Creating a Custom Oracle® Solaris 11.4 Image

Part No: E60981
November 2020

Creating a Custom Oracle Solaris 11.4 Image

Part No: E60981

License Restrictions Warranty/Consequential Damages Disclaimer

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.

Warranty Disclaimer

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.

Restricted Rights Notice

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 embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

Hazardous Applications Notice

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.

Trademark Notice

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

Intel and Intel Inside 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, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Third-Party Content, Products, and Services Disclaimer

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.

Pre-General Availability Draft Label and Publication Date

Pre-General Availability: 2020-01-15

Pre-General Availability Draft Documentation Notice

If this document is in public or private pre-General Availability status:

This documentation is in pre-General Availability status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

Oracle Confidential Label

ORACLE CONFIDENTIAL. For authorized use only. Do not distribute to third parties.

Revenue Recognition Notice

If this document is in private pre-General Availability status:

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your pre-General Availability trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

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: E60981

Restrictions de licence/Avis d'exclusion de responsabilité en cas de dommage indirect et/ou consécutif

Exonération de garantie

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.

Avis sur la limitation des droits

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 embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

Avis sur les applications dangereuses

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 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.

Marques

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 Inside 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, Epyc, et le logo AMD sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Avis d'exclusion de responsabilité concernant les services, produits et contenu tiers

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.

Date de publication et mention de la version préliminaire de Disponibilité Générale ("Pre-GA")

Version préliminaire de Disponibilité Générale ("Pre-GA") : 15.01.2020

Avis sur la version préliminaire de Disponibilité Générale ("Pre-GA") de la documentation

Si ce document est fourni dans la Version préliminaire de Disponibilité Générale ("Pre-GA") à caractère public ou privé :

Cette documentation est fournie dans la Version préliminaire de Disponibilité Générale ("Pre-GA") et uniquement à des fins de démonstration et d'usage à titre préliminaire de la version finale. Celle-ci n'est pas toujours spécifique du matériel informatique sur lequel vous utilisez ce logiciel. Oracle Corporation et ses affiliés déclinent expressément toute responsabilité ou garantie expresse quant au contenu de cette documentation. Oracle Corporation et ses affiliés ne sauraient en aucun cas être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'utilisation de cette documentation.

Mention sur les informations confidentielles Oracle

INFORMATIONS CONFIDENTIELLES ORACLE. Destinées uniquement à un usage autorisé. Ne pas distribuer à des tiers.

Avis sur la reconnaissance du revenu

Si ce document est fourni dans la Version préliminaire de Disponibilité Générale ("Pre-GA") à caractère privé :

Les informations contenues dans ce document sont fournies à titre informatif uniquement et doivent être prises en compte en votre qualité de membre du customer advisory board ou conformément à votre contrat d'essai de Version préliminaire de Disponibilité Générale ("Pre-GA") uniquement. Ce document ne constitue en aucun cas un engagement à fournir des composants, du code ou des fonctionnalités et ne doit pas être retenu comme base d'une quelconque décision d'achat. Le développement, la commercialisation et la mise à disposition des fonctions ou fonctionnalités décrites restent à la seule discrétion d'Oracle.

Accessibilité de la documentation

Pour plus d'informations sur l'engagement d'Oracle pour l'accessibilité de la documentation, visitez le site Web Oracle Accessibility Program, à l'adresse : http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

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

- **Overview** – Describes how to build custom Oracle Solaris installation packages using the distribution constructor tool
- **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=E37838-01`.

Feedback

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

♦ ♦ ♦ CHAPTER 1

Introduction to Creating a Custom Installation Image

System administrators and application developers can use the distribution constructor tool to build custom Oracle® Solaris installation images. This chapter covers the following topics:

- "About the Distribution Constructor"
- "Image Creation Process"
- "About Rights Profiles in Oracle Solaris"

About the Distribution Constructor

The distribution constructor (DC) is a command-line tool for building preconfigured Oracle Solaris images. The tool can build an ISO image as well as a USB image that is based on a generated ISO image. The USB image can work in various types of flash memory devices that have driver support provided by Oracle Solaris.

Depending on the image configuration, ISO or USB images can be bootable. An ISO image can be burned to a CD or DVD while a USB image can be copied to a flash drive.

To copy the USB image to a USB flash drive, you use either the `usbcopy` command, which is available in the `distribution-constructor` package, or the `dd` command.

You can create the following types of Oracle Solaris images:

- Oracle Solaris x86 or SPARC Text Installation Image – For use when installing Oracle Solaris with the text installer.

  See *Manually Installing an Oracle Solaris 11.4 System*.
- x86 or SPARC ISO Image for Automated Installations – For use when installing Oracle Solaris with automated installer (AI).

  See *Automatically Installing Oracle Solaris 11.4 Systems*.

Image Creation Process

The distribution constructor creates images based on settings specified in DC manifests. DC manifests are XML files that contain specifications for the images you want to create. The distribution constructor package includes sample manifests to help you build the images.

All the fields in each DC manifest file provide preset default values that will create the type of image you need. You can edit fields in the manifest file to further customize the resulting image.

You can also create *custom scripts* to modify your installation image. Then, you add checkpoints to the manifest to run these custom scripts. Checkpoints enable you to stop and start the image building process for purposes of debugging. At the end of the process, a simple log file and a detailed log file are generated.

About Rights Profiles in Oracle Solaris

Oracle Solaris implements role-based access control (RBAC) to control system access. To perform specific tasks and run privileged commands on the system, you must have the profiles that provide you the authorization.

The following list shows some of the profiles that need to be assigned to you to work on manifests and building and installing these images.

- Install Client Management enables you to install Oracle Solaris on client systems.
- Install Manifest Management enables you to create or configure manifests to customize the installation.
- Install Profile Management enables you to create and configure system configuration profiles to customize the installation.

Some profiles are supersets of a combination of profiles. For example, the Install Service Management profile contains the three profiles in the previous list.

The list of required profiles expands if you perform additional tasks that might be indirectly connected to your current one, such as network configuration or zone configuration.

An administrator that has the `solaris.delegate.*` authorization can assign the necessary profiles to users to enable them to perform administrative tasks in Oracle Solaris.

For example, an administrator assigns the Install Service Management rights profile to user `jdoe`. Before `jdoe` executes a privileged installation command, `jdoe` must be in a profile shell.

The shell can be created by issuing the `pfbash` command. Or, `jdoe` can combine `pfexec` with every privilege command that is issued, such as `pfexec installadm`.

As an alternative, instead of assigning profiles directly to users, a system administrator can create a role that would contain a combination of required profiles to perform a range of tasks.

Suppose that a role `installadmin` is created with the profiles for installation as well as for zone creation and configuration. User `jdoe` can issue the `su` command to assume that role. All roles automatically get `pfbash` as the default shell.

For more information about rights profiles, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

♦ ♦ ♦ CHAPTER 2

Design a Custom Installation Image

This chapter provides system requirements for building images and describes how to design a custom installation image by creating a DC manifest and scripts. It includes the following topics:

- "Requirements for Building Images"
- "About Sample DC Manifest Files"
- "Modifying the Manifest Content"
- "Creating and Using Custom Scripts"

Note - To create and install custom images, and to run commands documented here, ensure that you have the correct profiles. See "About Rights Profiles in Oracle Solaris" on page 12.

Requirements for Building Images

To use the distribution constructor, your system must meet the following requirements.

- 8 Gbytes of disk space for the distribution constructor workspace.
- Access to Image Packaging System (IPS) repositories on the network.
- Platform that matches the type of image you want to create. For example, to create SPARC images, you must use a SPARC system.
- OS release that corresponds to the image you want to create. For example, to create Oracle Solaris 11.4 images, the system must be running that OS version.
- `distribution-constructor` package installed on the system.

  This package is automatically included when you install Oracle Solaris 11.4.

About Sample DC Manifest Files

Depending on your system's platform, the `distribution-constructor` package includes the following sample manifest files that are stored in `/usr/share/distro_const`.

- On SPARC systems:
  - `dc_text_sparc.xml` – SPARC text installation image
  - `dc_ai_sparc.xml` – SPARC AI ISO image
- On x86 systems
  - `dc_text_x86.xml` – x86 text installation image
  - `dc_ai_x86.xml` – x86 AI ISO image

These manifests use the following environment variables:

- `PKG_IMAGE_PATH` - during the installation image build process this variable is resolved to the path to the package image.
- `BOOT_ARCHIVE` - during the installation image build process this variable is resolved to the path to the installation image. For instance the path to the `/etc/system` file in the archive would be `{BOOT_ARCHIVE}/etc/system`.

All the fields in each DC manifest file contain default values for the type of ISO image you need. You can manually edit these preset fields or replicate elements to further customize the image.

DC manifests contain the following primary elements:

- `distro` – name of the image.
- `boot_mods` – editable boot menu options.
- `target` – ZFS dataset that will hold the installation image when it is created.
- `software` – publisher for both the installation image and the install client, as well as the packages to be installed or uninstalled.
- `execution` – checkpoints for actions to be performed during the image building process. You can also add your own checkpoints.

For detailed information, see the `dc_manifest(5)` man page.

Modifying the Manifest Content

Most of the default values in the sample manifests would work on any environment. However, some sections might require specific configurations appropriate to your setup.

In certain sections of the manifests, elements are defined twice to provide you with alternative attribute configurations. The alternative definitions are inside comment marks. To use the alternative option, you remove the comment marks and set the correct values to the attributes. See the example in "Providing the Image Title" on page 17.

Note - Do not edit the original sample file directly. Instead, as best practice, make a copy of the file you want to use. Revise its contents and then specify it with the command to build the image.

Tip - To facilitate understanding the next sections, have a copy of a sample manifest open to serve as reference.

Providing the Image Title

The `distro name` element enables you to provide a custom name for the image. For example:

```
<distro name="Oracle_Solaris_Text_X86" add_timestamp="false">
<!-- Uncomment if http_proxy needs to be set
  <distro name="Oracle_Solaris_Text_X86" add_timestamp="false"
      http_proxy="http://example.com">
-->
```

The `add_timestamp` attribute determines whether a time stamp is appended to the image name. It enables you to retain a series of builds of the same image.

A second `distro name` definition inside comment marks enables you to specify an HTTP proxy. To use, remove the comment marks and then provide the proxy location. Then, remove the first option.

In Oracle Solaris 11.4, long image names are truncated to 32 characters at the end of the process.

Modifying the Boot Menu

The `boot_mods` element specifies boot menu modifications associated with the image.

In the following example, a specialized boot menu with the title `boot1` is applied to the image. Based on `timeout`, the default boot entry is automatically activated within 5 seconds.

```
<boot_mods title="boot1" timeout="5">
```

You can add individual boot menu entries by using `boot_entry` sub-elements. For boot entry attributes, `title_suffix` is required, while the following are optional:

- `default_entry` – sets the boot entry to be the default entry. This attribute is normally set to `false`.
- `insert_at` – determines the entry's position on the menu list: at the beginning if set to `start` or at the end if set to `end`. By default, a new entry is added at the end of the list.

The following example shows a `boot_entry` configuration. The entry will be inserted at the end of the menu list. Also, the entry is not designated to be the default entry.

The *arguments* variable refers to kernel settings that are passed to the kernel by the boot loader.

```
<boot_entry default_entry="false" insert_at="end">
   <title_suffix>title</title_suffix>
   <kernel_args>arguments</kernel_args>
</boot_entry>
```

Specifying the Dataset to be Used

The `target` element defines the ZFS build dataset where the image is created. You must provide a valid dataset location. Check the default build area to make sure that the build does not destroy content that you need to keep on your system. The path to the dataset is relative to the pool name defined. In the following example, the dataset would be installed as `rpool/dc/sample-dataset-location`.

Note - The file system name should not include the name of the zpool.

```
<target>
  <logical>
    <zpool action="use_existing" name="rpool">
      <dataset>
        <filesystem name="dc/sample-dataset-location"
         action="preserve"/>
      </dataset>
    </zpool>
  </logical>
</target>
```

Specifying Software Sources and Packages

The software section of the manifest contains software node names under which elements and attributes enable you to define additional settings.

Software and package parameters are defined under the following heading:

```
<software name="transfer-ips-install" type="IPS">
```

To specify publisher information such as name, origin, and mirror settings, look for the `source` element. You can specify multiple publishers, as shown in the example.

The `source` element is followed by `software_data` where you add the names of packages to form the installed image. You can add multiple packages as needed.

By default, the most current package version available in the specified repository is installed. If you want to install a different version, append the version number to the package reference.

The following example shows how publishers and packages are defined in the manifest:

```
<source>    for specifying publishers
   <publisher name="publisher1">
      <origin name="http://example.oracle.com/primary-pub"/>
      <mirror name="mirror.example.com"/>
   </publisher>
   <publisher name="publisher2">
       <origin name="http://example2.com/dev/solaris"/>
   </publisher>
   <publisher name="publisher3">
       <origin name="file:///net/myserver/publisher4/repo"/>
   </publisher>
</source>

<software_data action="install">    for specifying packages
   <name>pkg:/group/system/solaris-large-server</name>
   <name>pkg:/group/system/solaris-desktop@0.5.11-0.build#</name>
</software_data>
```

To specify the default publisher to be set on the system after it has been installed, configure the elements under the following heading:

```
<software name="set-ips-attributes" type="IPS">
```

Here you can add not only the name of the default publisher, but also any mirror information. Note that other than publisher information, you would rarely configure other elements under this heading.

Setting Up Build Checkpoints

The `execution` element lists a series of checkpoints for the image construction process. Checkpoints identify moments in the building process where the tool performs actions as defined in corresponding software node names in the manifest. For example, at the `transfer-ips-install` checkpoint, the tool processes all the definitions under `<software name=" transfer-ips-install" type="IPS">`.

Checkpoints are executed in the order in which they are listed under the `execution` element. During the image construction process, the checkpoints modify the contents of the build area that is specified in the manifest.

The build area consists of a `pkg_image` directory and a `boot_archive` directory. The contents of the final image are added to the `pkg_image` directory. The files in the separate `boot_archive` directory are used during the build process to create a boot archive file, which is also then added to the `pkg_image` directory.

By default, the following checkpoinst are predefined in the sample manifests:

```
transfer-ips-install
set-ips-attributes
pre-pkg-img-mod
ba-init
ba-config
ba-arch
boot-setup
pkg-img-mod
create-iso
create-usb
checksums
```

Each checkpoint element includes the `mod-path` attribute that specifies where the checkpoint script is located.

The following checkpoint example from the `dc_ai_sparc.xml` sample manifest creates the boot archive for the image build and points to a script that will build the image. It also includes argument fields with specific values provided for each argument.

```
<checkpoint name="ba-arch"
    desc="Boot Archive Archival"
    mod_path="solaris_install/distro_const/checkpoints/boot_archive_archive"
    checkpoint_class="BootArchiveArchive">
    <kwargs>
        <arg name="size_pad">0</arg>
        <arg name="bytes_per_inode">0</arg>
        <arglist name="uncompressed_files">
             <argitem>etc/svc/repository.db</argitem>
             <argitem>etc/name_to_major</argitem>
             <argitem>etc/minor_perm</argitem>
             <argitem>etc/driver_aliases</argitem>
             <argitem>etc/driver_classes</argitem>
             <argitem>etc/path_to_inst</argitem>
             <argitem>etc/default/init</argitem>
             <argitem>etc/nsswitch.conf</argitem>
             <argitem>etc/passwd</argitem>
             <argitem>etc/shadow</argitem>
             <argitem>etc/inet/hosts</argitem>
        </arglist>
    </kwargs>
</checkpoint>
```

As shown in this example, the `kwargs` element contains keyword arguments that need to be passed into the checkpoint during the build. Within the `kwargs` element are `arg name` elements that can be used to specify individual keywords to be passed into the checkpoint. The `arglist` element contains a list of multiple `argitem` values to be passed into the checkpoint. This example includes a list of uncompressed files in the `arglist` element.

Each `kwargs` list item is enclosed in double quotes. If no double quotes are used or if one set of double quotes encloses the entire string, the entire string including spaces and new lines is interpreted as one argument. Do not use commas between arguments.

If you create a custom script to be used during the building of an image, you must add a checkpoint element pointing to the script location. The checkpoint for a custom script needs only an `args` element that points to the custom script location. For further information and examples, see "Creating and Using Custom Scripts" on page 23.

EXAMPLE 1 Adding SVR4 Packages to An Installation Image

In this example, a new checkpoint is added to the manifest. This new checkpoint lists SVR4 packages to be added to the image and their location. This new checkpoint then is referenced in the execution section.

First, the new checkpoint is created by adding a new `software` element. This checkpoint specifies SVR4 as the software type, where to find the packages, and where to install the packages.

In addition, the specific SVR4 packages to be installed are listed in the `software_data` element.

```
<software name="transfer-svr4-install" type="SVR4">
   <destination>
       <dir path="{PKG_IMAGE_PATH}"/>
   </destination>
   <source>
    <publisher/>
      <origin name="path-to-packages"/>
    </publisher>
   </source>
   <software_data action="install">
      <name>SUNWpackage1</name>
      <name>SUNWpackage2</name>
   </software_data>
</software>
```

If included in the checkpoint, the values of `{PKG_IMAGE_PATH}` and `{BOOT_ARCHIVE}` are replaced by the `distro_const` command with the path to the build area for the package image and the boot archive, respectively. In this example, the SVR4 packages will be installed into the package image directory.

Finally, the new checkpoint is referenced in the execution section. The checkpoint name can be any string, but for this example, the `checkpoint_class` value must be `TransferSVR4`.

```
<execution stop_on_error="true">
   <checkpoint name="transfer-ips-install"
       desc="Transfer pkg contents from IPS"
       mod_path="solaris_install/transfer/ips"
       checkpoint_class="TransferIPS"/>
   <checkpoint name="set-ips-attributes"
       desc="Set post-install IPS attributes"
       mod_path="solaris_install/transfer/ips"
       checkpoint_class="TransferIPS"/>
   <checkpoint name="transfer-svr4-install"
      desc="Transfer pkg contents from SVR4 packages"
      mod_path="solaris_install/transfer/svr4"
      checkpoint_class="TransferSVR4"/>
```

Note that the software name must match the checkpoint name. In this example, both are “transfer-svr4–install.”

EXAMPLE 2 Creating Hashes of the Media in an Installation Image

The `checksums` checkpoint enables users to automatically generate hashes of the media generated by the `distro_const` command.

```
    <checkpoint name="checksums"
      desc="Checksum calculation for media"
      mod_path="solaris_install/distro_const/checkpoints/checksums"
      checkpoint_class="Checksums">
      <kwargs>
        <arglist name="algorithms">
          <argitem file_path="/tmp/md5sums.txt">md5</argitem>
        <argitem>sha1</argitem>
        <argitem>sha224</argitem>
        <argitem>sha256</argitem>
        <argitem>sha384</argitem>
        <argitem>sha512</argitem>
      </arglist>
    </kwargs>
  </checkpoint>
```

The `arglist` element includes all of the algorithms that are used to generate hashes for the generated media. Each `argitem` specifies an algorithm. The valid algorithms can be determined by running the `/usr/bindigest -l` command. Each `argitem` can have a `path` attribute that specifies the absolute path of an additional file that will be appended with the hashes produced by that algorithm. If no algorithms are specified, the default is `md5`.

While the image is built, files will be generated for each algorithm containing checksums for each media.

Creating and Using Custom Scripts

You can add scripts to further customize the image creation process. These scripts are referenced in the execution section of the manifest files. You can specify any number of `custom-script` checkpoints.

Often custom scripts are used to modify a configuration file or make some other changes that can not be done using a manifest.

Scripts specified in the `execution` section of the manifest file are run during the image creation process. The execution section does not reference pre-install or post-install scripts.

Note - Do not change scripts that are installed from packages. To prevent problems with future package updates, make any changes in a script you create.

When you create your own custom scripts, note the following:

- Scripts can be Python programs, shell scripts, or binaries.
- Scripts are executed in the order in which they are listed in the execution section of the manifest file.
- Standard output (`stdout`) and error output (`stderr`) of commands that are executed within the scripts (both shell and Python modules) are captured in log files that report on the completed or attempted build.

▼ How to Create and Use a Custom Script to Build an Installation Image

1. **Create a new script.**

2. **Add the new script to your home directory or elsewhere on the system or network.**

   Make sure that a user assuming the `root` role can execute the script.

3. **Modify the manifest.**

   Add information referencing the new script in the `execution` section of the manifest. To decide where to add the new checkpoint, review "Setting Up Build Checkpoints" on page 20.

   Be sure to specify the full path to your scripts. Checkpoints are executed in the order in which they are listed in the `execution` section of the manifest.

   When you add a reference for a new script in the `execution` section of a manifest file, you must specify a checkpoint name that can be used to pause the image build before or after the script performs its task. Optionally, you can include a custom message associated with the checkpoint name. If this message is omitted, the path of the script is used as the default checkpoint message. The checkpoint message displays when the checkpoint is run during the build process.

   Note - Use meaningful names for checkpoint names rather than ordinal numbers. If you use numbers, adding new checkpoints for new scripts will disrupt the numbered checkpoint order.

   The following example checkpoint references a custom script named `my-script`.

```
<checkpoint name="my-script"
        desc="my new script"
        mod_path="solaris_install/distro_const/checkpoints/custom_script"
        checkpoint_class="CustomScript">
        <args>/tmp/myscript.sh</args>
</checkpoint>
```

4. **Build the image.**

   You can build the image in one step or stop and restart the build at various checkpoints to check the status of the build.

   For instructions, see Chapter 3, "Building an Image".

5. **(Optional) After the build is complete, view the log file for the build process.**

   The build output displays the location of the log files.

Example 3 Using Environment Variables in a Checkpoint

In the following example, the image directory path is used as an argument to `myscript.sh`.

```
<checkpoint name="my-script"
        desc="my new script"
        mod_path="solaris_install/distro_const/checkpoints/custom_script"
        checkpoint_class="CustomScript">
        <args>/tmp/myscript.sh {PKG_IMAGE_PATH}</args>
</checkpoint>
```

Example 4 Including a Short Script in a Custom DC Manifest

The following script will set the password for the `solaris` user which can be used to access an install client during the installation process.

```
     <checkpoint name="set-ips-attributes"
         desc="Set post-install IPS attributes"
         mod_path="solaris_install/transfer/ips"
         checkpoint_class="TransferIPS"/>
     </checkpoint>
     <checkpoint name="solaris-password"
         desc="Set the password for the solaris account used during the installation
 process"
         mod_path="solaris_install/distro_const/checkpoints/custom_script"
         checkpoint_class="CustomScript">
         <args>sed 's/solaris:.[^:]*:/solaris:string:/g' {PKG_IMAGE_PATH}/etc/shadow
> {PKG_IMAGE_PATH}/etc/shadow.new; cp {PKG_IMAGE_PATH}/etc/shadow.new
{PKG_IMAGE_PATH}/etc/shadow; rm {PKG_IMAGE_PATH}/etc/shadow.new</args>
     </checkpoint>
     <checkpoint name="pre-pkg-img-mod"
```

♦ ♦ ♦ CHAPTER 3

Building an Image

After you have set up the manifest file that you plan to use and, if desired, customized the finalizer scripts, you are ready to build an image by running the `distro_const` command.

This chapter covers the following topics:

- "About the `distro_const` Command"
- "Building an Installation Image"

About the `distro_const` Command

To build an image, use the following command syntax:

```
$ distro_const build options manifest
```

The command takes the following options:

- `-v` – executes the process in verbose mode.
- `-l` – lists all valid checkpoints defined in the manifest.
- `-p` *checkpoint* – pauses building process at a specified checkpoint.
- `-r` *checkpoint* – resumes building process from a specified checkpoint.
- `-h` – displays the Help section.

Building an Installation Image

You can build an image with or without using checkpoints. With checkpoints, the build process is completed in stages. This section describes the two ways you can build an image.

▼ How to Build an Image

1. **Perform one of the following steps:**

   - **To build the image without checkpoints, issue the following command:**

     ```
     $ distro_const build manifest.xml
     ```

     If necessary, include the path when specifying the manifest.

   - **To build the image in stages:**

     a. **Display the checkpoints defined in the manifest.**

        For example:

        ```
        $ distro_const build -l manifest.xml
        Checkpoint            Resumable Description
        --------------        --------- -------------
        transfer-ips-install          X Transfer package contents from IPS
        set-ips-attributes            X Set post-installation IPS attributes
        pre-pkg-img-mod               X Pre-package image modification
        ba-init                       X Boot archive initialization
        ba-config                     X Boot archive configuration
        ba-arch                       X Boot archive archiving
        pkg-img-mod                     Package image area modifications
        create-iso                      ISO image creation
        create-usb                      USB image creation
        ```

        In this sample command output, an `X` in the resumable field indicates that you can restart the build from this checkpoint.

     b. **Build the image while pausing at the specified checkpoint.**

        The following command starts building an image and pauses the build before `ba-arch` modifies the image area:

        ```
        $ distro_const build -p ba-arch manifest.xml
        ```

     c. **Resume building the image from a specified checkpoint.**

        You can only resume from the checkpoint at which the build process stopped or from an earlier checkpoint. You cannot skip to resume the build at a later checkpoint.

        The following command resumes building the image at the `ba-arch` stage.

        ```
        $ distro_const build -r ba-arch manifest.xml
        ```

Note - You can combine the pause and resume options in a `build` command.

2. **(Optional) View the log file for the build process.**

   The build output displays the location of log files.

Index