmed for the OpenSolaris GUI installer using a predefined library of C functions called 'liborchestrator'.

### 4.2.1 Process flow of the Installer

From the time the computer is turned on to when OpenSolaris is actually booted, a series of events occur which enable the operating system to be loaded successfully. For the text-based installer, the series of events follows a somewhat different path than a default startup. A typical text-based install will follow the series of events shown in figure 2 and outlined below.

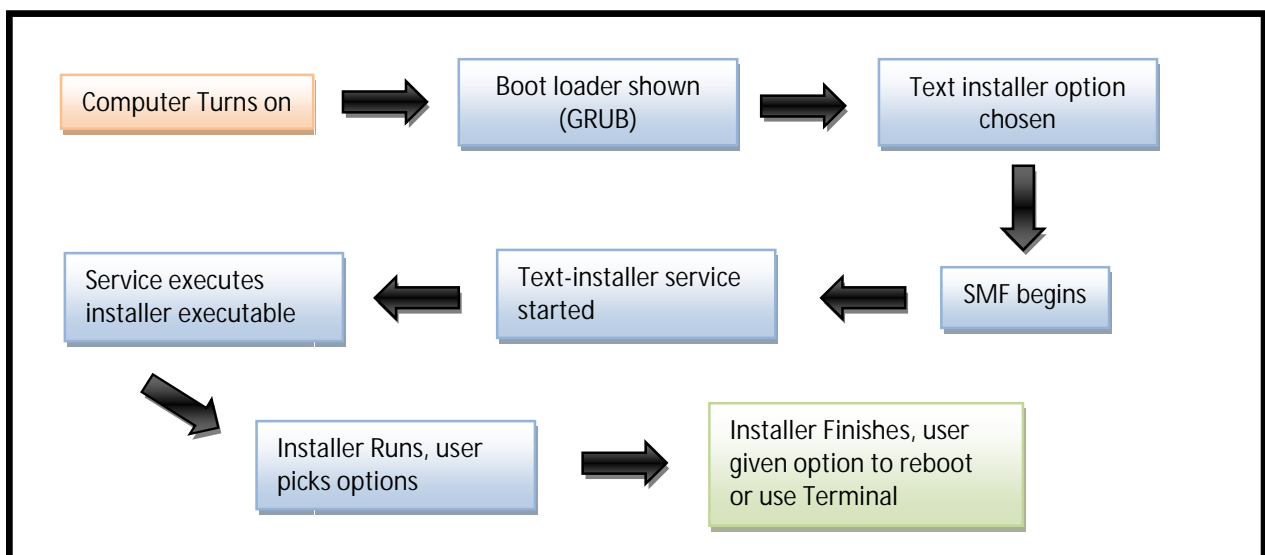


Figure 2: Text installer process flow from boot up to termination

When the computer turns on, the system searches for a boot loader on media specified in the system BIOS (hard drive, flash drive, CD, etc.). For OpenSolaris, the boot loader used is called GRUB, and when found, a list of potential boot options are shown, if they exist. Once a user selects the 'Text Installer' boot option in GRUB, the system starts loading services via SMF. The text installer service starts when all necessary services needed for an install are started. The user then proceeds through the install and has the option to reboot or exit to a text based console at the end of the execution of the program.

#### 4.2.2 NCURSES User Interface

The text installer NCURSES user interface is designed to follow the mockups created by the Human Computer Interaction (HCI) team at Sun. To ensure that each screen can be tailored uniquely depending on what needs to be displayed and to minimize duplication of code, any code dealing with NCURSES was abstracted away from code relating to the main flow of the program. Figure 3 shows the relationships between the files within the text installer.

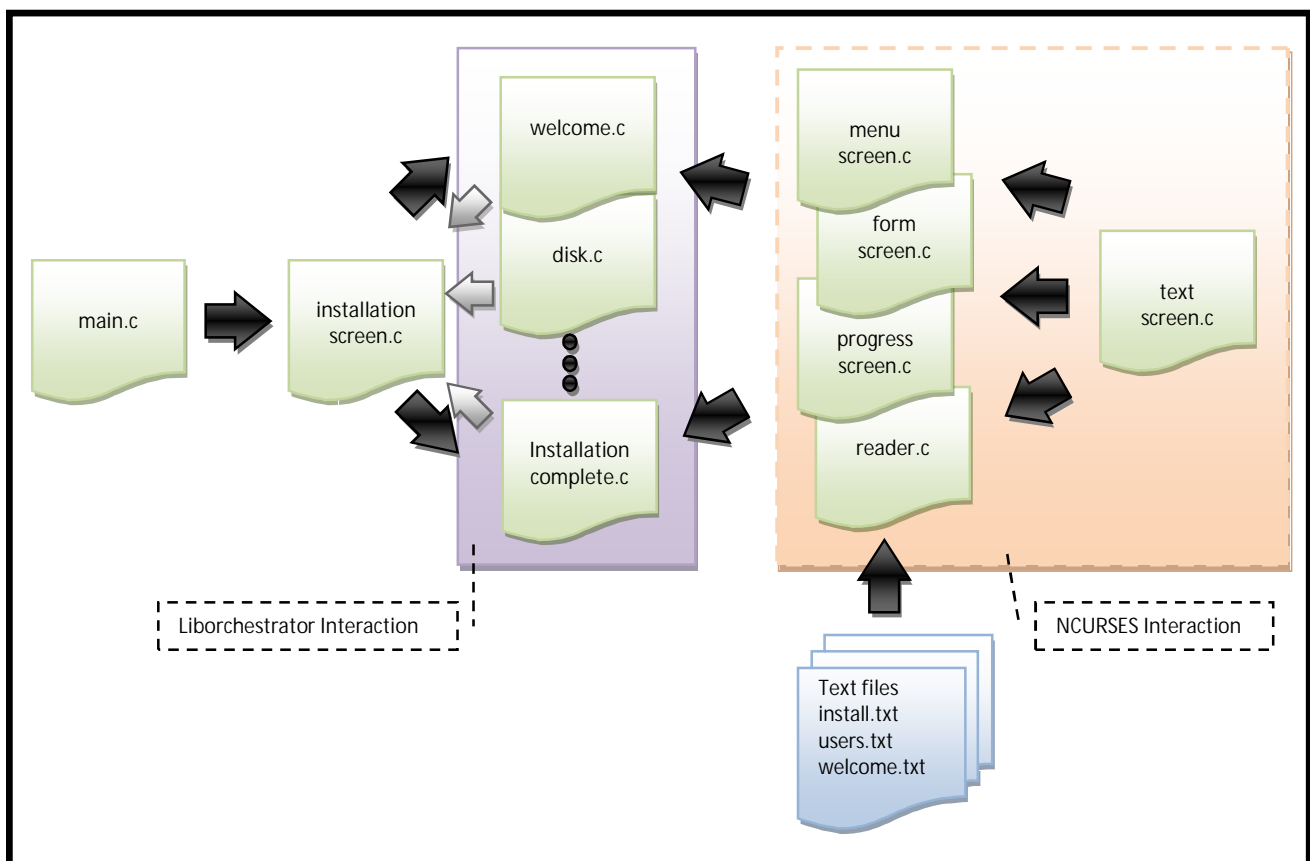


Figure 3: Text installer code design

The program enters at the left, in main.c, proceeds to installation\_screen.c, where the main flow of the program is defined – welcome screen goes to keyboard screen, keyboard screen

goes to disk screen, etc. Each of the `welcome.c`, `keyboard.c`, `disk.c`, and other screen files simply call initialization functions from the `menu_screen.c`, `form_screen.c`, `progress_screen.c`, and `reader.c` files to show the interface for that page depending on what needs to be shown on that page. Finally, each of the `menu`, `form`, and `progress` files are all built on top of the `text_screen.c` file which simply places text on the screen in a window.

#### 4.2.3 Installation/NCURSES Connection

As shown in figure 3, the `welcome`, `disk`, `installation complete`, and other screen files contain function calls to the library `liborchestrator`. This library already existed for LiveCD and has the necessary functions needed to gather environment information and accept feedback from the user to start the installation process. The new `ncurses` code interacts with these functions in a very similar manner as the GUI installer.

#### 4.3 Interdependencies

The text-based installer project relies on the current LiveCD and slim install code, because the installer is offered as an option on the LiveCD. Also, the installation code itself has not been changed. The text-based installer gathers information from the system and user about the set-up of the installation and then passes the necessary information on to the existing installation code.

Also, the code relies on the `curses`, `menu`, and `liborchestrator` libraries to compile. The user must have a terminal capable of displaying colors, or else the `ncurses` code will not run.

#### 4.4 Supported Configurations

The text-based installer is designed to run on all x86 machines currently supported by the LiveCD installer. It is not necessary that these systems have compatible graphics cards, only that a terminal is available from which to perform the install.

## 5.0 IMPLEMENTATION DETAILS AND RESULTS

### 5.1 Design Decisions



The beginning of the design process included a lot of research and conversation with Sun team members. One very important decision made was to use a purely ncurses interface. Many options were considered such as using a utility called dialog, which provides already implemented ncurses screens, and implementing ncurses in python, which also provides many utilities and functions to ease the design of the user interface. However, other considerations were also important to this decision. The user experience and the process of integrating the text-based installer code with the library calls in C that were needed to actually perform an install were very important factors. Using a purely C code implementation of ncurses would make it a lot easier to manipulate the different screens being presented to the user, rather than using predefined forms and functions. It also allowed for a less complex interface with the liborchestrator libraries. These are libraries that Sun already uses in their GUI installer, and they were not only important to us for performing an install, but also for gathering information about the system to present to the user. These factors, in addition to the recommendations of senior engineers at Sun, led to the conclusion that using the C implementation of ncurses was the best choice.

Another important decision made, was to provide the text-based installer as an option on the LiveCD. The LiveCD provides a GUI-based installer, but Sun also has an Automated Installer that was considered as a starting point for our work. Since the goal was to provide a similar experience to the user as the GUI installer and because of the ability to reuse a lot of existing libraries and functions, the LiveCD approach seemed to be a good decision given the 6-week time frame we had.

## 5.2 Results

The final product is able to install OpenSolaris with any options specified by the user through a text-based interface. The end product closely resembles and functions like the GUI installer, as can be seen below in figure 4. Current limitations of the product are mostly limited to bugs found in the code which were not able to be fixed within the timeframe of the product. These bugs have been logged and are included with the source code for future development.



Figure 4: The completed text installer (Users screen)

## 6.0 SCOPE

The primary objective of this project was to provide a text-based installer containing the same functionality as the GUI installer currently available on the LiveCD on the x86 architecture. This goal was successfully met, as OpenSolaris can be installed by using the tool created. Initially, the plan was to develop the text-based installer for both x86 and SPARC architectures. This added goal was dropped, however, due to time constraints with finishing the x86 architecture in time.

## 7.0 FUTURE WORK

While the text-based installer meets the basic requirements outlined in this paper, there is much room for further development. Both in-depth testing of the product and a code review need to be performed. After these tasks are completed, there is room for expansion on the existing design. Providing a purely line-by-line interface for those systems which will not support the ncurses interface, implementing the installer to run on SPARC architecture, and the ability to do a network install are all possibilities that could extend the capabilities of the text-based installer.

## GLOSSARY

AI: Automated installer

GNOME: A display manager used to display a graphical user interface for a variety of open source operating systems.

GUI installer: an installer for OpenSolaris that provides a graphical user interface (a window with buttons and forms) for the user to select installation options

HCI expert: Someone who specializes in human-computer interaction (i.e. the user's experience)

Install wizard: A program that enables the user to install something in a step-by-step fashion, often by clicking a next button to go from one screen to another.

LiveCD: A CD that can boot an operating system without installing it to the hard drive. Also provides the option to install the operating system to a hard drive.

NCURSES: A text-based user interface that provides "graphics" using ASCII characters

## REFERENCES/OTHER DOCUMENTATION

Slim Install Project:

[http://opensolaris.org/os/project/caiman/Slim\\_Install/](http://opensolaris.org/os/project/caiman/Slim_Install/)

Distribution Constructor:

[http://opensolaris.org/os/project/caiman/Constructor/design\\_notes/](http://opensolaris.org/os/project/caiman/Constructor/design_notes/)