

**Transitioning From Oracle® Solaris  
10 JumpStart to Oracle Solaris 11.3  
Automated Installer**

**ORACLE®**

**Part No: E54837**  
April 2020



## Transitioning From Oracle Solaris 10 JumpStart to Oracle Solaris 11.3 Automated Installer

### Part No: E54837

Copyright © 2011, 2020, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

**Référence: E54837**

Copyright © 2011, 2020, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou ce matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

**Accès aux services de support Oracle**

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> si vous êtes malentendant.

# Contents

---

- Using This Documentation** ..... 9
  
- 1 Comparing JumpStart and Automated Installer** ..... 11
  - Similarities and Differences Between JumpStart and AI ..... 11
  - JumpStart to AI Migration Information ..... 12
  
- 2 Converting JumpStart rules File and Profiles** ..... 15
  - About AI Client Criteria and AI Manifests ..... 15
  - Comparing rules File Keywords and AI Criteria Keywords ..... 15
  - Comparing Profile Keywords and AI Manifest Elements ..... 17
  - Using js2ai to Convert JumpStart rules File and Profiles to AI Criteria and Manifests ..... 22
    - How js2ai Converts JumpStart Device Specifications ..... 23
    - Converting Software Packages ..... 25
    - rules File and Profiles Conversion Using js2ai ..... 28
    - Displaying Additional Profile Conversion Information ..... 37
    - Validating an Output AI Manifest ..... 38
  
- 3 Converting sysidcfg Configuration Files** ..... 39
  - Comparing sysidcfg Keywords to System Configuration Profile Directives ..... 39
  - Using js2ai to Convert sysidcfg Files to System Configuration Profiles ..... 41
    - js2ai Conversion Warning ..... 42
    - sysidcfg Conversion Using js2ai ..... 43
    - Displaying Additional Configuration Conversion Information ..... 45
    - Validating an Output System Configuration Profile ..... 46

<b>4 Installing Oracle Solaris 10 by Using JumpStart on an Oracle Solaris 11 Server</b> .....	47
Setting Up an Oracle Solaris 11 System as an Oracle Solaris 10 JumpStart Server .....	47
▼ How to Set Up an Oracle Solaris 11 System as an Oracle Solaris 10 JumpStart Server .....	47
<b>Index</b> .....	53

## Tables

---

<b>TABLE 1</b>	Comparing JumpStart and AI Tasks .....	11
<b>TABLE 2</b>	Comparing JumpStart rules File and AI Criteria Keywords .....	15
<b>TABLE 3</b>	Comparing JumpStart Profile Keywords and AI Manifest Elements .....	17
<b>TABLE 4</b>	Comparing sysidcfg Keywords and Configuration Profile Directives .....	39





## Using This Documentation

---

- **Overview** – Describes how to migrate from Oracle Solaris 10 JumpStart to Oracle Solaris 11 Automated Installer
- **Audience** – Technicians, system administrators, and authorized service providers
- **Required knowledge** – Experience administering an Oracle Solaris system

## Product Documentation Library

Documentation and resources for this product and related products are available at <http://www.oracle.com/pls/topic/lookup?ctx=E53394-01>.

## Feedback

Provide feedback about this documentation at <http://www.oracle.com/goto/docfeedback>.



# Comparing JumpStart and Automated Installer

---

This chapter describes the similarities and differences between JumpStart and Automated Installer (AI). It also provides resources that can help you migrate from JumpStart to AI.

## Similarities and Differences Between JumpStart and AI

Both JumpStart and AI provide hands-free installation of multiple systems on a network. Clients are network booted, and once the client is booted, the installer takes over.

JumpStart installs the Oracle Solaris 10 OS and earlier versions of the Oracle Solaris OS. AI installs the Oracle Solaris 11 OS and update releases.

JumpStart and AI share the following characteristics:

- Provide hands-free network installation of multiple clients by storing system configurations on an install server
- Provide for different kinds of installations on different clients in one automated installation
- Install both x86 and SPARC clients

**TABLE 1** Comparing JumpStart and AI Tasks

Step	JumpStart	AI
Set up an install server.	Use the <code>setup_install_server</code> command.	Use the <code>installadm create-service</code> command.
Add clients to the installation.	Use the <code>add_install_client</code> command.	Use the <code>installadm create-client</code> command.
Create derived profiles	Use a <code>begin</code> script.	Use the derived manifests mechanism.
Specify installation instructions.	Use profile files.	Use AI manifest files.
Provision the client.	Use DVDs.	Use a package repository on the Internet or on a local network.
Specify client customizations.	Use rules files to associate clients with profile files.	Use the <code>create-manifest</code> , <code>create-profile</code> , or <code>set-criteria</code> subcommands of the <code>installadm</code>

Step	JumpStart	AI
Specify client configuration.	Use finish scripts and sysidcfg files.	command to associate clients with AI manifests and system configuration profiles. Use SMF (Oracle Solaris Service Management Facility) system configuration profiles. Use scripts executed by an SMF service that runs once at first boot.

## JumpStart to AI Migration Information

Which path to migrate from JumpStart to AI is best for your environment depends on the resources you have available and how completely you would like to configure the AI client. A more complete AI client configuration can require extra steps. Some of the strategies that you could use when migrating to use the Automated Installer include the following:

- **Use one server as both a JumpStart install server and an AI install server.**  
This configuration allows you to support JumpStart capable systems from an Oracle Solaris 11 server. For more information, see [Chapter 4, “Installing Oracle Solaris 10 by Using JumpStart on an Oracle Solaris 11 Server”](#).
- **Use a software package repository for AI installations.**
  - Use an Oracle Solaris 11 package repository on the Internet, such as [pkg.oracle.com](http://pkg.oracle.com).
  - Make a local copy of a package repository, as described in [Copying and Creating Package Repositories in Oracle Solaris 11.3](#).
- **Create a custom ISO image.**  
See [Creating a Custom Oracle Solaris 11.3 Installation Image](#) for information about creating a custom installation image.
- **Convert JumpStart rules, profiles, and configuration files to AI criteria, AI manifests, and SMF system configuration profiles.**  
The following resources provide instructions on how to convert JumpStart data to AI data:
  - [js2ai\(1M\)](#) man page
  - [Chapter 2, “Converting JumpStart rules File and Profiles”](#)
  - [Chapter 3, “Converting sysidcfg Configuration Files”](#)
- **Dynamically derive an AI client provisioning manifest.**  
JumpStart begin scripts provide the ability to dynamically manipulate installation parameters that are passed to the installer. AI provides the ability to query client attributes at client installation time and dynamically derive a provisioning manifest customized for that client. Environment variables specify hardware attributes of the client, and most of

these attributes are the same as environment variables used with JumpStart begin scripts. See “Creating an AI Manifest at Client Installation Time” in *Installing Oracle Solaris 11.3 Systems*.

- **Provide system configuration instructions.**

See Chapter 11, “Defining AI Client System Configuration Parameters” in *Installing Oracle Solaris 11.3 Systems* for information about creating SMF profiles.

See Chapter 12, “Installing and Configuring Zones” in *Installing Oracle Solaris 11.3 Systems* for information about creating an AI manifest and SMF profiles to install non-global zones as part of an AI client installation.

- **Create an SMF service that runs once at first boot and executes a user-defined script.**

See Chapter 13, “Running a Custom Script During First Boot” in *Installing Oracle Solaris 11.3 Systems*.



## Converting JumpStart rules File and Profiles

---

This chapter describes how to use the `js2ai` command to convert JumpStart rules and profiles to AI criteria files and AI manifests.

### About AI Client Criteria and AI Manifests

Automated Installer (AI) manifests are XML files that specify system provisioning such as disk layout and software packages to install. See [Chapter 10, “Defining AI Client Installation Parameters” in \*Installing Oracle Solaris 11.3 Systems\*](#) and the `ai_manifest(4)` man page for more information about AI manifests.

AI uses client criteria to select the AI manifests that client systems should use to complete their installation. When an AI manifest is added to an AI install service, you can specify criteria on the command line or in a file. See [Chapter 9, “Assigning Customizations to AI Clients” in \*Installing Oracle Solaris 11.3 Systems\*](#) for more information about specifying AI criteria.

### Comparing rules File Keywords and AI Criteria Keywords

The following table compares JumpStart rules file keywords with AI criteria keywords. AI uses these criteria to apply the correct AI manifest or system configuration profile to a particular client.

**TABLE 2** Comparing JumpStart rules File and AI Criteria Keywords

JumpStart rules File Keyword	AI Criteria File Keyword	Command-Line Example	Criteria File Example
any	Unsupported. For client systems that do not match any		

Comparing rules File Keywords and AI Criteria Keywords

JumpStart rules File Keyword	AI Criteria File Keyword	Command-Line Example	Criteria File Example
	selection criteria, the AI install service provides a default AI manifest.		
arch	cpu	-c cpu="sparc"	<ai_criteria name="cpu"> <value>sparc</value> </ai_criteria>
disksize	Unsupported. AI installs on the target specified in the AI manifest if that disk is at least the minimum required size.		
domainname	Unsupported.		
hostaddress	ipv4	-c ipv4="10.6.68.127"	<ai_criteria name="ipv4"> <value>10.6.68.127</value> </ai_criteria>
hostname	Unsupported. To uniquely identify a host in AI, use either the IP address as described in hostaddress or use the MAC address.	-c mac="0:3:ba:33:9d:b6"	<ai_criteria name="mac"> <value>0:3:ba:33:9d:b6</value> </ai_criteria>
installed	Unsupported. AI installs on the target specified in the AI manifest if that disk is at least the minimum required size.		
karch	arch	-c arch="i86pc"	<ai_criteria name="arch"> <value>i86pc</value> </ai_criteria>
memsize	mem	-c mem="2048"	<ai_criteria name="mem"> <value>2048</value> </ai_criteria>
model	platform	-c platform="SUNW,Sun-Fire-T200"	<ai_criteria name="platform"> <value>SUNW,Sun-Fire-T200</value> </ai_criteria>
network - for network addresses	For an individual network address or a range of addresses within one network address, use ipv4	single network address: -c ipv4="10.6.68.127"  range of addresses: -c ipv4="10.0.0.1-10.0.0.64"	single network address:  <ai_criteria name="ipv4"> <value>10.6.68.127</value> </ai_criteria>  range of addresses:  <ai_criteria name="ipv4"> <range>10.0.0.1 10.0.0.64</range> </ai_criteria>
network - for network numbers	For an individual network number or a range of network numbers, use network	single network number: -c network="10.0.0.0"  range of networks: -c network="10.0.0.0-12.0.0.0"	single network number:  <ai_criteria name="network"> <value>10.0.0.0</value> </ai_criteria>  range of networks:



JumpStart <code>rules</code> File Keyword	AI Criteria File Keyword	Command-Line Example	Criteria File Example
			<pre>&lt;ai_criteria name="network"&gt;   &lt;range&gt;     10.0.0.0     12.0.0.0   &lt;/range&gt; &lt;/ai_criteria&gt;</pre>
<code>osname</code>	Unsupported.		
<code>probe</code>	Unsupported.		
<code>totaldisk</code>	Unsupported. AI installs on the target specified in the AI manifest if that disk is at least the minimum required size.		

## Comparing Profile Keywords and AI Manifest Elements

The following table compares JumpStart profile keywords with AI manifest elements. AI uses XML manifest files to define the client installation. For more information about AI manifests, see [Chapter 10, “Defining AI Client Installation Parameters” in \*Installing Oracle Solaris 11.3 Systems\*](#) and see the `ai_manifest(4)` man page.

To specify values that are not known until the client installation process has started, such as devices specified using the any keyword, consider using a derived manifests script. You can use a derived manifests script to specify swap size based on disk size or to specify mirroring based on available disks, for example. For information about derived manifests scripts, see [“Creating an AI Manifest at Client Installation Time” in \*Installing Oracle Solaris 11.3 Systems\*](#).

**TABLE 3** Comparing JumpStart Profile Keywords and AI Manifest Elements

JumpStart Profile Keyword	AI Manifest Elements
<code>archive_location</code>	Unsupported. AI does not install from flash archives.
<code>backup_media</code>	Unsupported. The <code>backup_media</code> keyword is used only with the upgrade option of <code>install_type</code> . AI does not support the upgrade install type; AI supports only initial install.
<code>boot_device device</code>  Example:  <code>boot_device c1t0d0</code>	Example:  <pre>&lt;target&gt;   &lt;disk whole_disk="true"&gt;     &lt;disk_name name="c1t0d0" name_type="ctd"/&gt;   &lt;/disk&gt;   &lt;logical nodump="true" noswap="false"/&gt; &lt;/target&gt;</pre>
<code>boot_device device eeprom</code>	The value for the <code>eeprom</code> keyword (update for SPARC systems and preserve for x86 systems) is not supported in AI. In AI, the EEPROM on SPARC

## Comparing Profile Keywords and AI Manifest Elements

JumpStart Profile Keyword	AI Manifest Elements
<p>Example:</p> <pre>boot_device c0t0d0s0 update</pre>	<p>systems is always updated to the specified target device so that the installed system automatically boots from that device. On x86 systems, the firmware is never updated.</p>
bootenv	Unsupported. AI creates a boot environment and installs the Oracle Solaris 11 OS into that boot environment.
client_arch	Unsupported. The client architecture to be installed is defined by the AI install service, not in the AI manifest.
client_root	Unsupported. You can specify the type, vendor, and size of the target disk. You cannot specify the amount of root space on the target disk. See the <a href="#">ai_manifest(4)</a> man page.
client_swap	Unsupported. By default, AI creates a swap volume in the root pool. You can specify a swap slice. See the <a href="#">ai_manifest(4)</a> man page.
cluster <i>cluster-name</i>	<p>Unsupported. See “<a href="#">Converting Software Packages</a>” on page 25 for information about how to handle cluster and package specifications.</p> <p>The Oracle Solaris 11 OS uses group packages as defined in the <a href="#">pkg(5)</a> man page. Group packages are specified just as any other package is specified in the manifest. The default AI manifest includes the packages needed for a standard Oracle Solaris 11 installation. You can customize this list of packages.</p>
cluster <i>cluster-name</i> delete	Unsupported. The delete switch is used only with the upgrade option of <code>install_type</code> . AI does not support the upgrade install type; AI supports only initial install.
dontuse	Unsupported.
fdisk <i>disk-name type size</i>	<p>Example:</p> <pre>&lt;target&gt;   &lt;disk&gt;     &lt;disk_name name="c0t3d0" name_type="ctd"/&gt;     &lt;partition action="create" name="1" part_type="191"/&gt;   &lt;/disk&gt;   &lt;logical nodump="true" noswap="false"/&gt; &lt;/target&gt;</pre> <p>For a full list of the disk and partition attributes supported by AI, see the <a href="#">ai_manifest(4)</a> man page.</p> <p>Translation by js2ai:</p> <p>For js2ai translation, the value of <i>disk-name</i> must be a device. A device of all is not supported. The <i>fdisk type</i> must be <code>solaris</code>. A size of 0 or <code>delete</code> is not supported. If partitioning is default and the <code>rootdisk</code> has not been set, js2ai sets the first <code>fdisk solaris</code> partition encountered as the root disk.</p>
filesystem	<p>UFS file systems are not supported. AI installs ZFS file systems.</p> <p>Translation by js2ai:</p> <p>If there is no other way to determine the device to be used for root, the device from the <code>filesystem</code> line with the / mount point is used for the root pool.</p>

JumpStart Profile Keyword	AI Manifest Elements
	<p>The local and mirrored file systems are supported when the mount point specified is / or swap.</p> <p>If the mount point is not / or swap, the line is logged and then ignored. JumpStart <i>fsoptions</i> are not supported.</p> <p>No validation of the size is performed. You might need to adjust the size specified in the resulting AI manifest to achieve a successful installation with this manifest.</p>
<p>Example:</p> <pre>fileysys c1t0d0s0 10000 /</pre>	<p>The following partial AI manifest is for x86 platforms.</p> <pre>&lt;target&gt;   &lt;disk&gt;     &lt;disk_name name="c1t0d0" name_type="ctd"/&gt;     &lt;partition action="create" name="1" part_type="191"&gt;       &lt;slice action="create" force="true"         in_vdev="rpool_vdev" in_zpool="rpool" name="0"&gt;         &lt;size val="10000mb"/&gt;       &lt;/slice&gt;     &lt;/partition&gt;   &lt;/disk&gt;   &lt;logical nodump="true" noswap="false"&gt;     &lt;zpool is_root="true" name="rpool"&gt;       &lt;vdev name="rpool_vdev" redundancy="none"/&gt;     &lt;/zpool&gt;   &lt;/logical&gt; &lt;/target&gt;</pre> <p>Translation by js2ai:</p> <p>The js2ai command supports translations only of the root file system (/) and swap.</p>
<p>Example:</p> <pre>fileysys mirror:rpool c6t0d0s0 c6t1d0s0 60048 /</pre>	<p>The following partial AI manifest is for x86 platforms. For SPARC platforms, no partition element is specified.</p> <pre>&lt;target&gt;   &lt;disk&gt;     &lt;disk_name name="c6t1d0" name_type="ctd"/&gt;     &lt;partition action="create" name="1" part_type="191"&gt;       &lt;slice action="create" force="true"         in_vdev="rpool_vdev" in_zpool="rpool" name="0"&gt;         &lt;size val="60048mb"/&gt;       &lt;/slice&gt;     &lt;/partition&gt;   &lt;/disk&gt;   &lt;disk&gt;     &lt;disk_name name="c6t0d0" name_type="ctd"/&gt;     &lt;partition action="create" name="1" part_type="191"&gt;       &lt;slice action="create" force="true"         in_vdev="rpool_vdev" in_zpool="rpool" name="0"&gt;         &lt;size val="60048mb"/&gt;       &lt;/slice&gt;     &lt;/partition&gt;   &lt;/disk&gt;   &lt;logical nodump="true" noswap="false"&gt;     &lt;zpool is_root="true" name="rpool"&gt;       &lt;vdev name="rpool_vdev" redundancy="mirror"/&gt;     &lt;/zpool&gt;   &lt;/logical&gt; &lt;/target&gt;</pre>

## Comparing Profile Keywords and AI Manifest Elements

JumpStart Profile Keyword	AI Manifest Elements
	<pre>&lt;/logical&gt; &lt;/target&gt;</pre>
fileysys <i>server:path</i>	Unsupported. AI does not support remote file systems.
forced_deployment	Unsupported. AI does not install from flash archives.
geo	Unsupported. In AI, geographic regions for language support are specified in the system configuration profile. See <a href="#">Chapter 11, “Defining AI Client System Configuration Parameters”</a> in <i>Installing Oracle Solaris 11.3 Systems</i> .
install_type	AI supports only <code>initial_install</code> .
layout_constraint	Unsupported. AI supports only <code>initial_install</code> .
local_customization	Unsupported. AI does not install from flash archives.
locale	Unsupported. In AI, locale support is specified in the system configuration profile. See <a href="#">Chapter 11, “Defining AI Client System Configuration Parameters”</a> in <i>Installing Oracle Solaris 11.3 Systems</i> .
metadb	Unsupported.
no_content_check	Unsupported. AI does not install from flash archives.
no_master_check	Unsupported. AI does not install from flash archives.
num_clients	Unsupported.
package <i>package</i>  Example:  package SUNWpampkcs11 add	Example:  <pre>&lt;software type="IPS"&gt;   &lt;software_data action="install"&gt;     &lt;name&gt;pkg:/SUNWpampkcs11&lt;/name&gt;   &lt;/software_data&gt; &lt;/software&gt;</pre> <p>See <a href="#">“Converting Software Packages”</a> on page 25 for information about how to verify the package name is valid.</p>
package <i>package</i> add nfs <i>server:path</i>	<p>In an AI manifest, all software is retrieved from IPS (Image Packaging System) package repositories. IPS repository locations can be a URL or file name. You can specify multiple IPS repositories as software sources. See the <a href="#">ai_manifest(4)</a> man page for information about how to specify publishers.</p> <p>Translation by js2ai:</p> <p>The js2ai command ignores the <i>retrieval-type</i> location argument.</p>
package <i>package</i> delete	Example:  <pre>&lt;software type="IPS"&gt;   &lt;software_data action="uninstall"&gt;     &lt;name&gt;pkg:/SUNWpampkcs11&lt;/name&gt;   &lt;/software_data&gt; &lt;/software&gt;</pre> <p>Because AI only supports initial installations, you should not need to use an AI manifest to delete packages.</p>

JumpStart Profile Keyword	AI Manifest Elements
<p>partitioning default</p> <p>partitioning explicit</p>	<p>If no target installation device is specified, AI determines a default installation device and a default partitioning. To specify custom partitioning, see the <a href="#">ai_manifest(4)</a> man page.</p> <p>Translation by js2ai:</p> <p>For js2ai, partitioning must be default or explicit. For explicit, only swap and / are supported</p>
<p>partitioning existing</p>	<p>Unsupported. For js2ai, partitioning must be default or explicit.</p>
<p>patch</p>	<p>Unsupported. AI supports only initial_install.</p> <p>To update your system or specific packages, use the pkg update command on the installed system.</p>
<p>pool <i>newpool</i> auto auto auto <i>device</i></p>	<p>Example:</p> <pre data-bbox="711 785 1235 1104"> &lt;target&gt;   &lt;disk&gt;     &lt;disk_name name="c1t0d0" name_type="ctd"/&gt;     &lt;partition action="create" name="1" part_type="191"&gt;       &lt;slice action="create" force="true"         in_vdev="rpool_vdev" in_zpool="newpool" name="0"/&gt;     &lt;/partition&gt;   &lt;/disk&gt;   &lt;logical nodump="false" noswap="false"&gt;     &lt;zpool is_root="true" name="newpool"&gt;       &lt;vdev name="rpool_vdev" redundancy="none"/&gt;     &lt;/zpool&gt;   &lt;/logical&gt; &lt;/target&gt; </pre> <p>Translation by js2ai:</p> <p>If a pool is specified in a profile, js2ai creates the ZFS root pool using the specified devices. The pool keyword supersedes all other keywords when js2ai determines which devices to use for the ZFS root pool.</p> <p>The js2ai command does not perform any validation of the pool size, swap size, or dump size. You might need to adjust these sizes in the resulting AI manifest to achieve a successful installation with this manifest.</p>
<p>pool <i>newpool</i> auto auto auto any</p>	<p>For js2ai, if you specify any instead of a physical device name, you must provide device information prior to the specification that includes the any parameter. For example, you could provide a root_device or usedisk specification before this pool specification. See <a href="#">Example 6, "Fixing mirrorpool.profile Errors,"</a> on page 34 for an example.</p>
<p>root_device c1t0d0s0</p>	<p>Example:</p> <pre data-bbox="711 1558 1227 1711"> &lt;target&gt;   &lt;disk&gt;     &lt;disk_name name="c1t0d0" name_type="ctd"/&gt;     &lt;partition action="create" name="1" part_type="191"&gt;       &lt;slice action="create" force="true"         in_vdev="rpool_vdev" in_zpool="rpool" name="0"/&gt;     &lt;/partition&gt; </pre>

JumpStart Profile Keyword	AI Manifest Elements
	<pre> &lt;/disk&gt; &lt;logical nodump="true" noswap="false"&gt;   &lt;zpool is_root="true" name="rpool"&gt;     &lt;vdev name="rpool_vdev" redundancy="none"/&gt;   &lt;/zpool&gt; &lt;/logical&gt; &lt;/target&gt; </pre> <p>Translation by js2ai:</p> <p>When <code>root_device</code> is specified, js2ai sets the <code>rootdisk</code> to the specified device.</p>
<code>system_type</code>	<p>AI manifests do not differentiate system types.</p> <p>Translation by js2ai:</p> <p>Only the value <code>standalone</code> is supported.</p>
<code>usedisk</code>	<p>Translation by js2ai:</p> <p>The js2ai command might use the specified device or devices to resolve subsequent any or <code>rootdisk</code> specifications. Devices specified that are not used for this purpose are added to the ZFS root pool by js2ai when that pool is not mirrored.</p>

## Using js2ai to Convert JumpStart rules File and Profiles to AI Criteria and Manifests

Use the `js2ai` command with the `-r` option to convert both JumpStart rules and their associated profiles to AI criteria and manifests. Initially, use the `-S` option to skip validation. This command performs a conversion operation on the `rules` file and the profiles referenced by the `rules` file.

Each profile referenced in the `rules` file is processed against the AI client provisioning manifest, `/usr/share/auto_install/manifest/default.xml`. This step creates a directory named `AI_profile-name` for each profile specified in the JumpStart `rules` file. The `AI_profile-name` directory contains an AI criteria file in the form `criteria-rule-number.xml` that corresponds to the rule that referenced this profile. The `AI_profile-name` directory also contains AI manifest files in the form `profile-name.arch.xml` that correspond to the `profile-name` file.

When you receive a message that the conversion completed successfully, run the `js2ai` command without the `-S` option to validate the output AI manifests. Validation errors must be corrected in the AI manifest files.

If you do not see a message that the conversion was successfully completed, examine the error report and the `js2ai.log` file. The error report and the log file provide warnings, process errors, unsupported items, conversion errors, and validation errors. The error report is a table output to `stdout` that shows the number of each type of error that was encountered in converting the rules file and profiles. The log file describes the problems.

1. Correct any process errors.
2. Remove any lines from the `rules` file and profiles that are listed as unsupported items.
3. Examine the conversion errors and correct the errors if possible. Otherwise, remove the lines that are causing the errors.
4. Examine any warning messages and make sure no corrections are necessary.

Note the following options for the `js2ai` command:

- To display more information about a rule file or profile conversion, use the `-v` option. For more information, see [“Displaying Additional Profile Conversion Information” on page 37](#).
- To validate a specific output AI manifest against the appropriate AI DTD, use the `-V` option. For more information, see [“Validating an Output AI Manifest” on page 38](#).
- To convert one JumpStart profile only, use the `-p` option. See the examples in [“Fixing AI Manifest Files” on page 32](#).

## How js2ai Converts JumpStart Device Specifications

This section describes how `js2ai` determines some target elements for the AI manifest from the JumpStart profile specifications.

### How the System's Root Disk Is Determined

Because `js2ai` does not have access to the client system profile references during the profile translation process, `js2ai` attempts to determine the root disk during translation using a process that matches JumpStart as much as possible.

The `js2ai` command performs the following steps to determine what device to use for the root disk:

1. If the `root_device` keyword is specified in the profile, `js2ai` sets `rootdisk` to the device on which the slice resides.

2. If `rootdisk` is not set and the `boot_device` keyword is specified in the profile, `js2ai` sets `rootdisk` to the boot device.
3. If `rootdisk` is not set, `partitioning default` is specified, and a `solaris fdisk` entry is encountered, `js2ai` sets `rootdisk` to the specified disk name.
4. If `rootdisk` is not set and a `filesys cwtxdysz size /` entry is specified in the profile, `js2ai` sets `rootdisk` to the `cwtxdysz` disk specified in the entry.
5. If `rootdisk` is not set and a `usedisk disk-name` entry is specified in the profile, `js2ai` sets `rootdisk` to the `disk-name` disk specified in the entry.
6. If `rootdisk` is not set and the following specification is encountered in the profile where `size` is not 0 or `delete` and `disk-name` is not `all`, then `rootdisk` is set to this disk name.

`fdisk disk-name solaris size`

7. If `rootdisk` is not set, any occurrence where the device is specified as `rootdisk` generates a conversion error.

## How the any Keyword Is Translated

The `js2ai` command performs the following steps to determine what device to use when the any keyword is specified:

1. If the device is specified with the any keyword and the keyword action specified (non-mirrored pool, or `filesys` with a / mount point), the device is set to `rootdisk` if `rootdisk` is set.
2. If the device has not been translated and a `usedisk` statement exists in the profile, the device specified with the any keyword is set to the device specified by the `usedisk` statement.
3. If the device has not been translated and the action where the any keyword is specified causes the ZFS root pool to be created, AI chooses the device unless a mirrored pool is specified.

## How the ZFS Root Pool Is Determined

The `js2ai` command performs the following steps to determine which device to use for the ZFS root pool. Once the ZFS root pool is determined, subsequent definitions encountered are flagged as errors if they conflict with the ZFS root pool that has already been determined.

1. If the profile specifies the `pool` keyword, `js2ai` sets the ZFS root pool to the devices specified by the `pool` keyword.



2. If the ZFS root pool has not been determined and the profile specifies a file system with a mount point of /, the ZFS root pool is created using the devices specified.
3. If the ZFS root pool has not been determined, all keywords in the profile have been processed, and rootdisk is set, the ZFS root pool is created using the rootdisk device.
4. If the ZFS root pool has not been determined and the partition type is default, AI chooses the device to use for the ZFS root pool.
5. If the ZFS root pool has not been determined and no errors have occurred during processing, AI chooses the device to use for the ZFS root pool.
6. If the ZFS root pool is not a mirrored pool and one or more usedisk devices that were specified have not been used for a rootdisk or any device translation, those disks are added to the ZFS root pool.

## Converting Software Packages

JumpStart profiles use the `cluster` and `package` keywords to install software on the system. The `cluster` keyword is not supported by AI. Because in IPS the syntax to install an incorporation or group package is the same as the syntax to install any other package, if you simply change `cluster` to `package` in the JumpStart profile, the `js2ai` command creates the correct package installation specification in the AI manifest.

---

**Tip** - Verify the package names in the AI manifests. If a package specified for installation in an AI manifest is not available from any publisher origin specified in that AI manifest, then that client installation fails.

---

IPS package names are different from SVR4 package names. For example, the SVR4 package `SUNWpampkcs11` is renamed to `library/security/pam/module/pam-pkcs11` in IPS.

If an SVR4 package name exists in IPS, you can install the IPS package by using the SVR4 name. For example, if an AI manifest specifies installation of the `SUNWpampkcs11` package, the `library/security/pam/module/pam-pkcs11` package is automatically installed. In these cases, the package has been renamed.

If an SVR4 package name does not exist in IPS, you must change the package name or delete that specification from the AI manifest. For example, the `SUNWCa11` and `SUNWCuser` packages have not been renamed in IPS. If the AI manifest specifies those packages, the installation fails.

Use the `pkg list` command on an Oracle Solaris 11 system to determine whether a particular package name can be used in your AI manifest. Be sure to use the `-g` option to list packages from an IPS package repository origin that is specified in the AI manifest.

The `js2ai` command uses the `/usr/share/auto_install/manifest/default.xml` AI manifest as a base to build a new AI manifest that includes specifications from the JumpStart profile. This default AI manifest specifies installation of two packages that install the base operating system: `entire` and `solaris-large-server`. In addition to those two packages, you probably need to specify only the installation of additional tools and applications.

**EXAMPLE 1** Determining Whether a Package Name Can Be Used

In this example, the AI manifest specifies the `http://pkg.oracle.com/solaris/release` repository origin.

```
$ pkg list -af -g http://pkg.oracle.com/solaris/release SUNWCall SUNWCuser
pkg list: no packages matching 'SUNWCuser, SUNWCall' known
```

This message confirms that these two packages cannot be used in this AI manifest.

**EXAMPLE 2** Working With a Renamed Package

You can use the `pkg list` command to identify packages that have been renamed.

```
$ pkg list -af -g http://pkg.oracle.com/solaris/release SUNWpampkcs11
NAME (PUBLISHER)                                VERSION                                IFO
SUNWpampkcs11                                  0.6.0-0.133                          --r
```

The “r” in the last column indicates that this package is renamed. You can use this name in the AI manifest but you might want to use the `pkg info` command to determine the new name of the package.

See the “Renamed to” line in the following output. The `SUNWpampkcs11` package has been renamed to `library/security/pam/module/pam-pkcs11`. You might want to specify `library/security/pam/module/pam-pkcs11` in your AI manifest for greater compatibility with future Oracle Solaris updates.

```
$ pkg info -r SUNWpampkcs11
Name: SUNWpampkcs11
Summary:
State: Not installed (Renamed)
Renamed to: library/security/pam/module/pam-pkcs11@0.6.0-0.133
            consolidation/sfw/sfw-incorporation
Publisher: solaris
Version: 0.6.0
Build Release: 5.11
Branch: 0.133
Packaging Date: Wed Oct 27 18:50:11 2010
Size: 0.00 B
```

```

    FMRI: pkg://solaris/SUNWpampkcs11@0.6.0,5.11-0.133:20101027T185011Z
$ pkg info -r pam-pkcs11
    Name: library/security/pam/module/pam-pkcs11
    Summary: The OpenSC PKCS#11 PAM Login Tools
    Category: System/Security
    State: Not installed
    Publisher: solaris
    Version: 0.6.0
    Build Release: 5.11
    Branch: 0.175.2.0.0.27.0
    Packaging Date: Mon Nov 11 17:47:35 2013
    Size: 1.74 MB
    FMRI: pkg://solaris/library/security/pam/module/pam-pkcs11@0...

```

### EXAMPLE 3 Working With an Obsolete Package

This example shows how you would replace SUNWmysql with database/mysql-51 in your AI manifest.

```

$ pkg list -af -g http://pkg.oracle.com/solaris/release SUNWmysql
NAME (PUBLISHER)                                VERSION                                         IFO
SUNWmysql                                           4.0.24-0.142                                  --o

```

The “o” in the last column indicates that this package is obsolete. This package name cannot be used in an AI manifest. Use the `pkg list` command with wildcards or the `pkg search` command to determine whether the package is available under a different name that can be used.

```

$ pkg list -af SUNWmysql*
NAME (PUBLISHER)                                VERSION                                         IFO
SUNWmysql                                           4.0.24-0.142                                  --o
SUNWmysql-base                                     0.5.11-0.133                                  --r
SUNWmysql-python                                   0.5.11-0.162                                  --o
SUNWmysql-python26                                 0.5.11-0.133                                  --r
SUNWmysql5                                         5.0.86-0.171                                  --o
SUNWmysql5                                         5.0.86-0.133                                  --r
SUNWmysql51                                        5.1.37-0.133                                  --r
SUNWmysql51lib                                     5.1.37-0.133                                  --r
SUNWmysql51test                                    5.1.37-0.133                                  --r
SUNWmysql5jdbc                                    5.1.5-0.171                                   --o
SUNWmysql5jdbc                                    5.1.5-0.133                                   --r
SUNWmysql5test                                     5.0.86-0.171                                  --o
SUNWmysql5test                                     5.0.86-0.133                                  --r
SUNWmysqlt                                         4.0.24-0.142                                  --o
$ pkg info -r SUNWmysql51
    Name: SUNWmysql51

```

```
Summary:
  State: Not installed (Renamed)
Renamed to: database/mysql-51@5.1.37-0.133
            consolidation/sfw/sfw-incorporation
Publisher: solaris
Version: 5.1.37
Build Release: 5.11
Branch: 0.133
Packaging Date: Wed Oct 27 18:49:18 2010
Size: 0.00 B
FMRI: pkg://solaris/SUNWmysql51@5.1.37,5.11-0.133:20101027T184918Z
```

## rules File and Profiles Conversion Using js2ai

This section shows using a single js2ai command to convert a JumpStart rules file and all the profiles referenced by that rules file. By default, each converted profile is output to a directory named *AI\_profile-filename*. The rule to select that profile is output to that same directory as an AI criteria file.

### Sample JumpStart rules File and Profiles

The examples in this section use the following rules file:

```
# The following rule matches only one system:

hostname sample_host - fdisk.profile -

# The following rule matches only one system:

hostaddress 10.6.68.127 - mirrorfileys.profile -

# The following rule matches any system that is on the 924.222.43.0 network:

network 924.222.43.0 - rootdisk.profile -

# The following rule matches all x86 systems:

arch i386 - mirrorpool.profile -
```

The fdisk.profile file has the following content:

```
install_type initial_install
system_type server
root_device c1t0d0s0
```

```

usedisk c1t0d0
fdisk rootdisk solaris all
partitioning explicit
filesystems rootdisk.s1 5000 swap
filesystems rootdisk.s0 10000 /
cluster SUNWCall

```

The mirrorfilesystems.profile file has the following content:

```

install_type initial_install
partitioning default
filesystems mirror c6t0d0s0 c6t1d0s0 60048 /
cluster SUNWCuser

```

The mirrorpool.profile file has the following content:

```

install_type initial_install
partitioning default
pool newpool auto auto auto mirror any any
cluster SUNWCuser

```

The rootdisk.profile file has the following content:

```

install_type initial_install
partitioning explicit
filesystems rootdisk.s0 15000 /
filesystems rootdisk.s1 1000 swap
cluster SUNWCall

```

## Using js2ai to Process a rules File With Associated Profiles

Use the following command to process the rules file. In the error report, validation errors are shown as a hyphen character because validation was not done. Validation is suppressed by the -S option.

```

# js2ai -rS

```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
rules	0	0	1	0	-
fdisk.profile	0	0	2	0	-
mirrorfilesystems.profile	0	0	2	0	-
mirrorpool.profile	0	0	1	1	-
rootdisk.profile	0	0	1	2	-

Conversion completed. One or more failures and/or warnings occurred.

```
For errors see js2ai.log
# cat js2ai.log
rules:line 3:UNSUPPORTED: unsupported keyword: hostname
fdisk.profile:line 2:UNSUPPORTED: unsupported value for 'system_type' specified: server
fdisk.profile:line 9:UNSUPPORTED: unsupported keyword: cluster
mirrorfileys.profile:line 3:UNSUPPORTED: unsupported mount point of 'unnamed'
specified,
mount points other than '/' and 'swap' are not supported
mirrorfileys.profile:line 4:UNSUPPORTED: unsupported keyword: cluster
mirrorpool.profile:line 3:CONVERSION: unable to convert 'any' device to physical device.
Replace 'any'
with actual device name
mirrorpool.profile:line 4:UNSUPPORTED: unsupported keyword: cluster
rootdisk.profile:line 3:CONVERSION: unable to convert 'rootdisk.s0'. Replace 'rootdisk.'
with actual
device name
rootdisk.profile:line 4:CONVERSION: unable to convert 'rootdisk.s1'. Replace 'rootdisk.'
with actual
device name
rootdisk.profile:line 5:UNSUPPORTED: unsupported keyword: cluster
```

Output is stored in directories named *AI\_profile-filename*. AI criteria files created from the JumpStart rules are named for the position of the rule in the rules file. AI manifests are named *profile-filename.arch.xml*, where *arch* is generic, x86, or sparc.

```
# ls AI_*
AI_fdisk.profile:
fdisk.profile.x86.xml

AI_mirrorfileys.profile:
criteria-2.xml      mirrorfileys.profile.generic.xml

AI_mirrorpool.profile:
criteria-2.xml      mirrorpool.profile.generic.xml

AI_rootdisk.profile:
criteria-3.xml      rootdisk.profile.generic.xml
```

## Replacing the hostname Keyword

The log file reports that the JumpStart rule keyword `hostname` is not supported as an AI criteria keyword for selecting an AI manifest. The `hostname` keyword specifies which clients should use the `fdisk.profile` profile. Because `hostname` is not a supported keyword for selecting AI manifests, the new `AI_fdisk.profile` directory does not contain an AI criteria file.

You could fix this problem by changing `hostname` to `hostaddress` in the rules file.

You could also fix this problem by creating an AI criteria file that specifies a MAC address or IP address to identify the hostname system. For example, the following criteria file is equivalent to the JumpStart rule `hostname sample_host` if `0:14:4F:20:53:97` is the MAC address of `sample_host`:

```
<?xml version="1.0" encoding="utf-8"?>
<ai_criteria_manifest>
  <ai_criteria name="mac">
    <value>0:14:4F:20:53:97</value>
  </ai_criteria>
</ai_criteria_manifest>
```

To find the MAC address of a system, use the `dladm` command as described in the [dladm\(1M\)](#) man page.

For the JumpStart rule `hostaddress 10.6.68.127`, the `js2ai` command automatically created the AI criteria file `AI_mirrorfilesys.profile/criteria-2.xml`, replacing the JumpStart `hostaddress` keyword with the AI `ipv4` keyword:

```
<?xml version="1.0" encoding="utf-8"?>
<ai_criteria_manifest>
  <ai_criteria name="ipv4">
    <value>
      10.6.68.127
    </value>
  </ai_criteria>
</ai_criteria_manifest>
```

For the JumpStart rule `network 924.222.43.0`, the `js2ai` command automatically created the AI criteria file `AI_rootdisk.profile/criteria-3.xml`, specifying a range of IP addresses based on the given network address:

```
<?xml version="1.0" encoding="utf-8"?>
<ai_criteria_manifest>
  <ai_criteria name="ipv4">
    <range>
      924.222.43.0 924.222.43.255
    </range>
  </ai_criteria>
</ai_criteria_manifest>
```

For the JumpStart rule `arch i386`, the `js2ai` command automatically created the AI criteria file `AI_mirrorpool.profile/criteria-4.xml`, replacing the JumpStart `arch` keyword with the AI `cpu` keyword:

```
<?xml version="1.0" encoding="utf-8"?>
<ai_criteria_manifest>
```

```
<ai_criteria name="cpu">
  <value>
    i386
  </value>
</ai_criteria>
</ai_criteria_manifest>
```

## Fixing AI Manifest Files

The js2ai command often creates an AI manifest for each JumpStart profile even though errors are reported. This section describes how to address some common errors so that the output AI manifests are more complete.

### EXAMPLE 4 Fixing fdisk.profile Errors

The js2ai command showed the following errors for the fdisk.profile JumpStart profile:

```
fdisk.profile:line 2:UNSUPPORTED: unsupported value for 'system_type' specified: server
fdisk.profile:line 9:UNSUPPORTED: unsupported keyword: cluster
```

These two lines are ignored and do not affect the output AI manifest. You could delete these two lines if you want the conversion to avoid the error messages. The fdisk.profile file would then have the following content:

```
install_type initial_install
root_device c1t0d0s0
usedisk c1t0d0
fdisk rootdisk solaris all
partitioning explicit
fileys rootdisk.s1 5000 swap
fileys rootdisk.s0 10000 /
```

Use the -p option of the js2ai command to process just this profile.

```
# js2ai -p fdisk.profile
Successfully completed conversion
```

The output AI manifest, AI\_fdisk.profile/fdisk.profile.x86.xml, has the following content in the target stanza:

```
<target>
  <disk whole_disk="true">
    <disk_name name="c1t0d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice name="1" action="create" force="true" is_swap="true">
```



```

        <size val="5000mb"/>
    </slice>
    <slice name="0" action="create" force="true" in_zpool="rpool" in_vdev="
rpool_vdev">
        <size val="10000mb"/>
    </slice>
</partition>
</disk>
<logical noswap="false" nodump="true" >
    <zpool name="rpool" is_root="true">
        <vdev name="rpool_vdev" redundancy="none"/>
        <filesystem name="export" mountpoint="/export"/>
        <filesystem name="export/home"/>
        <be name="solaris"/>
    </zpool>
</logical>
</target>

```

The software stanza is the same as in `/usr/share/auto_install/manifest/default.xml` because this profile does not contain any package specifications.

#### EXAMPLE 5 Fixing mirrorfilesystem.profile Errors

The `js2ai` command showed the following errors for the `mirrorfilesystem.profile` JumpStart profile:

```

mirrorfilesystem.profile:line 3:UNSUPPORTED: unsupported mount point of 'unnamed'
specified,
mount points other than '/' and 'swap' are not supported
mirrorfilesystem.profile:line 4:UNSUPPORTED: unsupported keyword: cluster

```

In a JumpStart profile, you can omit the *file-system* parameter in the `filesystem mirror` specification. In a `js2ai` conversion, you cannot omit the *file-system* parameter, which must have one of the following two values: `/` or `swap`.

Edit the `mirrorfilesystem.profile` file to add `"/` at the end of the `filesystem` specification and to delete the `cluster` line. The `mirrorfilesystem.profile` file then has the following content:

```

install_type initial_install
partitioning default
filesystem mirror c6t0d0s0 c6t1d0s0 60048 /

```

Save the `AI_mirrorfilesystem.profile/criteria-2.xml` file in another location. Then, use the `-p` option of the `js2ai` command to process just this profile.

```
# js2ai -p mirrorfilesystem.profile
```

Successfully completed conversion

The `AI_mirrorfilesystem.profile` directory contains two output AI manifests: `mirrorfilesystem.profile.sparc.xml` and `mirrorfilesystem.profile.x86.xml`. The only difference between these two AI manifests is that `mirrorfilesystem.profile.x86.xml` contains two partition specifications. The `mirrorfilesystem.profile.x86.xml` manifest has the following content in the target stanza:

```
<target>
  <disk>
    <disk_name name="c6t1d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="
0">
        <size val="60048mb"/>
      </slice>
    </partition>
  </disk>
  <disk>
    <disk_name name="c6t0d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="
0">
        <size val="60048mb"/>
      </slice>
    </partition>
  </disk>
  <logical noswap="false" nodump="false" >
    <zpool name="newpool" is_root="true">
      <vdev name="rpool_vdev" redundancy="mirror"/>
      <filesystem name="export" mountpoint="/export"/>
      <filesystem name="export/home">
        <be name="solaris">
      </zpool>
    </logical>
  </target>
```

The software stanza is the same as in `/usr/share/auto_install/manifest/default.xml` because this profile does not contain any package specifications.

#### **EXAMPLE 6** Fixing `mirrorpool.profile` Errors

The `js2ai` command showed the following errors for the `mirrorpool.profile` JumpStart profile:

```
mirrorpool.profile:line 4:CONVERSION: unable to convert 'any' device to physical device.
Replace 'any' with actual device name
```

```
mirrorpool.profile:line 8:UNSUPPORTED: unsupported keyword: cluster
```

Use either of the following two methods to fix these errors. Both of these fixes result in the same AI manifest output. See also [“How the any Keyword Is Translated” on page 24](#).

- Edit the `mirrorpool.profile` profile, replacing both any entries with physical device names. Also, delete the `cluster` line. The `mirrorpool.profile` file will then have content similar to the following example:

```
install_type initial_install
partitioning default
pool newpool auto auto auto mirror c6t0d0s0 c6t1d0s0
```

- If you specify any instead of a physical device name in the `vdev-list` list in the pool specification, you must provide device information prior to the specification that includes the `any` parameter. Edit the `mirrorpool.profile` profile to add a `usedisk` specification before the pool specification. Also, delete the `cluster` line. The `mirrorpool.profile` file will then have content similar to the following example:

```
install_type initial_install
partitioning default
usedisk c6t0d0 c6t1d0
pool newpool auto auto auto mirror any any
```

Save the `AI_mirrorpool.profile/criteria-4.xml` file in another location. Then, use the `-p` option of the `js2ai` command to process just this profile.

```
# js2ai -p mirrorpool.profile
Successfully completed conversion
```

The `AI_mirrorpool.profile` directory contains two output AI manifests: `mirrorpool.profile.sparc.xml` and `mirrorpool.profile.x86.xml`. The only difference between these two AI manifests is that `mirrorpool.profile.x86.xml` contains two partition specifications. The `mirrorpool.profile.x86.xml` manifest has the following content in the target stanza:

```
<target>
  <disk>
    <disk_name name="c6t1d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="newpool" name="
0"/>
    </partition>
  </disk>
  <disk>
    <disk_name name="c6t0d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="newpool" name="
0"/>
```

```
    </partition>
  </disk>
  <logical noswap="false" nodump="true">
    <zpool name="rpool" is_root="true">
      <vdev name="rpool_vdev" redundancy="mirror"/>
      <filesystem name="export" mountpoint="/export"/>
      <filesystem name="export/home">
      <be name="solaris">
    </zpool>
  </logical>
</target>
```

The software stanza is the same as in `/usr/share/auto_install/manifest/default.xml` because this profile does not contain any package specifications.

#### **EXAMPLE 7** Fixing rootdisk.profile Errors

The `js2ai` command showed the following errors for the `rootdisk.profile` JumpStart profile:

```
rootdisk.profile:line 3:CONVERSION: unable to convert 'rootdisk.s0'.
Replace 'rootdisk.' with actual device name
rootdisk.profile:line 4:CONVERSION: unable to convert 'rootdisk.s1'.
Replace 'rootdisk.' with actual device name
rootdisk.profile:line 5:UNSUPPORTED: unsupported keyword: cluster
```

Use either of the following two methods to fix these errors. Both of these fixes result in the same AI manifest output.

- Edit the `rootdisk.profile` profile, replacing `rootdisk.` with `c0t0d0`. Also, delete the `cluster` line. The `rootdisk.profile` file then has the following content:

```
install_type initial_install
partitioning explicit
fileys c0t0d0s0 15000 /
fileys c0t0d0s1 1000 swap
```

- Edit the `rootdisk.profile` profile to add a `root_device` specification and to delete the `cluster` line. The `rootdisk.profile` file then has the following content:

```
install_type initial_install
partitioning explicit
root_device c0t0d0s0
fileys rootdisk.s0 15000 /
fileys rootdisk.s1 1000 swap
```

Save the `AI_rootdisk.profile/criteria-3.xml` file in another location. Then use the `-p` option of the `js2ai` command to process just this profile.

```
# js2ai -p rootdisk.profile
Successfully completed conversion
```

The `AI_rootdisk.profile` directory contains two output AI manifests: `rootdisk.profile.sparc.xml` and `rootdisk.profile.x86.xml`. The only difference between these two AI manifests is that `rootdisk.profile.x86.xml` contains a partition specification. The `rootdisk.profile.x86.xml` manifest has the following content in the target stanza:

```
<target>
  <disk>
    <disk_name name="c0t0d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0">
        <size val="15000mb"/>
      </slice>
      <slice action="create" force="true" is_swap="true" name="1">
        <size val="1000mb"/>
      </slice>
    </partition>
  </disk>
  <logical noswap="false" nodump="true" noswap="false">
    <zpool name="rpool" is_root="true" >
      <vdev name="rpool_vdev" redundancy="none"/>
      <filesystem name="export" mountpoint="/export"/>
      <filesystem name="export/home">
        <be name="solaris">
      </zpool>
    </logical>
  </target>
```

The software stanza is the same as in `/usr/share/auto_install/manifest/default.xml` because this profile does not contain any package specifications.

## Displaying Additional Profile Conversion Information

If you want to see additional information for a rule or profile conversion or AI manifest validation, specify the `-v` option with the `js2ai` command. When you specify the `-v` option, processing steps are displayed, and the error report displays zeroes instead of omitting the rules file or profile from the report.

```
# js2ai -v -p rootdisk.profile
Processing profile: rootdisk.profile
Performing conversion on: rootdisk.profile
```

```
Generating x86 manifest for: rootdisk.profile
Validating rootdisk.profile.x86.xml
Generating sparc manifest for: rootdisk.profile
Validating rootdisk.profile.sparc.xml
```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
-----	-----	-----	-----	-----	-----
rootdisk.profile	0	0	0	0	0

Successfully completed conversion

## Validating an Output AI Manifest

Use the -V option to validate the AI manifest that results from running the js2ai command.

```
# js2ai -V ./AI_rootdisk.profile/rootdisk.profile.sparc.xml
Successfully completed conversion
# js2ai -v -V ./AI_rootdisk.profile/rootdisk.profile.sparc.xml
Validating rootdisk.profile.sparc.xml
```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
-----	-----	-----	-----	-----	-----
rootdisk.profile.sparc	-	-	-	-	0

Successfully completed conversion

## Converting sysidcfg Configuration Files

---

This chapter describes how to convert a `sysidcfg` file to an AI system configuration profile. Much of the conversion can be done with the `js2ai` command. See the [js2ai\(1M\)](#) man page for more information about the `js2ai` command.

AI system configuration profiles are SMF XML profiles that specify system configuration. For more information about AI system configuration profiles, see [Chapter 11, “Defining AI Client System Configuration Parameters”](#) in *Installing Oracle Solaris 11.3 Systems*.

### Comparing sysidcfg Keywords to System Configuration Profile Directives

The following table compares `sysidcfg` file keywords with example AI system configuration profile specifications. Different SMF property value names might apply in different cases. See the examples in [“sysidcfg Conversion Using js2ai”](#) on page 43 and in [“Example System Configuration Profiles”](#) in *Installing Oracle Solaris 11.3 Systems*.

**TABLE 4** Comparing `sysidcfg` Keywords and Configuration Profile Directives

<code>sysidcfg</code> Keyword	System Configuration Profile Directives
keyboard	The <code>js2ai</code> command does not perform any translation. Make sure the keyboard specified in the <code>sysidcfg</code> file is supported in Oracle Solaris 11.  Service name: <code>system/keymap</code> Property group name: <code>keymap</code> Property value name: <code>layout</code>
name_service	AI supports DNS, NIS, and LDAP. The <code>js2ai</code> command supports values <code>NONE</code> , <code>DNS</code> , <code>NIS</code> , and <code>LDAP</code> . <code>NIS + name</code> services are translated as <code>NIS</code> . If a name service is configured, the network must be configured as <code>DefaultFixed</code> .  Service name: <code>system/name-service/switch</code>

sysidcfg Keyword	System Configuration Profile Directives
	<p>Property group name: config</p> <p>Property value name: netgroup</p>
network_interface	<p>AI supports configuring only a single interface as part of system installation. Because of this limitation, the js2ai command processes only the interface labeled PRIMARY or the first interface encountered in the sysidcfg file. The js2ai command sets the network to DefaultFixed if a name_service is specified. A properly configured DefaultFixed network needs to provide the host name, IP address, netmask, and gateway. Automated network configuration is supported only if no name service is specified.</p> <p>Oracle Solaris 11 by default uses the prefix net when assigning link names. (The older style used in Oracle Solaris 10 is available but is disabled by default.) When the js2ai command does a conversion of the network interface and sees an Oracle Solaris 10 style link name, the command will disable neutral link names. To use neutral link names in Oracle Solaris 11, you must change the network interface name specified in the sysidcfg file to an Oracle Solaris neutral link name like net0. For more information about neutral link names, see <a href="#">“Network Devices and Datalink Naming in Oracle Solaris” in Configuring and Managing Network Components in Oracle Solaris 11.3.</a></p>
nfs4_domain	<p>Service name: network/install</p> <p>Property group name: install_ipv4_interface</p> <p>Property value name: static_address</p> <p>Property group name: install_ipv6_interface</p> <p>nfs4_domain=dynamic is supported for Automatic and DefaultFixed networks. networks. nfs4_domain=custom-domain-name is only supported for DefaultFixed networks. The conversion of the network_interface keyword determines whether a DefaultFixed or Automatic network is configured.</p>
root_password	<p>Service name: network/nfs/mapid</p> <p>Property group name: nfs-props</p> <p>Property value name: nfsmapid_domain</p> <p>Oracle Solaris 11 uses roles instead of the root user. You need to define a user with root role privileges in order to access the system in multiuser mode. Because the necessary structure cannot completely be defined using the root_password keyword, you need to add the necessary structures used to define a user account with root role privileges to the sc_profile.xml file as a comment. If you do not specify the root_password keyword, you will also need to define the necessary datastructure for it as a comment.</p>
security_policy	<p>Service name: system/config-user</p> <p>Property group name: root_account</p> <p>Property value name: password</p> <p>The js2ai command supports the value NONE.</p>
service_profile	<p>The js2ai command supports the value limited_net.</p>
system_locale	<p>The js2ai command will check whether the locale specified is one of the default core locales supported and will generate a warning if the locale is not supported. The js2ai command will also attempt to convert non-core locales to core locales.</p> <p>For more information on core locales, see <a href="#">“Internationalization and Localization Changes” in Transitioning From Oracle Solaris 10 to Oracle Solaris 11.3.</a></p>
	<p>Service name: system/environment</p>



sysidcfg Keyword	System Configuration Profile Directives
	Property group name: environment Property value name: LC_ALL
terminal	The js2ai command does not perform any translation. Make sure the terminal type specified in the sysidcfg file is supported in Oracle Solaris 11.
	Service name: system/console-login Property group name: ttymon Property value name: terminal_type
timeserver	The js2ai command supports the value localhost.
timezone	The js2ai command does not perform any translation.
	Service name: system/timezone Property group name: timezone Property value name: localtime

## Using js2ai to Convert sysidcfg Files to System Configuration Profiles

Use the js2ai command with the -s option to convert any sysidcfg files that are associated with this JumpStart configuration to system configuration profiles. Initially, use the -S option to skip validation.

For each sysidcfg file processed, js2ai creates an AI system configuration profile named sc\_profile.xml in the directory where the js2ai command was invoked. Use the -D option to specify a different directory for the sc\_profile.xml file.

When you receive a message that the conversion completed successfully, run the js2ai command without the -S option to validate the output sc\_profile.xml file. Validation errors must be corrected in the sc\_profile.xml file.

If you do not see a message that the conversion was successfully completed, examine the error report and the js2ai.log file. The error report and the log file report warnings, process errors, unsupported items, conversion errors, and validation errors. The error report is a table output to stdout that shows the number of each type of error that was encountered in converting the sysidcfg file. The log file describes the problems.

1. Correct any process errors.
2. Remove any lines from the sysidcfg file that are listed as unsupported items.
3. Examine the conversion errors and correct the errors if possible. Otherwise, remove the lines that are causing the errors.

4. Examine any warning messages and make sure no corrections are necessary.

To display more information for a conversion use the `-v` option with the `js2ai` command. For more information, see [“Displaying Additional Configuration Conversion Information” on page 45](#).

To validate a specific output system configuration profile, run the `js2ai` command with the `-V` option. See the example in [“Validating an Output System Configuration Profile” on page 46](#).

```
# js2ai -V path/sc_profile.xml
```

## js2ai Conversion Warning

All `js2ai` conversions of `sysidcfg` files will generate the following warning:

```
sysidcfg:line #:WARNING: Oracle Solaris 11 uses roles instead of root user. An admin user with root role privileges will need to be defined in order to access the system in multi-user mode. The necessary xml structures have been added to sc_profile.xml as a comment. Edit sc_profile.xml to perform the necessary modifications to define the admin user.
```

The warning relates to the inability of the `js2ai` command to generate the necessary data structure from just the `root_passwd` keyword. If your `sysidcfg` file contained the `root_passwd` keyword in it, `js2ai` would generate a `system/config-user` structure like the following:

```
<service name="system/config-user" type="service" version="1">
  <instance enabled="true" name="default">
    <!--
    Configures user account as follows:
    * User account name 'jack'
    * password 'jack'
    * GID 10
    * UID 101
    * root role
    * bash shell
    -->
    <!--
  <property_group name="user_account" type="application">
    <propval name="login" type="astring" value="jack"/>
    <propval name="password" type="astring" value="9Nd/cwBcNWFZg"/>
    <propval name="description" type="astring" value="default_user"/>
    <propval name="shell" type="astring" value="/usr/bin/bash"/>
    <propval name="gid" type="astring" value="10"/>
    <propval name="uid" type="astring" value="101"/>
```

```

<propval name="type" type="astring" value="normal"/>
<propval name="roles" type="astring" value="root"/>
<propval name="profiles" type="astring" value="System Administrator"/>
<propval name="sudoers" type="astring" value="ALL=(ALL) ALL"/>
</property_group>
-->
  <property_group name="root_account" type="application">
    <propval name="login" type="astring" value="root"/>
    <propval name="password" type="astring" value="{your_root_passwd}"/>
    <propval name="type" type="astring" value="role"/>
  </property_group>
</instance>
</service>

```

In XML, a comment is started with `<!--`. To define the user necessary to support Oracle Solaris 11, remove the XML comment markers around the `user_account` property group structure. Next, modify the `user_account` property structure to create the user that will have root role privileges. If no `root_password` keyword was specified, comment out the `root_account` property group structure.

## sysidcfg Conversion Using js2ai

This section shows how to use `js2ai` to convert a `sysidcfg` file to an AI system configuration profile. For each `sysidcfg` file processed, `js2ai` creates an AI system configuration profile named `sc_profile.xml` in the directory where the `js2ai` command was invoked. Use the `-D` option to specify a different directory for the `sc_profile.xml` file. To aid in the conversion process, make a copy of the original `sysidcfg` file and work from a copy. The conversion process might require that you edit and modify the `sysidcfg` file before it can be successfully converted.

### EXAMPLE 8 Converting a sysidcfg File That Includes Errors

This example uses the following `sysidcfg` file:

```

timezone=US/Pacific
timeserver=localhost
keyboard=US-English
system_locale=en_US.UTF-8
terminal=vt100
network_interface=primary { hostname=host1
  ip_address=192.0.2.4
  netmask=255.255.255.224
  protocol_ipv6=yes
  default_route=192.0.2.3 }

```

```

name_service=DNS{domain_name=example.com
    name_server=192.0.2.1,192.0.2.2
    search=example.com,example.org}
root_password=rJmv5LUXM10cU
security_policy=none
nfs_domain=dynamic

```

Use the following command to process this sysidcfg file. In the error report, validation errors are shown as a hyphen character because the -S option suppressed validation.

```

# js2ai -sS

```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
sysidcfg	1	0	0	1	-

Conversion completed. One or more failures and/or warnings occurred.  
For errors see js2ai.log

```
# cat js2ai.log
```

```

sysidcfg:line 6:CONVERSION: DefaultFixed network configuration enabled.
Unable to complete network configuration, replace interface PRIMARY
with the actual interface you wish to configure.
sysidcfg:line 17:WARNING: Oracle Solaris 11 uses roles instead of root user.
An admin user with root role privileges will need to be defined in order to
access the system in multi-user mode. The necessary xml structures have been
added to sc_profile.xml as a comment. Edit sc_profile.xml to perform the
necessary modifications to define the admin user.

```

#### EXAMPLE 9 Converting a Corrected sysidcfg File

The following modified sysidcfg file addresses the errors reported in the previous example. The PRIMARY interface specification is replaced with the interface name e1000g.

```

timezone=US/Pacific
timeserver=localhost
keyboard=US-English
system_locale=en_US.UTF-8
terminal=vt100
network_interface=e1000g { hostname=host1
    ip_address=192.0.2.4
    netmask=255.255.255.224
    protocol_ipv6=yes
    default_route=192.0.2.3 }
name_service=DNS{domain_name=example.com
    name_server=192.0.2.1,192.0.2.2
    search=example.com,example.org}

```

```

root_password=rJmv5LUXM10cU
security_policy=none
nfs_domain=dynamic

```

```
# js2ai -sS
```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
sysidcfg	2	0	0	0	-

Conversion completed. One or more failures and/or warnings occurred.

For details see /tmp/js2ai.log

```
# cat js2ai.log
```

```

sysidcfg:line 16:WARNING: Oracle Solaris 11 uses roles instead of root user.
An admin user with root role privileges will need to be defined in order to
access the system in multi-user mode. The necessary xml structures have been
added to sc_profile.xml as a comment. Edit sc_profile.xml to perform the
necessary modifications to define the admin user.

```

```

sysidcfg:line 6:WARNING: In order to support the direct translation of the
sysidcfg interface 'e1000g', Oracle Solaris 11 neutral link name support will
be disabled. If you wish to use neutral link names, change the interface
name specified in the sysidcfg file to a 'netx' style interface name or edit
the resulting sc_profile.xml file.

```

Although the network interface primary was replaced with an Oracle Solaris 10 interface name, you will still get a warning about neutral link names. If you want to use the traditional link names then no action is required. In most instances, the interfaces will map the same as they do in Oracle Solaris 10 but that outcome is not guaranteed.

If you want to use the neutral link names used by Oracle Solaris 11, you can change your network interface in the `sysidcfg` file to a `netn` style name. Typically the primary interface will be defined as `net0`. In order to address the user warning, edit the `sc_profile.xml` file accordingly as outlined in the `js2ai.log` file.

## Displaying Additional Configuration Conversion Information

If you want to see additional information for a `sysidcfg` file conversion or system configuration profile validation, specify the `-v` option with the `js2ai` command. When you specify the `-v` option, processing steps are displayed.

```

# js2ai -sv
Processing: sysidcfg
Performing conversion on: sysidcfg

```

```
Generating SC Profile
Validating sc_profile.xml
```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
sysidcfg	2	0	0	0	0

```
Conversion completed. One or more failures and/or warnings occurred.
For details see /tmp/js2ai.log
```

## Validating an Output System Configuration Profile

Use the `-V` option to validate the system configuration profile that results from running the `js2ai` command.

```
# js2ai -v -V ./sc_profile.xml
Validating sc_profile.xml
```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
sc_profile	-	-	-	-	0

```
Successfully completed conversion
```

## Installing Oracle Solaris 10 by Using JumpStart on an Oracle Solaris 11 Server

---

You can use JumpStart to install the Oracle Solaris 10 operating system on networked SPARC and x86 platforms, but not to install the Oracle Solaris 11 OS. However, the JumpStart install server can be an Oracle Solaris 11 system.

Your Oracle Solaris 11 server can do two different jobs:

- Serve Oracle Solaris 11 OS installations using Automated Installer. For more information, see [Part 3, “Installing Using an Install Server,” in \*Installing Oracle Solaris 11.3 Systems\*](#).
- Serve Oracle Solaris 10 OS installations using JumpStart. This chapter describes how to set up a JumpStart install server on an Oracle Solaris 11 system. For more information about JumpStart, see [Oracle Solaris 10 1/13 Installation Guide: Live Upgrade and Upgrade Planning](#).

### Setting Up an Oracle Solaris 11 System as an Oracle Solaris 10 JumpStart Server

The following procedure shows how to create an Oracle Solaris 11 JumpStart Install server to install the Oracle Solaris 10 OS on client systems.

#### ▼ How to Set Up an Oracle Solaris 11 System as an Oracle Solaris 10 JumpStart Server

1. Install the Oracle Solaris 11 OS.
2. Set up a static IP address.

For information about how to set a static IP address, see the instructions in “[How to Configure an IPv4 Interface](#)” in *Configuring and Managing Network Components in Oracle Solaris 11.3* and the `ipadm(1M)` man page.

### 3. Install the `system/boot/network` package from the solaris publisher.

```
# pkg publisher
PUBLISHER          TYPE    STATUS  URI
solaris            origin  online  http://pkg.oracle.com/solaris/release/
# pkg install pkg:/system/boot/network
    Packages to install: 1
    Variants/Facets to change: 3
    Create boot environment: No
    Create backup boot environment: No
    Services to change: 1

DOWNLOAD                                PKGS    FILES    XFER (MB)
Completed                                1/1     13/13     0.0/0.0

PHASE                                ACTIONS
Install Phase                          34/34

PHASE                                ITEMS
Package State Update Phase              1/1
Image State Update Phase                 2/2

PHASE                                ITEMS
Reading Existing Index                   8/8
Indexing Packages                        1/1
```

### 4. Download the latest Oracle Solaris 10 DVD image.

The current image is at <https://www.oracle.com/technetwork/server-storage/solaris10/downloads/index.html>. This image is the source of your install image and JumpStart tools. Note that JumpStart can be used to install only the Oracle Solaris 10 OS, not the Oracle Solaris 11 OS.

### 5. Set up a JumpStart install server.

The following example uses the Oracle Solaris 10 1/13 DVD image for SPARC.

```
# cd /media/SOL_10_0113_SPARC/Solaris_10/Tools
# ./setup_install_server /export/s10u11_sparc
Verifying target directory...
Calculating the required disk space for the Solaris_10 product
Calculating space required for the installation boot image
Copying the CD image to disk...
Copying Install Boot Image hierarchy...
```



```
Copying /boot netboot hierarchy...
Install Server setup complete
```

**6. Copy the jumpstart\_sample files to the JumpStart profile location.**

```
# cd /media/SOL_10_0113_SPARC/Solaris_10/Misc/jumpstart_sample
# mkdir -p /export/profiles/s10profile
# cp -pr * /export/profiles/s10profile
```

**7. Update the check script.**

Replace the first line of the /export/profile/s10profile/check script with the following line:

```
#!/usr/sunos/bin/sh
```

**8. Validate your JumpStart profile.**

In the following example, many lines of output are omitted for brevity. Note that the -p option does not work with all Oracle Solaris 10 update releases. Make sure you are using at least Oracle Solaris 10 Update 7.

```
# cd /export/profile/s10profile
# ./check -p /export/s10u11_sparc
45 blocks
Validating rules...
Validating profile host_class...
Validating profile zfsrootsimple...
Validating profile net924_sun4c...
Validating profile upgrade...
Validating profile x86-class...
Validating profile any_machine...
The custom JumpStart configuration is ok.
```

**9. Make sure that udp6 is available and online.**

The udp6 service is delivered in the tftp package. Make sure the tftp package is installed.

```
# pkg list tftp
```

If the tftp package is not installed, use the following command to install it:

```
# pkg install pkg:/service/network/tftp
```

Once you have installed the tftp package, make sure the udp6 service is available and online.

```
# svcs udp6
STATE          STIME   FMRI
disabled      8:36:55 svc:/network/tftp/udp6:default
# svcadm enable network/tftp/udp6
```

```
# svcs udp6
STATE          STIME   FMRI
online         8:38:28 svc:/network/tftp/udp6:default
```

**10. If you plan to use this server as both a JumpStart install server and an AI install server, link the /tftpboot directory to the /etc/netboot directory.**

If the /etc/netboot directory does not exist, create it.

Although JumpStart uses /tftpboot for network booting, AI uses /etc/netboot. Therefore, /tftpboot needs to be a symbolic link to /etc/netboot so that the two tools can coexist.

---

**Tip** - If you plan to use this server as only a JumpStart install server or only an AI install server, then this step is not required. However, doing this step avoids failed netboot errors if you change your plan later.

---

```
# ls /etc/netboot
/etc/netboot: No such file or directory
# mkdir -m 755 /etc/netboot
# ln -s /etc/netboot /tftpboot
# ls -l /tftpboot
lrwxrwxrwx  1 root      root           12 Sep 14 8:46:51 /tftpboot -> /etc/netboot
```

**11. Install the client.**

Make sure to use the fully qualified host name for the client. In this example, the client host name is line2-x4100.example.com.

```
/export/s10u11_sparc/Solaris_10/Tools/add_install_client -e 8:0:20:fd:f2:18 \
-c line2-x4100.example.com:/export/profiles/s10profile \
-p line2-x4100.example.com:/export/profiles/s10profile line2-t1 sun4u
/export/home is already shared. However, the zfs file system
/export/home must be shared read-only with root access. Use the "zfs
set" command to set the sharenfs property for file system /export/home
as follows:Use ro and either anon=0 or
root=line2-x4100.example.com for /export/home. This must be
fixed and /export/home shared before line2-x4100.example.com can boot.
.
.
```

Use the following commands to verify how the /export/home file system is shared, correct the share options if needed, and then check how the file system is shared to verify the change. In this example, the anon=0 option has not been set.

```
# share
home /export/home nfs sec=sys,ro
# share -F nfs -o ro,sec=sys,anon=0 /export/home
# share
```

```
home    /export/home    nfs    anon=0,sec=sys,ro
```



# Index

---

## A

AI *See* Automated Installer (AI)

AI criteria

- compared to JumpStart keywords, 15
- description of, 15

AI manifests

- converting JumpStart files to, 15
- definition of, 15
- displaying validation information, 37
- elements compared to JumpStart keywords, 17
- specifying with AI criteria, 15

AI profiles

- converting JumpStart files to, 15

any JumpStart rules file keyword, 15

any keyword

- js2ai command process, 24

arch AI criteria keyword, 16

arch JumpStart rules file keyword, 16

archive\_location JumpStart profile keyword, 17

Automated Installer (AI)

- comparison with JumpStart, 11

## B

backup\_media JumpStart profile keyword, 17

boot\_device JumpStart profile keyword

- described, 17

- eeprom keyword and, 17

bootenv JumpStart profile keyword, 18

## C

client\_arch JumpStart profile keyword, 18

client\_root JumpStart profile keyword, 18

client\_swap JumpStart profile keyword, 18

cluster JumpStart profile keyword, 18

comparing

- JumpStart and AI installation tasks, 11

console-login service

- system configuration profile, 41

converting

- JumpStart rules file and profiles, 15

cpu AI criteria keyword, 16

criteria directives *See* AI criteria

## D

-D destination option

- js2ai command, 41

device translation

- any keyword, 24

differences

- JumpStart and AI, 11

directives *See* AI criteria *See* AI manifest elements

disksize JumpStart rules file keyword, 16

displaying profile conversion information, 37

domainname JumpStart rules file keyword, 16

dontuse JumpStart profile keyword, 18

## E

eeprom JumpStart profile keyword

- boot\_device keyword and, 17

environment service

- system configuration profile, 40

**F**

fdisk JumpStart profile keyword, 18  
filesys JumpStart profile keyword, 18  
forced\_deployment JumpStart profile keyword, 20

**G**

geo JumpStart profile keyword, 20

**H**

hostaddress JumpStart rules file keyword, 16  
hostname JumpStart rules file keyword  
  described, 16  
  replacing, 30

**I**

install service  
  system configuration profile, 40  
install\_type JumpStart profile keyword, 20  
installed JumpStart rules file keyword, 16  
installing  
  Oracle Solaris 10 from an Oracle Solaris 11  
  server, 47  
IPS package names, 25  
ipv4 AI criteria keyword, 16, 16

**J**

js2ai command  
  conversion warning message, 42  
  converting JumpStart files, 22  
  -D destination option, 41  
  determining the root disk, 23  
  determining the ZFS root pool, 24  
  package installation and, 25  
  -S skip option, 41  
  -s sysidcfg option, 41  
  translating the any keyword, 24

  using to convert sysidcfg files to system  
  configuration profiles, 41  
  -v validate option, 38, 45, 46  
  -v verbose option, 37

**JumpStart**

  comparison with AI, 11  
  using on an Oracle Solaris 11 server, 47

**JumpStart profiles**

  converting to AI files, 15  
  JumpStart keywords compared to AI manifest  
  elements, 17  
  unsupported keywords in AI, 17  
  using js2ai to convert to AI criteria and  
  manifests, 22

**JumpStart rules file**

  converting to AI files, 15  
  keywords compared to AI criteria, 15  
  unsupported keywords in AI, 15  
  using js2ai command to convert to AI criteria and  
  manifests, 22

JumpStart to AI migration overview, 12

**K**

karch JumpStart rules file keyword, 16  
keyboard keyword  
  sysidcfg file, 39  
keymap service  
  system configuration profile, 39

**L**

layout\_constraint JumpStart profile keyword, 20  
local\_customization JumpStart profile keyword, 20  
locale JumpStart profile keyword, 20

**M**

mac AI criteria keyword, 16  
manifest directives *See* AI manifest elements  
manifests *See* AI manifests

mem AI criteria keyword, 16  
memsize JumpStart rules file keyword, 16  
metadb JumpStart profile keyword, 20  
model JumpStart rules file keyword, 16

## N

name-config-user service  
    system configuration profile, 40  
name-nfs/mapid service  
    system configuration profile, 40  
name-service/switch service  
    system configuration profile, 39  
name\_service keyword  
    sysidcfg file, 39  
network AI criteria keyword, 16  
network JumpStart rules file keyword, 16  
network\_interface keyword  
    sysidcfg file, 40  
nfs4\_domain keyword  
    sysidcfg file, 40  
no\_content\_check JumpStart profile keyword, 20  
no\_master\_check JumpStart profile keyword, 20  
num\_clients JumpStart profile keyword, 20

## O

osname JumpStart rules file keyword, 17

## P

package installation  
    js2ai command and, 25  
package JumpStart profile keyword, 20  
package names  
    IPS and SVR4, 25  
partitioning JumpStart profile keyword, 21  
patch JumpStart profile keyword, 21  
platform AI criteria keyword, 16  
pool JumpStart profile keyword, 21

probe JumpStart rules file keyword, 17

## R

replacing hostname keyword, 30  
root disk  
    determining, 23  
root\_device JumpStart profile keyword, 21  
root\_password keyword  
    sysidcfg file, 40

## S

-S skip option  
    js2ai command, 41  
security\_policy keyword  
    sysidcfg file, 40  
selection criteria *See* AI criteria  
service\_profile keyword  
    sysidcfg file, 40  
similarities  
    JumpStart and AI, 11  
SMF services  
    in system configuration profile, 39  
SVR4 package names, 25  
sysidcfg files  
    comparing keywords to system configuration profile  
    directives, 39  
    converting to an AI system configuration  
    profile, 39  
    js2ai command process for converting, 41  
system\_locale keyword  
    sysidcfg file, 40  
system\_type JumpStart profile keyword, 22

## T

terminal keyword  
    sysidcfg file, 41  
timeserver keyword  
    sysidcfg file, 41

timezone keyword  
    sysidcfg file, 41  
timezone service  
    system configuration profile, 41  
totaldisk JumpStart rules file keyword, 17

## U

usedisk JumpStart profile keyword, 22

## V

-v option  
    js2ai command, 45  
-V validate option  
    js2ai command, 38, 46  
-v verbose option  
    js2ai command, 37

## W

warning message  
    from js2ai command, 42

## Z

ZFS root pool  
    js2ai command process for determining, 24