Building a JumpStart™ Infrastructure

By Alex Noordergraaf - Enterprise Engineering

Sun BluePrints™ OnLine - April 2001

http://www.sun.com/blueprints

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 816-0428-10
Revision 01, April 2001

Building a JumpStart™ Infrastructure

This article describes the required components of a JumpStart™ architecture and how they interact. The goal of this article is to enable system administrators to get a basic automated JumpStart environment up and running as quickly as possible. To accomplish this goal, this article describes the following overviews and components of a basic automated JumpStart environment:

- Boot process
- Server components
- Server directory architecture
- Server installation
- JumpStart automation
- Client installation

This article is an excerpt from the forthcoming Sun BluePrints™ book titled "*JumpStart™ Technology: Effective use in the Solaris™ Operating Environment*" (ISBN# 0-13-062154-4) by John S. Howard and Alex Noordergraaf. This book, which discusses each section of this article in greater detail, is scheduled for publication by Prentice Hall in the Summer of 2001 and will be available through `http://www.sun.com/books`, amazon.com, fatbrain.com, and Barnes & Noble bookstores.

Server Components

Three specific types of services are required for a successful JumpStart installation. These services, which can be provided from one or multiple servers, are boot, profile, and install. This section discusses each service separately, but they are provided from the same server in this example.

JumpStart Boot Server

The JumpStart boot server provides the services most critical to a successful JumpStart installation and supplies the following information:

- IP address of the client
- IP address(es) of both the JumpStart profile and install servers

A JumpStart boot server does not have to be a separate server from the profile and install servers. However, it may have to be a separate server when using RARP to provide the IP address to a JumpStart client. The RARP protocol is not a routed protocol, so RARP requests are not forwarded by routers between subnets.

When using DHCP, it is not necessary to have a JumpStart boot server on each network segment. However, DHCP requires a relay server on each segment to forward DHCP requests to the DHCP server.

In WAN installations, it is necessary to have at least a boot or relay server on each subnet. A more practical resolution to this problem, instead of having separate boot clients on each network segment, is to have either one boot or install server connected to multiple networks, or to actually have a dedicated JumpStart network that a system is connected to *only* during JumpStart installations. It is also possible to have a single centralized boot server on the network and use BOOT/DHCP relay agents on routers to forward packets appropriately.

The forthcoming book, "*JumpStart™ Technology: Effective use in the Solaris™ Operating Environment*," presents an example of JumpStart installations with a boot server on a separate subnet than the profile and install servers. In addition, the book presents an example of using DHCP to avoid the necessity of a JumpStart boot server.

JumpStart Profile Server

One of the major benefits of JumpStart technology is the ability to automate system installations so the installation occurs without any human intervention. This is particularly important in large environments where new systems are always arriving and the ratio of systems to administrators is large, or in lights-out datacenter environments where there are no human operators and all operations are automated.

This configuration information can be provided through several different mechanisms. The minimum amount of information required, to automate a JumpStart installation, is the following:

- System locale
- Time zone
- Netmask
- IPv6
- Terminal type
- Security policy

- Name service
- Timeserver

If any of this information is not provided through either the configuration files on the profile server (for example, a `sysidcfg` file), or through a name service, the JumpStart installation defaults to an interactive installation.

With the release of the Solaris 8 Operating Environment (Solaris OE), not all of the required configuration information can be specified through a name service. Specifically, the Internet Protocol version 6 (IPv6) and security policy information (for example, whether they should be enabled or not) cannot be provided through NIS or NIS+ maps. The Solaris 8 OE does not support the use of LDAP as a repository for JumpStart profile information. A `sysidcfg` file must be provided to the JumpStart client, with at least the IPv6 and security policy information, to fully automate the Solaris 8 OE JumpStart installations.

A JumpStart profile server provides the configuration information needed by a JumpStart client so that the JumpStart installation can continue without any requests for additional information. With the `add_install_client` command, specify the profile server to be used and a fully qualified path to the `sysidcfg` file. This command runs on the JumpStart boot server.

JumpStart Install Server

Another major piece of information provided to the JumpStart client by the JumpStart boot server is the `rules` file entry for the JumpStart client. The `rules` file specifies what script is run before installation, the profile used, and the script that is run after the Solaris OE installation. These parameters must be present in the `rules` file but can be placeholders (for example, a hyphen) and not specify actual scripts.

For maximal flexibility, the `begin` and `finish` scripts specified in the `rules` file are typically `driver` scripts. A `driver` script is used to call other scripts. This provides a mechanism for a more modular JumpStart environment because it avoids the need for one monolithic script and instead allows a number of modular scripts.

The scripts and profile specified in the `rules` file of the JumpStart boot server actually run from the JumpStart install server to the directory specified through the `add_install_client` command. This directory must contain the Solaris OE source files and all scripts referenced in the `rules` file. In the recommended JumpStart directory architecture, (presented in the following section) all the JumpStart related information is kept in specific directories to simplify management, version control, and backups.

Server Directory Architecture

The three different types of JumpStart servers discussed in the previous section (boot, profile, and install) can reside on the same or separate systems. However, regardless of the use of one or more systems, the recommended directory architecture is the same on all the systems.

The recommended directory architecture for the JumpStart server is a set of directory structures to logically organize the required JumpStart server information in one place. This architecture allows system administrators to easily manage and maintain JumpStart server information. Ideally, all of the JumpStart information is stored in dedicated filesystem partitions. The root directory of these partitions is called `/jumpstart` throughout this article. It is in this base directory of the JumpStart server that the `rules.ok` file is stored. For this reason, the `/jumpstart` directory is used as the base directory for the install server option in the `add_install_client` command.

Within the `/jumpstart` directory, create the following directories:

- `Begin`
- `Drivers`
- `Finish`
- `OS`
- `Packages`
- `Patches`
- `Profiles`
- `Sysidcfg`

This section briefly describes these directories.

`Begin` Directory

The `Begin` directory contains the `begin` scripts that perform system modifications and updates during installation. Begin scripts can be written to perform any tasks that must be performed *before* the OE is installed on the JumpStart client to backup data or any other scripts.

Drivers Directory

The Drivers directory contains all driver scripts. Driver scripts are the scripts listed in the rules files that call all other scripts during a JumpStart execution. The driver scripts determine what modifications are made to each system by calling specified finish scripts. The finish scripts perform the actual modifications to the Solaris OE on the JumpStart clients.

Finish Directory

The Finish directory contains the finish scripts that perform system modifications and updates during installation. finish scripts can be written to perform various tasks such as patch and software installation.

OS Directory

The OS directory must contain *only* the Solaris OE files. These files are used by the JumpStart server (over the network) to build the client. Different Solaris OE releases should be stored in subdirectories within this subdirectory. The recommended naming convention for these subdirectories within /jumpstart/OS is Solaris_<version #>_<4 digit-year>_<2 digit month>. The installation process presented in this article is based on a Solaris 8 OE 1/0 (update 3) CD; therefore, the directory is named Solaris_8_2001-01. By creating different directories to store separate updates and releases of the Solaris OE, fine-grained control can be maintained for testing and deployment purposes.

Packages Directory

The Packages directory contains third-party software packages that are installed by the finish scripts. For example, third-party provided encryption software, such as a OpenSSH package, is stored in the Packages directory so the appropriate finish script can install and configure the software package.

Patches Directory

The Patches directory contains the recommended and security patch clusters (and individual patches). Subdirectories are created in the Patches directory for each of the Solaris OE versions being used. The patch clusters are extracted into the individual subdirectories. This enables the patch installation script to run without having to first extract the patch cluster for each system installation.

Profiles Directory

The Profiles directory contains all profiles. A profile is a file that contains configuration information used by the JumpStart software to determine which Solaris cluster to install (Core, End User, Developer, or Entire Distribution), the disk layout to use, and the type of installation to perform (for example, standalone). Use these configuration files to define how specific systems or groups of systems are built.

Sysidcfg Directory

The Sysidcfg directory contains directories with OE and host specific sysidcfg files. Due to the OE specific nature of the sysidcfg file, a generic version can no longer be used for all Solaris OE releases. The subdirectories use a naming convention similar to that recommended for the /jumpstart/OS directory (see "OS Directory" section). The recommended installation convention is Solaris_<version #>. The sysidcfg files for the Solaris OE version 2.6 are stored in a subdirectory named Solaris_2.6.

JumpStart Server Installation

This section provides step-by-step instructions on how to install and configure a JumpStart server and client running the Solaris 8 OE. Each step in configuring the server and client shows the commands and associated output. Explanations of the JumpStart configuration files and options are also provided. However, this section only details the fundamental JumpStart commands. To further simplify the examples presented in this section, no name services are used. All JumpStart information is provided to the JumpStart client through the use of configuration files. To further simplify the process, DHCP is not used to provide the JumpStart client IP address information.

The scenario detailed in this section consists of two systems:

- js-server01
- js-client01

Both of these systems are connected to the same isolated network segment.

The js-server01 system functions as the JumpStart boot server for js-client01. In addition, the js-server01 system provides all the information for a successful JumpStart installation. This includes the initial RARP, TFTP, sysidcfg file, begin scripts, Solaris OE packages, finish scripts, and any other information or scripts called by the begin or finish scripts.

JumpStart Server Software Installation Steps

The first step in building a JumpStart server is to install the latest Solaris 8 OE. The examples in this article are based on Solaris OE version 8 1/01 or Update 3. No additional JumpStart specific features were added with this release. It is always recommended that the most-recent Solaris OE release be used. Similarly, once the OE is installed, the latest patch cluster available from SunSolve Online^{SM} should be applied to the Solaris OE image. It is assumed, for the sake of this example, that all required network configuration, account management, and any other requirements to make the system functional have been performed.

Once these tasks are performed, define a JumpStart partition with adequate space on the system. Given the size of the Solaris OE image set aside on js-server01, a partition of several gigabytes should be allocated.

The JumpStart server built in the following example will provide all three of the JumpStart services required (i.e., install, boot, and config) from one system.

▼ Solaris OE Image Installation

This sequence of steps will copy the Solaris 8 OE 1/01 image onto the JumpStart servers /jumpstart/OS/Solaris_8_2001-01 directory.

1. To create a Solaris 8 OE JumpStart server, insert the first Solaris 8 OE software CD (labeled 1 of 2) into the CD-ROM drive and enter the following commands:

```
 # pwd
/cdrom/sol_8_sparc/s0/Solaris_8/Tools
# ./setup_install_server /jumpstart/OS/Solaris_8_2001-01
```

This command, `setup_install_server`, produces the following output:

```
 Verifying target directory...
Calculating the required disk space for the Solaris_8 product
Copying the CD image to disk...
Install Server setup complete
```

2. The first CD of the Solaris 8 OE is now installed. Insert the second CD (labeled 2 of 2) into the CD-ROM drive and enter the following command:

```
# pwd
/cdrom/sol_8_sparc_2/Solaris_8/Tools
# ./add_to_install_server /jumpstart/OS/Solaris_8_2001-01
```

This command, `add_to_install_server`, produces the following output:

```
The following Products will be copied to
/jumpstart/OS/Solairs_8_2001-01/Solaris_8/Product:

Solaris_2_of_2

If only a subset of products is needed enter Control-C
and invoke ./add_to_install_server with the -s option.

Checking required disk space...

Copying the Early Access products...
41990 blocks

Processing completed successfully.
```

This completes the installation of the required Solaris OE software to the `/jumpstart` directory hierarchy. After the Solaris 8 OE software is installed on the JumpStart server, the `/jumpstart` directory must be made available to the JumpStart clients through the NFS system.

3. Add the following line to the `/etc/dfs/dfstab` file:

```
share -F nfs -o ro,anon=0 -d “Jumpstart Directory“ /jumpstart
```

4. Enter the following command to execute the share command:

```
# shareall
```

JumpStart Client Configuration

To perform a JumpStart installation successfully, the JumpStart server, js-server01, must know the ethernet address (MAC) and IP addresses of the JumpStart client(s). This information is provided to the JumpStart server through a name service such as NIS+ or NIS—or through the use of the /etc/hosts and /etc/ethers files. The add_install_client JumpStart script uses this information to create an entry in the /etc/bootparams file. To simplify this example, the /etc/ethers and /etc/hosts files are used for this procedure.

1. Create an /etc/ethers file, and add the following line:

```
8:0:20:82:d8:8f js-client01
```

2. Add the following line (for the JumpStart client) to the/etc/hosts file:

```
192.168.250.21 js-client01
```

   The JumpStart server issues the IP address, js-server01 192.168.250.20.

3. Add the JumpStart client, js-client01, to the JumpStart boot server with following command:

```
# pwd
/jumpstart/OS/Solaris_8_2001-01/Solaris/Tools
# ./add_install_client -c js-server01:/jumpstart js-client01 sun4u
```

The previous command produces the following output:

```
making /tftpboot
enabling tftp in /etc/inetd.conf
starting rarpd
starting bootparamd
starting nfsd’s
starting nfs mount
updating /etc/bootparams
copying inetboot to /tftpboot
```

Note the add_install_client command starts any services the JumpStart server requires to function correctly (which were not running before the add_install_client command was run). For example, if the NFS server on the JumpStart server isn’t running, the add_install_client command starts it.

The JumpStart server, js-server01, is now configured to supply js-client01 with an IP address and the Solaris OE. However, until a profile configuration file and rules file are created, the JumpStart server does not know what components of the Solaris OE to offer the client; therefore, an automated JumpStart installation is not possible. Although an automated installation is *not* possible, an interactive Solaris installation can be performed at this time.

JumpStart Automation

This section describes how to configure the JumpStart software to automatically install the JumpStart client, once the basic JumpStart components are in place.

Configuring the sysidcfg File

To fully automate an installation, all required information (netmask, locale, timeserver, etc.) must be available to the installation process. This information is provided through the sysidcfg configuration file, or a name service such as NIS+.

The add_install_client command provides an option, -p, to specify the location of the sysidcfg file. This option directs the JumpStart client to use the specified sysidcfg file from the /jumpstart directory on the JumpStart server.

Use the following sysidcfg file to fully automate the installation:

```
system_locale=en_US
timezone=US/Eastern
network_interface=primary {netmask=255.255.255.0
                        protocol_ipv6=no}
terminal=vt100
security_policy=NONE
root_password=DcwyMAx8TwtL2
name_service=NONE
timeserver=localhost
```

Note – The above sysidcfg file contains keywords specific to the Solaris 8 OE (for example, protocol_ipv6, and security_policy) and this file will not work with any previous versions.

Creating the JumpStart rules File

JumpStart software uses a rules file to specify how a JumpStart client is built without using interactive responses. The rules file is a text-based configuration file that contains a rule for each group of systems (or a single system), and contains information on configuring and installing the Solaris OE.

The rules file is created by a system administrator and should contain the rules for all the different types of systems that will be installed in the environment. The following is a sample rule in a rules file:

```
hostname js-client01 - Profiles/basic.profile -
```

A rules file entry has five fields and the syntax of the rules files must follow this convention:

```
rule_keyword rule_value begin profile finish
```

A rules file entry contains *at least* a rule_keyword, a rule_value, and a profile. In addition, begin and finish scripts can be included—which are executed by the JumpStart server before (or after) the Solaris OE is installed.

The examples in this article only use the following four of the five available fields in the rules file:

- **rule_keyword** – This field defines system attributes used in the rule_value to match a system with a corresponding value. The rules file provided in this article uses the keyword *hostname*.
- **rule_value** – The value of this field is the corresponding value of the rule_keyword. The samples used in this article specify the actual hostname of the system being JumpStart(ed).
- **profile** – This field points to a separate file that contains specific Solaris OE configuration information for a client. This configuration information may include disk layouts, Solaris cluster specifics (whether the JumpStart is an initial installation or upgrade), and other relevant information.
- **finish** – The value of this field is an executable Bourne shell script that runs after the Solaris OE installation is completed. In this article, this script is a driver script which calls other scripts in the JumpStart install server directory hierarchy.

A basic rules file entry is used for the JumpStart environment described in this article. The any argument in the rules file is used for JumpStart installations when the JumpStart client has not matched any other rule in the rules file. If you add this entry to the rules file, all JumpStart clients defined on the server can be installed using this entry. To implement the any argument, create a rules file in the /jumpstart directory, on the install server, by including *only* the following entry:

```
any - - Profiles/basic.profile -
```

This entry is used in the rules file for the examples described throughout this article.

Creating the JumpStart profile File

A rules file must specify a profile file—this file defines how the Solaris OE system is installed and configured. The profile file contains profile keywords and the corresponding value for each keyword. Each profile keyword is used to define a specific component of the Solaris OE installation and configuration process.

The following is a sample profile file named basic.profile:

```
# install_type MUST be listed first
install_type    initial_install

# start with the minimal required number of packages
cluster         SUNWCuser

# define how the disk is laid out
partitioning    default
```

This example is a minimal profile file. All profile files must contain *at least* the install_type keyword. The other keywords listed in the example are not required because they have default values that are used if *no* explicit definition is made.

The rules file for this JumpStart environment uses the basic.profile file (Profiles/basic.profile) to define the components installed on the JumpStart client. Based on this profile file, the following actions are performed:

1. install_type initial_install – Installs a new Solaris OE (as opposed to an upgrade).
2. cluster SUNWCuser – Installs the Solaris OE cluster SUNWCuser (including only the packages required for an End User installation). If this variable is not specified, the SUNWCuser cluster or End User cluster is installed by default.
3. partitioning default – Configures the hard drive with the Solaris OE requirements (with default specified). If the partitioning keyword is *not* specified in the profile, the drive is partitioned *as if* the partitioning default is specified.

By convention, all profile files are stored in the /jumpstart/Profiles directory of the JumpStart install server. These files are grouped by system function. For example, all web servers use the same profile file. The goal is to have systems that perform similar tasks, have an identical physical configuration, disk layout, and OE installation. This simplifies the installation process and streamlines administration and management.

check Script

The rules file, profile configuration files, and scripts require validation after creation or modification. They are validated by running the check script that creates a rules.ok file (if no errors are detected). The rules.ok file is used by the JumpStart server to install the Solaris OE. The check script is located on the JumpStart server in the directory,

/jumpstart/OS/Solaris_8_2001-01/Solaris_8/Misc/jumpstart_sample. This script is copied to the base JumpStart directory of the JumpStart install server, /jumpstart, and then executed as follows:

```
# pwd
/jumpstart
# ./check
```

This command generates the following output:

```
Validating rules...
Validating profile Profiles/basic.profile...
The custom JumpStart configuration is ok.
```

At this point the JumpStart client, js-client01, can be JumpStart(ed) automatically. Automation is accomplished by booting js-client01 to the ok prompt and entering the following command:

```
ok> boot net - install
```

JumpStart Client Installation

This section documents the messages generated from the console of the JumpStart client and a sampling of the network traffic generated between the different JumpStart servers and the JumpStart client.

The initial JumpStart client boot messages (using the sysidcfg file) are as follows:

```
ok boot net - install
Resetting ...

Sun Ultra 1 SBus (UltraSPARC 167MHz), No Keyboard
OpenBoot 3.1, 128 MB memory installed, Serial #8575119.
Ethernet address 8:0:20:82:d8:8f, Host ID: 8082d88f.

Rebooting with command: boot net - install
Boot device: /sbus/ledma@e,8400010/le@e,8c00000 File and args: -
install
2aa00
Booting the 32-bit OS ...

SunOS Release 5.8 Version Generic 32-bit
Copyright 1983-2000 Sun Microsystems, Inc.  All rights reserved.
whoami: no domain name
Configuring /dev and /devices
Using RPC Bootparams for network configuration information.
Configured interface le0
Using sysid configuration file 10.0.0.20:/jumpstart/sysidcfg
The system is coming up.  Please wait.
Starting remote procedure call (RPC) services: sysidns done.
Starting Solaris installation program...
Searching for JumpStart directory...
Using rules.ok from 10.0.0.20:/jumpstart.
Checking rules.ok file...
Using profile: Profiles/basic.profile
Using finish script: Drivers/bp-iplanetes.driver
Executing JumpStart preinstall phase...
Searching for SolStart directory...
Checking rules.ok file...
Using begin script: install_begin
Using finish script: patch_finish
Executing SolStart preinstall phase...
Executing begin script "install_begin"...
Begin script install_begin execution completed.
```

Summary

This article discussed how the core JumpStart components interact and provided recommendations on how to structure the JumpStart server architecture. In addition, this article provided step-by-step instructions on how to get a basic automated JumpStart environment up and running as quickly as possible.

Bibliography

Solaris Advanced Installation Guide, Sun Microsystems,
`http://docs.sun.com`

Noordergraaf, Alex and Brunette, Glenn, *JumpStart Architecture and Security Scripts for the Solaris Operating Environment - Part 1: Updated for Toolkit version 0.2,* Sun BluePrints OnLine, November 2000,
`http://www.sun.com/blueprints/1000/jssec-updt1.pdf`

Author's Bio: Alex Noordergraaf

Alex Noordergraaf has over nine years experience in the area of Computer and Network Security. As a Senior Staff Engineer in the Enterprise Engineering group of Sun Microsystems he is developing, documenting, and publishing security best practices through the Sun BluePrints OnLine program. He is a co-author of the freeware "JASS" Security Toolkit, which is available from http://www.sun.com/blueprints/tools.

Prior to his role in Enterprise Engineering he was a Senior Security Architect with Sun Professional Services where he worked with many Fortune 500 companies on projects that included security assessments, architecture development, architectural reviews, and policy/procedure review and development. In addition to providing billable services to customers he developed and delivered an Enterprise security assessment methodology and training curriculum to be used worldwide by SunPS. His customers have included major telecommunication firms, financial institutions, ISPs, and ASPs.