**Using Unified Archives for System Recovery and Cloning in Oracle® Solaris 11.4**

ORACLE®

Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4

**Part No: E60984**

Copyright © 2014, 2020, Oracle and/or its affiliates.

**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.

Ce document contient des informations qui sont la propriété exclusive d'Oracle, qu'il s'agisse de la version électronique ou imprimée. Votre accès à ce contenu confidentiel et son utilisation sont soumis aux termes de vos contrats, Contrat-Cadre Oracle (OMA), Contrat de Licence et de Services Oracle (OLSA), Contrat Réseau Partenaires Oracle (OPN), contrat de distribution Oracle ou de tout autre contrat de licence en vigueur que vous avez signé et que vous vous engagez à respecter. Ce document et son contenu ne peuvent en aucun cas être communiqués, copiés, reproduits ou distribués à une personne extérieure à Oracle sans le consentement écrit d'Oracle. Ce document ne fait pas partie de votre contrat de licence. Par ailleurs, il ne peut être intégré à aucun accord contractuel avec Oracle ou ses filiales ou ses affiliés.

**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 the creation and deployment of Oracle Solaris Unified Archives to perform system recovery and cloning operations on Oracle Solaris systems
- **Audience** – System administrators and others responsible for system recovery and cloning of Oracle Solaris systems
- **Required knowledge** – Experience administering an Oracle Solaris system

## Product Documentation Library

Documentation and resources for this product and related products are available at `https://docs.oracle.com/cd/E37838_01/`.

## Feedback

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

# 1

♦♦♦ **C H A P T E R  1**

# Oracle Solaris System Recovery and Cloning (Overview)

This chapter presents an overview of the Unified Archives feature and covers the following topics:

- "What's New with Unified Archives in Oracle Solaris 11.4"
- "About Oracle Solaris Unified Archives"
- "Creating and Deploying Unified Archives"
- "Using Rights Profiles with Unified Archives"
- "Best Practices Using Unified Archives"

## What's New with Unified Archives in Oracle Solaris 11.4

The following features are added in this Oracle Solaris release:

- Support for dehydrated clones enables you to create clones with reduced file sizes for easier transport. Reduced archive sizes help ISVs in archive distribution, transfer, or backup. Unified Archives can become unencumbered from copyright and distribution rights for the OS. See "Dehydrated Clone Archives" on page 13.
- Additional `archiveadm create` command options enable you to monitor the progress of the archiving process. See the command output in Example 1, "Creating a Clone Archive That Includes All Zones," on page 21. Additional options and improvements enable you to view more information about archives than previously available. See the different examples of the use of the `archiveinfo` command in this document as well as in the man page.

# About Oracle Solaris Unified Archives

The Unified Archives feature is a tool for creating archives of physical or virtual systems to be used for system recovery and cloning operations. Virtual systems can be global zones, non-global zones, or kernel zones. Zones can be archived individually, bundled together, or selectively archived as a subset of the total zones. These multiple systems or instances are archived in a single unified format.

When you select zones to archive, by default all datasets associated with those zones are included. However, you can exclude any number of datasets from the archive.

A non-global zone archive contains the zone boot environment (BE) datasets and all delegated datasets assigned to it. A global zone archive, in addition, would also contain all datasets not associated with a zone. Entire top-down pools are also included.

---

**Note -** Unified Archives contain only ZFS datasets (file systems, volumes, and snapshots). All other file system and volume types are excluded.

Likewise, any `fs` resources in a zone are also excluded from the zone's archived data. However, an exemption exists in the case where the `fs` resource is defined as follows:

```
type=lofs
special=path-to-a-ZFS-file-system
```

Because the `special` property refers to a directory for a ZFS file system, then an archive of the global zone would include the data in that directory.

---

With Unified Archives, you can perform the following:

- Recover a system that needs to be replaced due to failure.
- Duplicate or clone a system configuration that you want to install on multiple systems.
- Migrate an existing system to new hardware or to a virtual machine.

## Unified Archive Types

An archive can either be a *clone archive* or a *recovery archive*.

## Clone Archives

A clone archive is the default archive type. It is based on the system's active BE when the archive is created. Therefore, only one BE can be archived at a time. You use a clone archive type for rapid creation and deployment of custom images.

If the clone archive contains both the global and non-global zones, the data for each archived system is stored independently and clone relationships are not preserved. Thus, each archived system is independently deployable. A cloned zone in a clone archive uses almost the same amount of space as its original zone.

A clone archive does not include any of the OS instance's system configuration information nor any sensitive data such as SSH keys or passwords. Any system deployed from a clone archive will have all of the file systems and applications as the cloned system, but not the same host name, for instance. System configuration information and sensitive data are preserved only in recovery archives.

## Dehydrated Clone Archives

You can create a dehydrated clone archive, where information from one or more publishers is removed, and then deploy it. See Example 5, "Creating a Clone Archive without Content from Any Publishers," on page 23 and Example 6, "Creating a Partially Dehydrated Clone Archive," on page 23.

If the archive contains more than one deployable system, the publisher information is removed from all of the systems. If one of the systems does not include a named publisher, archive creation fails. You would need to create separate archives of those systems with their corresponding publishers.

You can rehydrate a dehydrated clone archive and then deploy it. Whether installing a dehydrated or rehydrated archive, you must use the same publisher with the same content version. Deploying a dehydrated clone archive takes longer due to the time to download information from the publishers.

**Note -** Like dehydration, the rehydration process updates information for one or more publishers for all of the deployable systems in the archive.

## Recovery Archives

A recovery archive is a full-system archive containing all BEs from all included instances. It consists of a single deployable system regardless of the combination of zones it contains. Unlike a clone archive, a recovery archive preserves zone clone relationships.

In a recovery archive, the active boot environment is the only fully prepared BE. Inactive BEs are archived mainly for data recovery regardless of their being bootable.

A recovery archive is intended for recovery operations, as in the case of hardware failure. Ideally, the archive should be deployed as part of a more comprehensive disaster recovery solution.

# Platform Portability of Unified Archives

Unified Archives are compatible across hardware platform models of the same instruction set architecture (ISA). The archives are also portable to virtual machines on the same ISA. For example, an archive of an Oracle SPARC T5 system is deployable on the more modern Oracle SPARC M7 line of systems. In addition, such an archive is deployable on a logical domain or a Solaris kernel zone that runs on the same ISA.

Archive portability enables you to perform image transforms between physical and virtual machines as well as between global and non-global zones. Because these transforms are committed during deployment, the transform is determined by the deployed system type.

Provided that system ISAs are compatible, Unified Archives offers migration paths for legacy systems. For example, you can archive legacy hardware installations and deploy them as virtual machine instances on new hardware. Similarly, you can archive non-global Oracle Solaris zones and deploy them as Oracle Solaris kernel zones.

Unified Archives cannot be used in the following migration cases:

- Oracle Solaris 10 branded zones cannot be transformed into any other system type. They may only be archived and deployed as an identical Oracle Solaris 10 branded zone instance.
- Trusted Extensions or labeled zone archives can not be transformed. They can be deployed only to a global zone context, such as a Logical Domain, to Oracle Solaris kernel zones, or to bare metal machines.
- Archives of labeled non-global zone archives are only deployable as 'labeled'-brand zones, running within the context of a Trusted Extensions global zone. See "Planning Your Labeled Zones in Trusted Extensions" in *Trusted Extensions Configuration and Administration*.

# Using Rights Profiles with Unified Archives

Oracle Solaris implements role-based access control (RBAC) to control system access. To create and deploy Unified Archives, you must be assigned at a minimum the Unified Archive Administration profile.

Other profiles are required if you perform additional tasks beyond working with unified archives. For example, to install Oracle Solaris from a unified archive, you would also need the following profiles:

- Install Manifest Management for creating and managing install manifests.
- Install Profile Management for creating and managing install service profiles.
- Install Client Management for creating and managing install services.

An administrator that has the `solaris.delegate.*` authorization can assign the required profiles to users.

For example, a system administrator assigns the Unified Archive Administration profile to user `jdoe`. Before `jdoe` executes a privileged command related to unified archives, `jdoe` must be in a profile shell. The shell can be created by issuing the `pfbash` command. Or, `jdoe` can combine `pfexec` with every privileged command that is issued, for example, `pfexec archiveadm`.

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

Suppose that a role `uadeploy` is created with the Unified Archive Administration profile as well as the profiles required for installation. As an authorized user, `jdoe` uses 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*.

# Creating and Deploying Unified Archives

With Unified Archives, you can perform the following tasks:

- Create an archive to be used for system recovery or system cloning.
- Create bootable media from an archive to enable archive deployment.
- Retrieve information related to an existing archive, such as archive creation time, architecture, and system content.

For detailed procedures, see Chapter 2, "Working With Unified Archives".

Three modes of archive deployment are available:

- Non-interactive network deployment using the Automated Installer (AI)

You would need to modify the AI manifest to include the location of the archive and the name of the target systems the archive would be deployed. See "How to Deploy a System From a Unified Archive by Using AI" on page 33.

- Zones deployment using the zones commands

  After deployment, the new zone mimics the original system's configuration. For specific instructions, see "How to Deploy a Zone From a Unified Archive" on page 30.

- Interactive deployment using bootable media containing either ISO or USB images

  Use this method if the install client is not networked or if no AI services are available. The media images deploy the contents of the archive non-interactively when the install client is booted. See "How to Deploy a System From a Unified Archive by Using Bootable Media" on page 34.

**Note -** The first and third methods support the installation of logical domains from a Unified Archive.

Any installation method that uses AI profiles, including zone installations, can use Unified Archives.

A `sysconfig` profile overrides any system configuration information in an archive except some SMF configuration data. The SMF data is then implemented during the deployment together with the profile contents. See see "Repository Layers" in *Managing System Services in Oracle Solaris 11.4*.

Where no `sysconfig` profile is used, some archive deployments require input through an interactive system configuration screen. See Chapter 3, "Working With System Configuration Profiles" in *Customizing Automated Installations With Manifests and Profiles*.

# Best Practices Using Unified Archives

Consider the following recommendations when using Unified Archives:

- Archive components are consistent with one another. When an archive is deployed, all data present post-deployment should be from the same point in time from the origin system's perspective. Consistency is guaranteed on a per-pool basis. If an Oracle Solaris instance spans multiple pools, as when an application or data pool is used separately from a root pool, quiesce any active workloads prior to archive creation. This step ensures consistency between the pools once they are deployed.

- The `archiveadm create -D` syntax enables you to exclude datasets not needed for redeployment. Remove temporary or unnecessary directories and data prior to archive creation. Or, place these in their own datasets so you can exclude them using the `-D` option.

- When creating the archive, specify an informative name. For example `noscratch-recovery.uar` is a recovery file without the scratch dataset. However, `archiveadm info` also displays informatioin about the archive contents.
- To take advantage of IPS functionality during archive creation, convert SVR4 packages to IPS packages and install the packages onto a ZFS dataset. The `archiveadm` command only works on data that is managed by ZFS. For more information, see "Converting SVR4 Packages To IPS Packages" in *Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.4*.

# 2

# Working With Unified Archives

This chapter discusses different ways of using Unified Archives. It covers the following topics:

- "Creating a Unified Archive"
- "Viewing Unified Archive Information"
- "Deploying Unified Archives"

## Creating a Unified Archive

To create an archive, you issue the `archiveadm create` command on the system you want to archive. The following options are useful when creating archives.

- `-D` to exclude specified datasets
- `-e` to exclude bootable media
- `-r` to create a recovery archive
- `-z` to include specified zones
- `-Z` to exclude specified zones
- `--dehydrate` to create a dehydrated clone archive.

  The `archiveadm rehydrate` command rehydrates a dehydrated clone archive.

- `--root-only` to create an archive that excludes all non-root pool data.

For a full list of all of the options see the `archiveadm`(8) man page.

## About Reverted Files in an Archive

During the archiving process, the contents of several files and directories are automatically reverted to their package defaults. Their contents are specific to the system and should not be propagated to other systems. Thus their information is removed when you create an archive.

Revert tag attributes identify these files:

- `system:clone` identifies files that are excluded in clone archives.
- `system:dev-init` identifies files that are excluded in either clone or recovery archives.

If you need to restore the reverted files after archive deployment, create copies of these files. You restore them manually or create an IPS package that includes those files. Additionally for AI deployments, you can create a first-boot script.

For more information, refer to the following sources:

- *Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.4*
- Chapter 6, "Running a Custom Script During First Boot" in *Customizing Automated Installations With Manifests and Profiles*

To identify reverted files for a clone archive, type the following:

```
$ pkg contents -o path,revert-tag -a revert-tag=system:dev-init \
  -a revert-tag=system:clone
```

To identify reverted files for a recovery archive, type the following:

```
$ pkg contents -o path,revert-tag -a revert-tag=system:dev-init
```

# About Unified Archives and Zones

Unified Archives can be used to archive zones in a system except for two cases that are unsupported.

- Zones that use shared storage
- Embedded zones in kernel zones

The process fails if you attempt to archive a system that contains either zones on shared storage or kernel zones that host other zones. You would need to create separate archives for each of those zones. See "How to Archive a System That Contains Zones on Shared Storage" on page 36 for instructions.

# Examples: Creating Clone Archives

This section provides examples of different ways to create clone archives. Some examples include how to display archive information.

> **Note -** The examples assume that the administrator has the correct privileges to perform these actions. See "Using Rights Profiles with Unified Archives" on page 14.

**EXAMPLE 1**     Creating a Clone Archive That Includes All Zones

This example shows how to archive the entire contents of a system into production1.uar.

```
$ archiveadm create /var/tmp/production1.uar

Logging to /system/volatile/archive_log.11248

  0% : Beginning archive creation.
  5% : Initializing Unified Archive creation resources...
  5% : Unified Archive initialized: /var/tmp/production1.uar
  6% : Executing dataset discovery...
 10% : Dataset discovery complete
 11% : Executing staging capacity check...
 12% : Staging capacity check complete
 15% : Creating zone media: UnifiedArchive [xxx]
 52% : CreateZoneMedia: UnifiedArchive [xxx] complete
 55% : Preparing archive image...
 73% : Archive image preparation complete
 75% : Beginning archive stream creation...
 92% : Archive stream creation complete
 93% : Beginning archive descriptor creation...
 94% : Archive descriptor creation complete
 95% : Beginning final archive assembly...
100% : Archive assembly complete

$ archiveadm info /var/tmp/production1.uar
Archive Information
          Creation Time:  2016-02-02T20:37:16Z
            Source Host:  example
           Architecture:  sparc
```

```
        Operating System:  Oracle Solaris 11.3 SPARC
  Dehydrated Publishers:
     Deployable Systems:  global,example-net,example-dev
```

**EXAMPLE 2**      Creating a Clone Archive of Selected Zones Only

This example shows how to create a clone archive of two zones, example-net and example-dev, into /var/tmp/zonearchive.uar.

```
$ archiveadm create -z example-net,example-dev /var/tmp/zonearchive.uar

Logging to /system/volatile/archive_log.10236

  0% : Beginning archive creation
 .
100% : Archive assembly complete

$ archiveadm info /var/tmp/zonearchive.uar
Archive Information
          Creation Time:  2016-03-02T09:37:10Z
            Source Host:  example
           Architecture:  sparc
       Operating System:  Oracle Solaris 11.3 SPARC
  Dehydrated Publishers:
     Deployable Systems:  example-net,example-dev
```

**EXAMPLE 3**      Creating a Clone Archive That Excludes ZFS Datasets

This example demonstrates how to exclude a ZFS dataset from a clone archive. Datasets must be identified by a path that is accessible from the system where the archiveadm command is executed.

Excluded datasets are not propagated to kernel zones which are being archived. If you need to exclude a dataset in a kernel zone, you must create the archive from the kernel zone.

```
$ archiveadm create -D tank/scratch /var/tmp/noscratch.uar

Logging to /system/volatile/archive_log.11248

  0% : Beginning archive creation
 .
100% : Archive creation complete
```

**EXAMPLE 4**     Creating a Clone Archive Without Bootable Media

By default, bootable media is created for each OS release in an archive. This media is embedded in the archive and used to boot and install kernel zones for each global zone in the archive. This process uses the `pkg://solaris/install-image/solaris-auto-install` package. If the package is not available, or if you want to make smaller archives, create a clone archive without bootable media by using the -e option.

```
$ archiveadm create -e /var/tmp/production2.uar

Logging to /system/volatile/archive_log.3093

  0% : Beginning archive creation...
 .
100% : Archive assembly complete
```

**EXAMPLE 5**     Creating a Clone Archive without Content from Any Publishers

This example shows how to create a dehydrated clone archive where information from any publishers is removed.

```
$ archiveadm create --dehydrate /var/tmp/production3.uar

Logging to /system/volatile/archive_log.808

  0% : Beginning media creation...
 .
100% : Archive assembly complete

$ archiveadm info /var/tmp/production3.uar
Archive Information
          Creation Time:  2016-10-03T23:45:40Z
            Source Host:  example
           Architecture:  sparc
       Operating System:  Oracle Solaris 11.3 SPARC
  Dehydrated Publishers:  database, solaris     Excluded publishers
      Deployable Systems:  global,example-net,example-dev
```

**EXAMPLE 6**     Creating a Partially Dehydrated Clone Archive

Instead of removing all publisher information, use the --publisher option to specify which publisher to remove, and thus create a partially dehydrated archive. Add each publisher in a comma-separated list.

This example removes only the `solaris` publisher from the archive.

```
$ archiveadm create --dehydrate --publisher solaris /var/tmp/production4.uar

Logging to /system/volatile/archive_log.808

  0% : Beginning media creation...
 .
100% : Archive assembly complete
```

**EXAMPLE 7**     Rehydrating a Clone Archive

Rehydrating an archive effectively means re-creating a dehydrated archive but with all of the removed publisher information restored. At this time, selective rehydration is not supported. You must supply the rehydrated archive with a new name.

```
$ archiveadm rehydrate production4.uar production4-rehydrate.uar

Logging to /system/volatile/archive_log.5282

  0% : Beginning rehydration...
 .
100% : Archive assembly complete
```

# Examples: Creating Recovery Archives

This section provides examples of different ways to create recovery archives:

- Example 8, "Creating a Recovery Archive of the Global Zone," on page 24
- Example 9, "Creating A Recovery Archive of a Zone," on page 25
- Example 10, "Creating a Recovery Archive That Excludes a ZFS Dataset," on page 25

**EXAMPLE 8**     Creating a Recovery Archive of the Global Zone

This example shows how to create a recovery archive comprising only the global zone.

```
$ archiveadm create -r -z global /var/tmp/globalrecovery.uar

Logging to /system/volatile/archive_log.7230

  0% : Beginning archive creation...
 .
100% : Archive assembly complete
```

```
# archiveadm info -v globalrecovery.uar
Archive Information
         Creation Time:  2016-02-22T22:51:43Z
           Source Host:  example
          Architecture:  sparc
      Operating System:  Oracle Solaris 11.3 SPARC
  Dehydrated Publishers:
      Recovery Archive:  Yes
           ...


Deployable Systems:
         'global'
           ...
```

**EXAMPLE 9**     Creating A Recovery Archive of a Zone

This example shows how to create a recovery archive of the zone example-net. Note that only one zone is allowed for creation of a recovery archive.

```
$ archiveadm create -r -z example-net /var/tmp/example-net-recovery.uar

Logging to /system/volatile/archive_log.6587

  0% : Beginning archive creation...
 .
100% : Archive assembly complete
$ archiveadm info -v example-net-recovery.uar
Archive Information
         Creation Time:  2016-02-23T23:51:27Z
           Source Host:  example
          Architecture:  sparc
      Operating System:  Oracle Solaris 11.3 SPARC
      Recovery Archive:  Yes
           ...


Deployable Systems:
     'example-net'
           ...
```

**EXAMPLE 10**     Creating a Recovery Archive That Excludes a ZFS Dataset

As with clone archives, you can exclude datasets from an recovery archive. This example shows how to create a recovery archive excluding the ZFS dataset rpool/scratch. The exclusion of a dataset is recursive, so all descendant datasets are also excluded.

```
$ archiveadm create -r -D rpool/scratch /var/tmp/noscratch-recovery.uar

Logging to /system/volatile/archive_log.6587

  0% : Beginning archive creation...
 .
100% : Archive creation complete
```

# Viewing Unified Archive Information

This section provides more examples of displaying archive information.

## Viewing Archive-Related Properties

The following list defines the native archive properties for a selected archive that may be selected with the `archiveadm info -o` command. These properties correspond to the fields displayed as archive information by the `archiveadm info -v` command.

- `archive_uuid` – displays the 128-bit unique identifier of the selected archive.
- `archive_version` – displays the version of the selected archive.
- `creation_time` – displays the creation timestamp (UTC) for the selected archive.
- `dehydrated_publishers` – displays the publishers, if any, of a dehydrated archive.
- `isa` – displays the architecture for the selected archive.
- `os_name` – displays information about the OS in the selected archive.
- `recovery` – displays whether the selected archive is a recovery archive.
- `source_host` – displays the hostname of the system where the archive was created.
- `systems` – displays a list of the deployable systems included in the selected archive.

The following list defines the native system properties for a deployable system in an archive. These properties correspond to the fields displayed for each deployable system by the `archiveadm info -v` command.

- `active_be` – displays the name of the active BE in the selected system.
- `ai_media` – displays the name of the AI media associated with the selected system.
- `brand` – displays brand of the zone of the deployable system.
- `name` – displays the name of the deployable system.
- `os_branch` – displays the OS branch of the `osnet` incorporation.

- os_version – displays the OS version.
- root_only – displays whether the selected system contains only root data.
- size – displays the estimated minimum amount of disk space needed to deploy the system. Does not include dump or swap space or other system specific configurations.
- system_uuid – displays the 128-bit unique identifier of the deployable system.

**Note -** You cannot view native archive properties and native system properties with a single command.

# Examples: Displaying Information With the archiveadm info Command

**EXAMPLE 11**     Viewing Standard Information About an Archive

This example shows how to display standard archive information.

```
$ archiveadm info production1.uar
Archive Information
         Creation Time:  2017-07-31T15:32:43Z
           Source Host:  example
          Architecture:  i386
      Operating System:  Oracle Solaris 11.4 X86
     Deployable Systems:  global
```

**EXAMPLE 12**     Viewing All Information About an Archive

Use the -v option to display full archive information.

```
$ archiveadm info -v production.uar
Archive Information
         Creation Time:  2017-07-31T15:32:43Z
           Source Host:  example
          Architecture:  i386
      Operating System:  Oracle Solaris 11.4 X86
   Dehydrated Publishers:
       Recovery Archive:  No
              Unique ID:  8ce62a0d-d98d-45ea-bbc8-c6c9df7dcdba
         Host OS Version:  11.4
          Host OS Branch:  11.4.0.0.0.2.38282
         Archive Version:  1.0

Deployable Systems
```

```
             'global'
                OS Version:  11.4
                 OS Branch:  11.4.0.0.0.2.38282
                 Active BE:  solaris
                     Brand:  solaris
                     Zones:
            Installed Size:  7.6GB
                 Unique ID:  b0db3abd-374c-4d17-9c1a-dbc726516ef0
                  AI Media:  11.4.0.0.0.2.38282_ai_i386.iso
                 Root-only:  Yes
```

**EXAMPLE  13**     Viewing Storage and Dataset Configuration Information from the Original System

Use the -t option to display the storage and dataset configuration information from the system
that was archived. The output can serve as a basis to create a modified AI manifest that would
match a new target system. If you are deploying a zone from the archive, the information helps
you to know how to prepare the zone on the target system. Suppose that the zone configuration
on the original system had the following dataset configuration.

```
$ zonecfg -z zone2 info
...
dataset:
    name: datapool3/mydeleg
    alias: mydelegated
```

That information is preserved when you create the archive. Assume that the zone is archived in
production.uar. The information is displayed as follows:

```
$ archiveadm info -t production.uar
<target name="origin">
  <disk in_zpool="rpool" in_vdev="rpool-none" whole_disk="true">
    <disk_prop dev_size="3400mb"/>
  </disk>
  <disk in_zpool="mydelegated" in_vdev="mydelegated-none" whole_disk="true">
    <disk_prop dev_size="200mb"/>
  </disk>
  <logical noswap="false" nodump="false">
    <zpool name="rpool" action="create" is_root="true" is_boot="false" mountpoint="/
rpool">
      <vdev name="rpool-none" redundancy="none"/>
    </zpool>
    <zpool name="mydelegated" action="create" is_root="false" is_boot="false"
 mountpoint="/mydelegated">
      <vdev name="mydelegated-none" redundancy="none"/>
    </zpool>
  </logical>
</target>
```

You can then modify the AI manifest that you intend to use to deploy the zone into a new system.

**EXAMPLE   14**     Viewing Specific Target Information in Archives With Multiple Deployable Systems

If the archive contains multiple deployable systems, you must use the -s option together with the -t for the archiveadm info to display the desired target information.

Suppose that clone.uar contains the global zone, example-net, and example-dev as deployable systems:

```
$ archiveadm info clone.uar
Archive Information
          Creation Time:  2016-02-02T20:37:16Z
            Source Host:  example
           Architecture:  sparc
       Operating System:  Oracle Solaris 11.4 SPARC
     Deployable Systems:  global,example-net,example-dev
```

To display target system information, you must specify the specific system:

```
$ archiveadm info -t -s example-net clone.uar
<target name="origin">
  <disk in_zpool="rpool" in_vdev="rpool-none" whole_disk="true">
    <disk_prop dev_size="4400mb"/>
  </disk>
  <disk in_zpool="datapool1" in_vdev="datapool1-none" whole_disk="true">
    <disk_prop dev_size="1300mb"/>
  </disk>
  <logical noswap="false" nodump="false">
    <zpool name="rpool" action="create" is_root="true" is_boot="false" mountpoint="/
rpool">
      <vdev name="rpool-none" redundancy="none"/>
    </zpool>
    <zpool name="datapool1" action="create" is_root="false" is_boot="false" mountpoint="
/datapool1">
      <vdev name="datapool1-none" redundancy="none"/>
    </zpool>
  </logical>
</target>
```

**EXAMPLE   15**     Displaying Specific Archive Property Information

The -o *property* option enables you to directly display specific properties instead of the full data that the -v option generates. For example, instead of displaying the full information to know when the archive was created, you can simply type:

```
$ archiveadm info -o creation_time production1.uar
```

Or, to check if an archive is a clone or recovery archive, you can type:

```
$ archiveadm info -o recovery production1.uar
```

For the complete list of property names that you can use with the -o option, refer to the archiveadm(8) man page.

# Deploying Unified Archives

This section describes procedures for deploying archives. When you deploy an archive, you are essentially performing an installation. For more information especially about automated installation, refer to *Automatically Installing Oracle Solaris 11.4 Systems*.

## ▼ How to Deploy a Zone From a Unified Archive

Zone deployment has two phases: zone creation using the zonecfg command and zone installation using the zoneadm install command. You can use Unified Archives on either operation or on both operations.

For more information about zones, see *Creating and Using Oracle Solaris Zones*.

**Before You Begin**    You must be an administrator with the correct rights profiles to perform this task. See "Using Rights Profiles with Unified Archives" on page 14.

1.  **Create the zone from an existing archive.**

    ```
    $ zonecfg -z new-zone create -a archive [-z archived-zone]
    ```

    If the archive contains multiple deployable systems, the second -z option specifies the zone in the archive that you want to use.

2.  **Install the zone.**

    ■  **To install the zone by using the same archive, type:**

    ```
    $ zoneadm -z new-zone install -a archive [-z archived-zone]
    ```

    If the archive contains multiple deployable systems, the second -z option specifes the zone in the archive that you want to use.

■ **To install the zone by using the IPS repository, type:**

```
$ zoneadm -z new-zone install
```

3. **Boot the zone.**

```
$ zoneadm -z new-zone boot
```

**Example 16** Configuring and Deploying a Zone

This example creates a clone of example-net named test-net. The archive only has one deployable system. Thus, a second -z option to define the zone to clone is not required.

```
$ archiveadm info /var/tmp/example-net-recovery.uar
Archive Information
         Creation Time:  2016-02-23T23:51:27Z
           Source Host:  example
          Architecture:  sparc
      Operating System:  Oracle Solaris 11.3 SPARC
       Recovery Archive:  Yes
     Deployable Systems:  example-net

$ zonecfg -z test-net create -a /var/tmp/example-net-recovery.uar
$ zoneadm list -cv
  ID NAME            STATUS     PATH                        BRAND      IP
   0 global          running    /                           solaris    shared
   - example-net     installed  /zones/example-net          solaris    excl
   - example-dev     installed  /zones/example-dev          solaris    excl
   - test-net        configured /zones/test-net             solaris    excl

$ zoneadm -z test-net install -a /var/tmp/example-net-recovery.uar
The following ZFS file system(s) have been created:
    rpool/VARSHARE/zones/test-net
Progress being logged to /var/log/zones/zoneadm.20160224T001725Z.test-net.install
      Image: Preparing at /system/zones/test-net/root.

 Install Log: /system/volatile/install.7980/install_log
 AI Manifest: /tmp/manifest.test-net.X7SKlb.xml
  SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
    Zonename: example-dev
Installation: Starting ...
  .
        Done: Installation completed in 777.186 seconds.
  Next Steps: Boot the zone, then log into the zone console (zlogin -C)
              to complete the configuration process.
Log saved in non-global zone as /system/zones/test-net/root/var/log/zones/
zoneadm.20160224T001725Z.test-net.install
```

```
$ zoneadm -z test-dev boot
```

**Example 17**    Installing an Existing Zone Configuration From a Unified Archive

This example installs a zone called web-server while using an existing zone configuration.

```
$ archiveadm info /var/tmp/example-dev-recovery.uar
Archive Information
          Creation Time:  2016-02-03T19:10:53Z
            Source Host:  example
           Architecture:  sparc
       Operating System:  Oracle Solaris 11.3 SPARC
      Deployable Systems:  example-dev
$ zoneadm list -cv
  ID NAME             STATUS      PATH                         BRAND      IP
   0 global           running     /                            solaris    shared
   - example-net      installed   /zones/example-net           solaris    excl
   - example-dev      installed   /zones/example-dev           solaris    excl
   - web-server       configured  /zones/web-server            solaris    excl
$ zoneadm -z web-server install -a /data/archives/example-dev-recovery.uar
The following ZFS file system(s) have been created:
    rpool/VARSHARE/zones/web-server
Progress being logged to /var/log/zones/zoneadm.20160224T001725Z.web-server.install
   .
       Done: Installation completed in 140.828 seconds.
 Next Steps: Boot the zone, then log into the zone console (zlogin -C)
            to complete the configuration process.
Log saved in non-global zone as
    /system/zones/web-server/root/var/log/zones/zoneadm.20160224T001725Z.web-
server.install

# zoneadm -z web-server boot
```

**Example 18**    Configuring Zones From a Unified Archive With Multiple Deployable Systems

If multiple deployable systems are contained in a Unified Archive, use the -z option in zonecfg
to specify which deployable system to use. You can use only one deployable system.

```
$ archiveadm info /var/tmp/zonearchive.uar
Archive Information
          Creation Time:  2016-02-02T17:04:11Z
            Source Host:  example
           Architecture:  sparc
       Operating System:  Oracle Solaris 11.3 SPARC
      Deployable Systems:  global,example-net,example-dev
$ zonecfg -z test-dev create -a /var/tmp/zonearchive.uar -z example-dev
```

You can now install and boot the cloned zone.

**Example 19**    Installing Zones From a Unified Archive With Multiple Deployable Systems

If multiple deployable systems are contained in a Unified Archive, use the `-z` option in `zoneadm` to specify the deployable system to install from.

```
$ zoneadm -z test-dev install -a /var/tmp/zonearchive.uar -z example-dev
```

## ▼ How to Deploy a System From a Unified Archive by Using AI

If you use Automated Installer (AI) to deploy an archive, ensure that you have a compatible AI service and AI manifest. For example, if the archived system is Oracle Solaris 11.3, any Oracle Solaris 11.3 or 11.3 SRU version AI service is compatible. Each archive does not require its own service. The specific archive deployed is determined by the manifest used for a specific deployment.

**Before You Begin**    You must be an administrator with the correct rights profiles to perform this task. See "Using Rights Profiles with Unified Archives" on page 14.

You must have an existing AI server with correct configurations, including the correct web server file directories. For details, see "Configuring an AI Server" in *Automatically Installing Oracle Solaris 11.4 Systems*.

1.  **(Optional) If needed, create a new install service on the AI server.**

    ```
    $ installadm create-service -n new-service -s iso-image
    ```

    Ensure that the ISO file includes the FMRI location for the current operating system version.

2.  **(Optional) If needed, associate the install client with a service.**

    ```
    $ installadm create-client -e client-MAC-address -n image-location
    ```

3.  **Configure an AI manifest for the install client.**
    Either edit the default manifest XML file or create a new manifest XML file referencing the archive changes. On the manifest, specify the following:

    a.  **For `software type`, specify `ARCHIVE`.**

    b.  **For `file uri`, specify the location of the archive.**

See Appendix A, "Example XML Manifests for Archive Deployment" for reference.

Refer to "Creating and Customizing an AI Manifest" in *Customizing Automated Installations With Manifests and Profiles* for different ways of creating and editing AI manifests.

4. **Create a new manifest with the updated manifest XML file.**

   $ `installadm create-manifest -n` *new-service* `-m` *manifest* `-f` *file*

5. **Confirm both that the service was created and that the new manifest was applied to the service.**

   a. **Verify the new service.**

      $ `installadm list -n` *new-service*

   b. **Verify the new manifest.**

      $ `installadm list -n` *new-service* `-m`

6. **Boot the install client.**

   ok `boot net:dhcp - install`

## ▼ How to Deploy a System From a Unified Archive by Using Bootable Media

The `create-media` subcommand creates a bootable media from a root-only Unified Archive. The resulting media image can then be used to boot and install a system from the archive content.

By default, the `archiveadm create-media` command uses USB file type when it creates a bootable media. If the image is larger than 4GB, the USB type is forced.

---

**Tip -** As best practice, use the -o option when you create bootable media to specify a path that has available space for the resulting media file. Without this option, the media would be written to a default output location. If the output location has insufficient space, media creation fails.

---

**Before You Begin**   You must be an administrator with the correct rights profiles to perform this task. See "Using Rights Profiles with Unified Archives" on page 14.

**1. Create a media image.**

Create a media image from an existing archive.

```
$ archiveadm create-media archive-name -o path-to-final-image
```

Ensure that the Unified Archive contains only root data. The step fails if the archive contains non-root data. Include the absolute or relative path to the archive when specifying *archive-name*.

**2.** Create bootable media.

- **If you downloaded the ISO image, ensure that it is on an accessible location.**

- **If you downloaded the USB image, copy the image to a USB media.**

   If the system does not have any Oracle Solaris installed, use the dd command. For example:

   ```
   # dd bs=16k conv=sync if=image-path of=/dev/rdsk/disk
   ```

   If the system with the downloaded image has an existing Oracle Solaris installation, you can use the usbcopy command instead of ddu, provided that the pkg:/install/distribution-constructor package has also been installed.

**3. Boot from the media.**

```
ok boot media - install
```

A "hands-free" installation is launched. After the installation completes and the system is rebooted, the operation prompts you for additional information to configure the system.

**Example 20**    Creating an ISO Media Image

To create an ISO image, you must include the -f option as shown below.

```
$ archiveadm create-media -f iso archive.uar -o path/image-name.iso
```

**Example 21**    Creating a Media Image Including a Specific Global Zone

By default the create-media subcommand creates media that includes all of the global zones that are included in the archive. To select a specific global zone, create the archive with the -g *global-zone* option. This command creates media from a specified zone. The resulting media image includes bootable media that contains the archive.

This example creates an archive from the global global zone.

```
$ archiveadm create-media -g global archive.uar -o path/image-name.usb
```

# ▼ How to Archive a System That Contains Zones on Shared Storage

This procedure shows how you can archive a system with zones on shared storage and deploy the archive to a kernel zone.

**Before You Begin**　You must be an administrator with the correct rights profiles to perform this task. See "Using Rights Profiles with Unified Archives" on page 14.

1. **Check which zones on shared storage exist on the system to be archived.**

   The following example shows that the system has a global zone and zone1 on shared storage.

   ```
   $ zoneadm list -icv
   ID NAME            STATUS     PATH                      BRAND       IP
    0 global          running    /                         solaris     shared
    - zone1           installed  /system/zones/zone1       solaris     excl
   $ zonecfg -z zone1 info
     zonename: zone1
     brand: solaris
     anet:
        linkname: net0
        configure-allowed-address: true
     rootzpool:
        storage: iscsi://hostname/luname.naa.600144f0dd9e090000005758ac0e0003
   ```

2. **Create a recovery archive for each system separately.**

   a. **Archive the system but exclude the zone on shared storage.**
      For example:

      ```
      $ archiveadm create -r -Z zone1 global_recovery.uar
      ```

   b. **Archive the zone on shared storage by itself.**
      For example:

      ```
      $ archiveadm create -r -z zone1 zone1_recovery.uar
      ```

3. **On the target system where you deploy the archive, configure a kernel zone, install it, then boot it.**

   ```
   $ zonecfg -z kz1 create -t SYSsolaris-kz
   ```

```
$ zoneadm -z kz1 install -a global_recovery.uar
$ zoneadm -z kz1 boot
```

4.  **Log in to the kernel zone to configure the new zone on shared storage.**

    Modify as well the storage devices for this zone. Then install the zone.

    ```
    $ zlogin -C kz1

    kz1$ zonecfg -z newzone1
    zonecfg:newzone1
    zonecfg:newzone1> create -a zone1_recovery.uar

    zonecfg:newzone1> select rootzpool
    zonecfg:newzone1:rootzpool> remove storage \
            iscsi://previous-hostname/luname.naa.600144f0dd9e090000005758ac0e0003
    zonecfg:newzone1:rootzpool> add storage \
            iscsi://new-hostname/luname.naa.600144F035FF8500000050C884E50001
    zonecfg:newzone1:rootzpool> end
    zonecfg:newzone1> commit
    zonecfg:newzone1> exit
    ```

5.  **Install, then boot, the new zone on shared storage.**

    ```
    kz1$ zoneadm -z newzone1 install -a zone1_recovery.uar
    kz1$ zlogin -C newzone1
    ```

3

# Troubleshooting Cloning and Archiving

This section discusses common troubleshooting problems for cloning and archiving. The following topics are covered:

- "Common Error Messages When Creating an Archive"
- "Common Error Messages When Deploying an Archive"
- "Common Error Messages When Creating Media from an Archive"

## Common Error Messages Encountered in Cloning and Archiving

Issues may be encountered when creating and deploying Unified Archives. Most issues related to archive creation involve system configuration, available space for archive storage, and connectivity to the package repositories which are used to prepare the image for redeployment. Common deployment issues include target configuration and selection problems and failure to configure the deployed system.

In the case of archive creation or deployment errors, logging files are created. These files contain detailed data regarding the errors encountered. Note that with Oracle Solaris Kernel Zones, some log files may be within the specific zone.

## Common Error Messages When Creating an Archive

`Archive creation failed: not enough space to create snapshot` *snapshot*. See `/system/volatile/`*archive-log*

> **Solution:** Ensure that you have enough disk space when archiving a zone to accommodate large swaps.

```
all installed zones excluded, at least one zone is required
```

**Solution:** When using the `archiveadm create` command, add a zone to be included using the `-z` option.

```
archives larger than 4GB can not use ISO format
```

**Solution:** When using the `archiveadm create-media` command, use the USB format for archives larger than 4GB.

```
estimated archive staging size of ##gb is larger than the available free space
in /dataset. Additional space needed ##gb
```

**Solution:** When an archive is created, data is streamed to the directory where the archive is created. A capacity check is performed prior to archive creation. Sometimes datasets have the `refreservation` property set which causes the capacity check to fail. In this case, it is possible for the pool to appear to have enough space for the archive, but the referred space reservation causes the failure. Note that ZFS volumes are created with a default `refreservation` which is equal to a little bit more than their size.

Reduce or remove the `refreservation` property value or run the command so that the archive is created in another dataset.

```
Executing staging capacity check... archiveadm create: unable to create
snapshot '<dataset>@<identifier>'. Command '['/usr/sbin/zfs', 'snapshot', '-r',
'<dataset>@<identifier>']' returned unexpected exit status 1 cannot create snapshot
'<zvol>@<identifier>': out of space no snapshots were created
```

**Solution:** Like the previous case, this message also indicates problems about disk space, for which the solution is identical.

When an archive is created, data is streamed to the directory where the archive is created. A capacity check is performed prior to archive creation. Sometimes datasets have the `refreservation` property set which causes the capacity check to fail. In this case, it is possible for the pool to appear to have enough space for the archive, but the referred space reservation causes the failure. Note that ZFS volumes are created with a default `refreservation` which is equal to a little bit more than their size.

Reduce or remove the `refreservation` property value or run the command so that the archive is created in another dataset.

```
excluded dataset dataset includes active boot environment, dataset dataset cannot be
excluded
```

**Solution:** When using the `archiveadm create` command, the dataset for the active boot environment cannot be excluded. Remove the dataset from the `-D` option list.

FS type '*fs-type*' is not allowed via fs-allowed property in the zone configuration. Media cannot be created.

**Solution:** When using the `archiveadm create-media` command, neither the `ufs` or `pcfs` file system type may be set as the `fs-allowed` property when creating a zone configuration.

Use the `zonecfg -z` *zone* `info` command to view the `fs-allowed` property for the zone.

multiple zones not allowed for recovery archive

**Solution:** When using the `archiveadm create` command, this error occurs when the `-z` option includes more than one zone. Instead, create a clone archive create a clone archive or create a recovery archive for each zone individually.

no global zones found in archive

When using the `archiveadm create-media` command, this error indicates that the archive does not contain a global zone. When using this command, the archive passed in needs to contain a global zone or a kernel zone archive.

only a single host global zone is allowed

**Solution:** When using the `archiveadm create-media` command, create media images for each global zone separately.

output path is not writeable

**Solution:** When using the `archiveadm create` command, check the permissions of the directory that you are trying to write the archive in.

required package system/boot/grub not installed

**Solution:** When using the `archiveadm create-media` command on an x86-based system, the system/grub/boot package must be installed. Install this package from a package repository using the following command: `pkg install system/grub/boot`.

source is not an Oracle Solaris Automated installer ISO image

**Solution:** When using the `archiveadm create-media` command, select an ISO image to use with the `-s` option.

`Source ISO does not support archive operations`

When using the `archiveadm create-media` command, the ISO image selected for media creation does not support unified archives. Create an ISO image on a system with at least Oracle Solaris 11.2 installed to create an ISO archive that supports unified archive operations.

`staging area is out of space`

**Solution:** It is possible for the pool to fill or dataset quota to be met after the capacity check is performed. If other archives are being created or data is somehow otherwise being written to the pool, it may fill. As with all IO operations, capacity planning will help avoid this issue.

`the following 'solaris-kz' branded zones must be running or excluded from archive creation:` *zone1, zone2*

**Solution:** When using the `archiveadm create` command, either boot the branded zone or exclude the zone using the `-Z` option.

`Unable to find an AI image source`

During archive or media creation, the AI image cannot be found and downloaded. Make sure the `install-image/solaris-auto-install` package is present on the available publishers.

`Unable to revert packages. Please check publishers.`

**Solution:** During archive creation, the archive package image is prepared for redeployment. Part of this process performs IPS package operations. It is likely that these operations will require data from the publishers be available. Therefore, the configured publishers must be available during archive creation.

`unable to verify ISO image provided`

During archive creation or media-creation, the AI ISO image cannot be mounted and verified. There may be a problem with the image, the mountpoint, or the LOFI device. Verify that each of these items is correct.

`'zone' and 'exclude_zone' are mutually exclusive`

**Solution:** When using the `archiveadm create` command, the same zone was listed with the `-z` option which lists zones to include when creating an archive and the `-Z` option which lists zones to exclude.

```
zone exclusion not allowed for recovery archive
```

**Solution:** When using the `archiveadm create` command, either create a clone archive with excluded zones or remove the `-Z` option to create a recovery archive.

```
zones must not be in incomplete or unavailable states for recovery archive creation
```

**Solution:** When using the `archiveadm create` command, you can exclude the incomplete or unavailable zones from the archive by using the `-Z` option.

# Common Error Messages When Deploying an Archive

```
Failed Checkpoints: target-selection Checkpoint execution error: Non-root zpool
'datapool' exists in this archive but is not specified in the manifest.
```

**Solution:** The archive which contains one or more non-root pools are being deployed without a target specification for those non-root pools. The Oracle Solaris Automated Installer will auto-select a boot device to install the root pool into, but non-root pool targets must be specified. Create an AI manifest that includes targets for non-root pools to enable the deployment to proceed.

```
IPS publishers are required for nonglobal-to-global transform deployments. Please
provide an IPS specification.
```

**Solution:** Various types of archive deployments (for example, nonglobal-to-global zone transforms) require access to IPS publishers during deployment. Publishers must be specified in an AI manifest and the publishers must be reachable during deployment from the install client.

```
zone of this brand not in archive archive.uar: Unified archive not usable
```

**Solution:** When deploying an archive to a zone, the brand in the zone configuration is not compatible with the brand of the zones in the unified archive. Either select an archive with a matching brand or change the brand.

# Common Error Messages When Creating Media from an Archive

```
unable to find a supported AI image version (consider using '--source' option)
```

**Solution:** Running the `create-media` command requires access to a compatible AI image. If the archive was created with -e option to skip the generation of bootable media, the AI image is not embedded in the archive, so the command attempts to access the AI image package from a repository. If a compatible version of the AI image package is not available from the system's publishers, the command will fail. Use the `-s` or `--source` option to specify either a compatible ISO image or the repository URI where a compatible version of the AI image package can be found.

A

# Example XML Manifests for Archive Deployment

This chapter includes example manifests for archive deployment. For further information about installation manifests, refer to Chapter 2, "Working With AI Manifests" in *Customizing Automated Installations With Manifests and Profiles*. Also refer to the `aimanifest(8)` man page. These sample manifests are provided:

## Example XML Manifests

Note that in the manifest, the archive location as specified in `file uri` must be on an accessible server, not on a client's local drive. The following `file uri`'s are valid:

- `<file uri="file:///net/`*server-IP*`/`*path*`/`*archive-file*`" />`
- `<file uri="file:///net/`*server-name*`/`*path*`/`*archive-file*`" />`
- `<file uri="http://`*server-name*`/`*archive-file*`" />`
- `<file uri="/net/`*server-IP*`/`*path*`/`*archive-file*`" />`

**EXAMPLE 22** AI Manifest to Deploy a Global Zone From a Clone Archive

In this example, the referenced clone archive contains a single global zone that can be used to deploy a global zone.

```
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="archive0">
    <target name="desired">
      <logical>
        <zpool name="rpool" is_root="true">
        </zpool>
      </logical>
    </target>
    <software type="ARCHIVE">
      <source>
        <file uri="http://example-ai.example.com/datapool/global.uar">
        </file>
      </source>
      <software_data action="install">
        <name>global</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>
```

**EXAMPLE 23** AI Manifest to Deploy a Non-Global Zone From a Clone Archive

The following example shows a manifest deploying a non-global zone from a cloned archive.

```
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="zone_default">
    <target>
      <logical>
        <zpool name="rpool">
        </zpool>
      </logical>
    </target>
    <software type="ARCHIVE">
      <source>
        <file uri="http://example-ai.example.com/archives/zone-01.uar"/>
      </source>
      <software_data action="install">
        <name>zone-01</name>
      </software_data>
    </software>
  </ai_instance>
```

```
</auto_install>
```

**EXAMPLE 24**    AI Manifest to Deploy a System Using a System Recovery Archive

The following example manifest demonstrates deploying a system using a system recovery
archive.

```
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="archive0">
    <target name="desired">
      <logical>
        <zpool name="rpool" is_root="true">
        </zpool>
      </logical>
    </target>
    <software type="ARCHIVE">
      <source>
        <file uri="http://example-ai.example.com/recovery.uar">
        </file>
      </source>
      <software_data action="install">
         <name>*</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>
```

**EXAMPLE 25**    AI Manifest to Deploy an AI Client From a Non-global Zone Archive

To transform a non-global zone to a global zone requires additional data such as drivers,
platform software and so on. The IPS publishers that contain this information must be reachable
from the AI client during the deployment. In this example, an AI client called test2 will be
configured using information from the archive named myarchive.uar and from packages from
http://pkg.oracle.com/solaris/release/. The location of the IPS repository is defined in
a second software declaration of type IPS. The IPS entry has to follow the ARCHIVE entry or the
install will fail. The archive will not be deployed on the system if the package install happens
first, so the ARCHIVE entry must be first.

```
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="default">
   <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
```

```
                      <filesystem name="export/home"/>
               </zpool>
          </logical>
       </target>  ...
       <software type="ARCHIVE">
         <source>
           <file uri="file:///net/system1/archives/myarchive.uar" />
           </file>
         </source>
         <software_data action="install">
            <name>test2</name>
         </software_data>
       </software>
       <software type="IPS">
         <source>
           <publisher name "solaris">
             <origin name="http://pkg.oracle.com/solaris/release/">
           </file>
         </source>
       </software>
  </ai_instance>
</auto_install>
```

**EXAMPLE 26**     AI Manifest to Deploy a Global Zone and Configure a Zone From the Same Archive

In addition to deploying a zone, you can also configure an additional zone from an archive by adding the following line to the manifest:

```
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="zone1">
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
        </zpool>
      </logical>
    </target>
    <software type="ARCHIVE">
      <source>
        <file uri="http://example.com/archives/mysystem-clone.uar">
        </file>
      </source>
      <software_data action="install">
        <name>zone1</name>
      </software_data>
    </software>
    <configuration type="zone" name="zone1" source="archive:myzone"
      archive="http://example.com/archives/mysystem-clone.uar"/>
```

```
      </ai_instance>
</auto_install>
```

This will configure zone1 based on the myzone instance in the mysystem-clone.uar file. Be default, zone1 will be installed from the IPS repository on first reboot, after the global zone has been installed. If you wish to use the archive to install the zone as well, use the following command:

```
# installadm create-manifest -n <service> -f <zone1-manifest-path> -m zone1 -c
 zonename="zone1"
```

**EXAMPLE  27**      AI Manifest to Deploy a Kernel Zone with Non-root Pools

This manifest requires that devices with id's 0 and 1 exist in the zone's configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE auto_install SYSTEM " file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="default">
    <target name="origin">
      <disk in_zpool="rpool" in_vdev="rpool-none" whole_disk="true">
        <disk_name name="c1d0" name_type="ctd"/>
      </disk>
      <disk in_zpool="tank" in_vdev="tank-none" whole_disk="true">
        <disk_name name="c1d1" name_type="ctd"/>
      </disk>
      <logical noswap="false" nodump="false">
        <zpool name="rpool" action="create" is_root="true" mountpoint="/rpool">
          <vdev name="rpool-none" redundancy="none"/>
        </zpool>
        <zpool name="tank" action="create" is_root="false"
            mountpoint="/tank">
          <vdev name="tank-none" redundancy="none"/>
        </zpool>
      </logical>
    </target>
    <software type="ARCHIVE">
      <source>
        <file uri="http://example.com/clone.uar"/>
      </source>
      <software_data action="install">
        <name>*</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>
```

# Index