

创建定制 Oracle® Solaris 11 安装映像

版权所有 © 2008, 2011, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，必须符合以下规定：

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并应按照许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的担保，亦不对其承担任何责任。对于因访问或使用第三方内容、产品或服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

目录

前言	5
1 创建定制安装映像简介	9
关于分发构造器	9
Oracle Solaris 映像类型	10
映像创建过程	10
SPARC 和 x86 归档文件的区别	11
2 设计定制安装映像	13
构建映像的系统要求	13
定制映像	14
样例清单	14
▼如何创建和构建定制映像	14
修改清单内容	15
创建和使用定制脚本	21
3 构建映像	23
distro_const 命令	23
▼如何一步构建映像	24
▼如何分阶段构建映像	24

前言

《创建定制 Oracle Solaris 11 安装映像》说明了如何使用 Oracle Solaris 分发构造器 (Distribution Constructor, DC) 工具构建定制 Oracle Solaris 安装映像。

相关信息

《安装 Oracle Solaris 11 系统》说明了如何使用以下任意方法安装和配置 Oracle Solaris 操作系统 (operating system, OS)：

- LiveCD 映像
- 交互式文本安装程序
- Oracle Solaris 自动化安装程序 (Automated Installer, AI) 功能
- Oracle Solaris SCI Tool 交互式系统配置工具
- `sysconfig(1M)` 命令行系统配置工具

《创建和管理 Oracle Solaris 11 引导环境》介绍了如何管理 Oracle Solaris 系统上的多个引导环境（包括非全局区域）。

《Oracle Solaris 管理：常见任务》中的第 6 章“管理服务（概述）”介绍了 Oracle Solaris 服务管理工具 (Service Management Facility, SMF) 功能。您可以使用 SMF 配置文件配置您的系统。

`pkg(5)` 手册页介绍了 Oracle Solaris 映像包管理系统 (Image Packaging System, IPS) 功能。使用该功能可以存储和检索用于安装的软件包。`pkg(1)` 手册页介绍了如何安装 IPS 软件包。

有关如何管理 Oracle Solaris 11 系统的更多信息，请参见 Oracle Solaris 11 系统管理文档。

《从 Oracle Solaris 10 JumpStart 转换至 Oracle Solaris 11 自动化安装程序》提供了可帮助您从 JumpStart 迁移到 AI 的信息，这两者都是 Oracle Solaris 的自动化安装功能。

获取 Oracle 技术支持

Oracle 客户可以通过 My Oracle Support 获取电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>，或访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>（如果您听力受损）。

印刷约定

下表介绍了本书中的印刷约定。

表 P-1 印刷约定

字体或符号	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 .login 文件。 使用 <code>ls -a</code> 列出所有文件。 <code>machine_name% you have mail.</code>
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	<code>machine_name% su</code> Password:
<i>aabbcc123</i>	要使用实名或值替换的命令行占位符	删除文件的命令为 <code>rm filename</code> 。
<i>AaBbCc123</i>	保留未译的新词或术语以及要强调的词	这些称为 <i>Class</i> 选项。 注意： 有些强调的项目在联机时以粗体显示。
新词术语强调	新词或术语以及要强调的词	高速缓存 是存储在本地的副本。 请勿保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

命令中的 shell 提示符示例

下表显示了 Oracle Solaris OS 中包含的缺省 UNIX shell 系统提示符和超级用户提示符。请注意，在命令示例中显示的缺省系统提示符可能会有所不同，具体取决于 Oracle Solaris 发行版。

表 P-2 shell 提示符

shell	提示符
Bash shell、Korn shell 和 Bourne shell	\$
Bash shell、Korn shell 和 Bourne shell 超级用户	#

表 P-2 shell 提示符 (续)

shell	提示符
C shell	machine_name%
C shell 超级用户	machine_name#

创建定制安装映像简介

系统管理员和应用程序开发者可以使用分发构造器工具构建定制 Oracle Solaris 安装映像。

- 如果您以前未创建过定制安装映像，请阅读第 9 页中的“关于分发构造器”。
- 如果您已准备好构建定制映像，请转至第 13 页中的“构建映像的系统要求”。

关于分发构造器

分发构造器是一种命令行工具，用于构建预配置的 Oracle Solaris 映像。该工具使用一个 XML 清单文件作为输入，并基于在该清单文件中指定的参数构建映像。

分发构造器可以构建 ISO 映像，该映像是一种归档文件，也称为磁盘映像，它的光盘采用国际标准化组织 (International Organization for Standardization, ISO) 定义的格式。您也可以基于生成的 ISO 映像创建 USB 映像。但与 ISO 映像不同，USB 映像只能在 x86 系统上创建和使用。

请注意以下事项：

- 根据映像配置，ISO 或 USB 映像可以是可引导的。
- ISO 映像和 USB 映像都可以安装到系统上或者在活动介质环境中运行。
- 可以将 ISO 映像刻录到 CD 或 DVD。
- 可以将 USB 映像复制到闪存驱动器。
- ISO 映像和 USB 映像都可以发布到 Internet 上。

分发构造器可创建可以在各种类型的闪存设备中使用的 USB 映像，前提是这些设备具有 Oracle Solaris 发行版提供的驱动程序支持。必须使用 `usbcopy` 实用程序才能将 USB 映像复制到 USB 闪存驱动器中。`distribution-creator` 软件包中提供了 `usbcopy` 实用程序。

Oracle Solaris 映像类型

您可以使用分发构造器创建下列类型的 Oracle Solaris 映像。

- **Oracle Solaris x86 LiveCD**—您可以创建与作为 Oracle Solaris 发行版分发的 LiveCD 映像类似的 x86 ISO 映像。还可以通过添加或删除软件包来修改该 ISO 映像的内容。您可以修改生成的引导环境的缺省设置来创建定制 ISO 映像或 USB 映像。

注—根据 LiveCD 映像中包含的软件包的大小，LiveCD 映像可能会替换为 LiveDVD 映像。

有关 LiveCD 安装的更多信息，请参见《[安装 Oracle Solaris 11 系统](#)》中的第 3 章“使用 LiveCD”。

- **Oracle Solaris x86 或 SPARC 文本安装映像**—您可以创建可用于对 Oracle Solaris 操作系统进行文本安装的 SPARC 或 x86 ISO 映像。文本安装程序可用于不需要图形卡的系统。

注—从 LiveCD 进行安装时，文本安装不会安装所包含的所有软件包。例如，文本安装程序不安装桌面。完成文本安装之后，可以添加其他软件包，例如 solaris-desktop 软件包。

有关文本安装的更多信息，请参见《[安装 Oracle Solaris 11 系统](#)》中的第 4 章“使用文本安装程序”。

- **用于自动化安装的 x86 或 SPARC ISO 映像**—Oracle Solaris 发行版包括自动化安装程序工具。自动化安装程序 (automated installer, AI) 用于自动完成在网络中的一个或多个 SPARC 和 x86 系统上安装 Oracle Solaris OS 的操作过程。安装过程可能因体系结构、安装的软件包、磁盘容量以及其他参数的不同而不同。自动化安装程序使用 SPARC 或 x86 AI ISO 映像将 Oracle Solaris OS 安装到客户机系统。可以使用分发构造器创建可用于在 SPARC 客户机上安装 Oracle Solaris OS 的 SPARC AI ISO 映像，或者创建可用于在 x86 客户机上安装 Oracle Solaris OS 的 x86 AI ISO 映像。

有关使用自动化安装程序的信息，请参见《[安装 Oracle Solaris 11 系统](#)》中的第 III 部分，“使用安装服务器安装”。

映像创建过程

分发构造器基于 XML 文件（称为“清单文件”）中指定的设置创建映像。清单文件包含使用分发构造器创建的 ISO 映像的内容和参数的规范。分发构造器包含可用于创建定制 LiveCD、x86/SPARC AI ISO 映像或 x86/SPARC 文本安装映像的样例清单。

每个清单文件中的所有字段提供预设的缺省值，这些值将创建您需要的映像类型。您可以编辑清单文件中的字段，以进一步定制所创建的映像。例如，您可以编辑清单中的目标元素，为可构造映像的构建区域指定其他位置。此外，您可以检查指定的发布者并确保您使用的系统可与该发布者联系，以便下载构建映像所需的软件包。如有必要，可以编辑软件名称元素以指定其他发布者和系统信息库位置。有关说明，请参见第 14 页中的“定制映像”。

还可以创建**定制脚本**来修改安装映像。然后，可以将检查点添加到清单文件中以运行这些定制脚本。有关详细信息，请参见第 21 页中的“创建和使用定制脚本”。

分发构造器软件包还包括一个命令行实用程序，即可解释清单规范和构建映像的 `distro_const` 命令。完成在清单文件中编辑映像蓝图以适合您的要求后，运行 `distro_const` 命令创建映像。有关详细信息，请参见第 3 章，构建映像。

您可以使用 `distro_const` 命令中提供的选项停止和重新启动处于映像生成过程中各个阶段的创建过程，以便检查和调试要创建的映像。创建过程期间停止和重新启动的该过程称为“**检查点**”。检查点是可选的。每个清单文件中都指定了缺省检查点。有关说明，请参见第 24 页中的“如何分阶段构建映像”或 `distro_const(1M)` 手册页。

SPARC 和 x86 归档文件的区别

x86 映像的根归档文件与 SPARC 映像的根归档文件不同。x86 映像的整个根归档文件（即 `boot_archive`）是 UFS 文件系统，使用 `lzma` 进行压缩。SPARC 平台不支持使用这种方法压缩整个根归档文件。而 SPARC 根归档文件使用 DCFS，它会分别压缩每个文件。这些分别压缩的文件可能需要在清单中进行特殊处理。有关说明，请参见 `dc_manifest(4)` 手册页中的 `<boot_archive_contents>` 字段。

设计定制安装映像

查看系统要求并根据本章中的介绍设计定制安装映像。

构建映像的系统要求

要使用分发构造器，必须在您的系统上进行以下设置。

表 2-1 系统要求

要求	描述
磁盘空间	分发构造器工作空间的建议最小大小是 8 GB。确认系统上有足够的空间以使用分发构造器。
Oracle Solaris 发行版	<p>必须在系统上安装 Oracle Solaris 操作系统 (operating system, OS)。请注意以下注意事项。</p> <ul style="list-style-type: none">■ 安装的系统必须具有网络访问权限。分发构造器访问网络上可用的映像包管理系统 (Image Packaging System, IPS) 系统信息库，以检索 ISO 映像的软件包。您必须具有对清单文件中指定的系统信息库的网络访问权限。■ 使用分发构造器时，在 SPARC 系统上只能创建 SPARC 映像，而在 x86 系统上只能创建 x86 映像。■ 您系统上的 Oracle Solaris 发行版本必须与分发构造器要创建的映像发行版本相同。 <p>注 - 必须成为 root 角色，才能运行分发构造器。</p>
必需的软件包	distribution-creator 软件包，包含分发构造器应用程序。

定制映像

分发构造器基于 XML 文件（称为“清单文件”）中指定的设置创建映像。清单文件包含使用分发构造器创建的 ISO 映像的内容和参数的规范。`distribution-creator` 软件包提供可用于创建定制 LiveCD、x86/SPARC AI ISO 映像或 x86/SPARC 文本安装映像的样例清单。

每个清单文件中的元素都提供预设的缺省值，这些值将创建您需要的 ISO 映像类型。可以手动编辑清单文件中的这些预设元素以定制所创建的映像。此外，还可以创建定制脚本以进一步修改映像。然后，引用清单文件中的新脚本。

样例清单

`distribution-creator` 软件包提供以下样例清单文件。

表 2-2 样例清单

清单类型	清单位置	描述
x86 LiveCD ISO 映像	<code>/usr/share/distro_const/ dc_livecd.xml</code>	用于创建与 Oracle Solaris LiveCD 类似的 ISO 映像
x86 文本安装映像	<code>/usr/share/distro_const/ dc_text_x86.xml</code>	用于创建可用于对 x86 Oracle Solaris 操作系统进行文本安装的 ISO 映像
SPARC 文本安装映像	<code>/usr/share/distro_const/ dc_text_sparc.xml</code>	用于创建可用于对 SPARC Oracle Solaris 操作系统进行文本安装的 ISO 映像
x86 AI ISO 映像	<code>/usr/share/distro_const/ dc_ai_x86.xml</code>	用于创建 x86 AI ISO 映像，以便将 Oracle Solaris 操作系统自动安装到 x86 客户机
SPARC AI ISO 映像	<code>/usr/share/distro_const/ dc_ai_sparc.xml</code>	用于创建 SPARC AI ISO 映像，以便将 Oracle Solaris 操作系统自动安装到 SPARC 客户机

▼ 如何创建和构建定制映像

- 1 下载 `distribution-creator` 软件包，它包含分发构造器应用程序和样例清单。

您可以使用软件包管理器工具安装必需的软件包。软件包管理器在 Oracle Solaris 操作系统桌面的菜单栏上提供。在菜单栏上，转至 "System"（系统）>"Administration"（管理）>"Package Manager"（软件包管理器）。

或者，使用如下 IPS 命令安装该软件包：

```
# pkg install distribution-creator
```

- 复制其中一个样例清单并使用新文件名创建定制清单文件。
使用 `distro_const` 命令创建映像时，将按名称引用清单文件。

注 – 复制原始清单文件和缺省脚本之前，始终对其进行备份。

- 编辑清单元素以适合您的需要。
例如，您可以编辑清单中的目标元素，为可构造映像的构建区域指定其他位置。并且，可以检查发布者以确保您的系统可与该发布者联系，以便下载构建映像所需的软件包。如有必要，可以编辑软件名称元素以指定其他发布者和系统信息库位置。
有关信息，请参见第 15 页中的“修改清单内容”和 `dc_manifest(4)` 手册页。
- 可选创建定制脚本以进一步修改映像。
如果确实要创建新脚本，请更新清单文件的执行部分中的脚本引用。
有关说明，请参见第 21 页中的“创建和使用定制脚本”。
- 运行 `distro_const` 实用程序以创建映像。
有关说明，请参见第 3 章，构建映像。

修改清单内容

每个清单文件中的所有字段都提供预设的缺省值，这些值将创建您需要的 ISO 映像类型。可以手动编辑清单文件中的这些预设字段，以进一步定制所创建的映像。

根据您选择的样例清单，主要元素如下所示。

表 2-3 清单元素

元素	描述
<code><distro name="Oracle_Solaris_Text_X86" add_timestamp="false"></code>	指定带有可选时间戳的映像名称
<code><boot_mods></code>	指定映像的 GRUB 菜单修改
<code><target></code>	定义可构建映像的 ZFS 构建数据集
<code><software name="transfer-ips-install" type="IPS"></code>	指定要安装的软件包的源
<code><software_data action="install"></code>	列出要安装的软件包
<code><software_data action="uninstall"></code>	列出要卸载的软件包
<code><software name="set-ips-attributes"></code>	安装完成后为 IPS 设置不同属性。

表 2-3 清单元素 (续)

元素	描述
<code><software name="ba-init"></code>	指定引导归档文件内容 注意-修改时请谨慎。如果引导归档文件不正确，则安装的系统将无法进行引导。
<code><execution stop_on_error="true"></code> <code><checkpoint name="transfer-ips-install"/></code>	列出构建检查点
<code><configuration name="pre-pkg-img-mod" type="sysconf"</code> <code>source="/etc/svc/profile/generic_limited_net.xml"></code>	指定构建期间要应用到介质的 SMF 服务 注意-很少进行修改。

提供映像标题

使用以下元素可为要构建的映像提供定制或缺省名称。

```
<distro name="Oracle_Solaris_Text_X86" add_timestamp="false">
```

如果您要构建一系列映像，并保留增量式映像，可以将时间戳变量更改为 "true"，然后时间戳将自动附加到每个映像的名称。

如果需要指定 HTTP 代理，请取消注释包含代理变量的 `distro name` 元素，然后输入代理位置。

修改引导菜单

该引导菜单元素指定要应用于映像的引导菜单修改。

在以下示例中，一个标题为 "boot1" 的专门的引导菜单将应用于映像。超时属性指定自动激活缺省引导条目之前的时间。

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

在引导菜单元素内，可以通过为每个新条目添加新的 `boot_entry` 元素来添加各个引导菜单条目。各条目被按顺序添加到引导菜单，该顺序基于每个引导条目的 `insert_at` 属性值 "start" 或 "end"。

注 - 在现有 "with magnifier" 条目之前添加新条目。

请参见以下单个 `boot_entry` 元素的示例。

```
<boot_entry>
  <title suffix>with screen reader</title suffix>
  <kernel_args>-B assistive_tech=reader</kernel_args>
</boot_entry>
```

有关详细信息，请参见 `dc_manifest(4)` 手册页。

指定构建区域

可以定制 `target` 元素。该元素定义将用于该构建的 ZFS 构建数据集。该数据集是将创建映像的区域。您必须输入有效的数据集位置。应当检查缺省构建区域，以确保该构建不会销毁需要保留在系统上的内容。如有必要，请修改构建区域。

注 - 文件系统名称不应当包括 `zpool` 的名称。

请参见以下示例。

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

指定发布者

以下元素指定分发构造器可以从其中下载软件包来构建映像的发布者。

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

在本部分的源元素中，编辑发布者名称和源名称元素来指定要使用的发布者以及软件包系统信息库所处的位置。可以列出多个发布者。当分发构造器尝试定位要安装的软件包时，按在此处列出的顺序搜索发布者。

如果需要指定发布者的镜像，请取消注释并编辑镜像名称元素。

请参见以下示例。

```
<source>
  <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"></origin>
  </publisher>
  <publisher name="publisher3.org">
    <origin name="http://example3.com/dev"></origin>
  </publisher>
</source>
```

有关使用发布者的详细信息，请参见 [《添加和更新 Oracle Solaris 11 软件包》](#)。

列出要安装的软件包

带有 `install` 属性的 `software_data` 元素根据您使用的清单，列出为了构建特定类型的映像而需安装的一组软件包。例如，`dc_livecd.xml` 清单列出构建 LiveCD 映像所需的软件包。每个名称标记列出一个软件包名称或包含许多软件包的软件包组的名称。

```
<software_data action="install">
  <name>pkg:/group/system/solaris-desktop</name>
  <name>pkg:/system/install/gui-install</name>
  <name>pkg:/system/install/media/internal</name>
</software_data>
```

如果您有要添加到映像的软件包，通过为每个软件包添加名称标记来附加软件包名称。

缺省情况下，将安装指定系统信息库中可用的最新软件包版本。如果需要其他版本，请使用以下格式将版本号附加到软件包引用中：

```
<name>pkg:/group/system/solaris-desktop@0.5.11-0.build#</name>
```

注 – 如果安装了其他冲突版本的软件包，则可能不会安装所指定的特定版本的软件包。有关详细信息，请参见 `pkg(5)` 手册页。

示例 2-1 添加软件包和其他发布者

在该示例中，指定了另外一个发布者 `mypublisher`。并且，指定了其他软件包 `mypackage1` 和 `mypackage2`。

在构建过程中，系统会按列出发布者的顺序检查这些发布者。如果在第一个发布者中没有找到软件包，将在下一个发布者中搜索指定的软件包。

```
<software name="transfer-ips-install" type="IPS">
  <destination>
    <xi:include xmlns:xi="http://www.w3.org/2003/XInclude"
      href="/usr/share/distro_const/lang_facets.xml"/>
  </destination>
  <source>
    <publisher name="solaris">
      <origin name="http://pkg.oracle.com/solaris/release"/>
    </publisher>
    <publisher name="mypublisher">
      <origin name="http://mypublisher.company.com"/>
    </publisher>
  </source>
  <software_data action="install">
    <name>pkg:/group/system/solaris-large-server</name>
    <name>pkg:/system/install/text-install</name>
    <name>pkg:/system/install/media/internal</name>
    <name>pkg:/mypackage1</name>
    <name>pkg:/mypackage2</name>
  </software_data>
</software>
```

列出要卸载的软件包

带有 `uninstall` 属性的 `software_data` 元素可用于卸载单个软件包或卸载软件包组定义。

在以下示例中，`solaris-desktop` 是包含许多单个软件包的软件包组的名称。

```
<software_data action="uninstall">
  <name>pkg:/group/system/solaris-desktop</name>
</software_data>
```

可以卸载软件包组。卸载软件包组实际只卸载了组定义。不会卸载之前作为该组的一部分安装的单个软件包。但是，可以在不卸载软件包组的情况下卸载这些单独的软件包。保留软件包组可用于正在进行的引用。还可以使用名称标记来卸载单个软件包。在卸载部分的末尾附加将卸载的其他软件包。

指定已安装系统的发布者

当使用映像（该映像是使用分发构造器创建的）安装系统后，以下元素将影响该系统。

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

提供发布者名称和可选镜像名称标记以指定已安装系统可从何处访问其他软件包并进行下载和安装。

也可以在该元素中设置 `IPS` 属性。有关 `IPS` 属性信息，请参见 `pkg(1)` 手册页。

设置构建检查点

清单中的执行元素列出在映像构建过程中执行的一系列检查点。按此部分中的排列顺序执行检查点。构建缺省安装映像所需的缺省检查点包含在每个清单中。

每个检查点名称标记都包括用于指定检查点脚本所在位置的 `mod-path` 属性。

某些缺省检查点标记包括提供了缺省值的参数。`dc_ai_sparc.xml` 样例清单的以下检查点示例会创建用于构建映像的引导归档文件，并指向将完成该任务的脚本。示例检查点还包括为每个参数提供了特定值的参数字段。

```
<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>

```

如此示例所示，<kwargs> 元素包含在构建期间需要传递到检查点的关键字参数。<kwargs> 元素内部包含 <arg name> 元素，这些元素可用于指定要传递到检查点的各个关键字。而 <arglist> 元素包含要传递到检查点的多个 <argitem> 值的列表。该示例的 <arglist> 元素中包含未压缩文件的列表。

每个 <kargs> 列表项均用双引号括起来。如果未使用双引号，或者使用一组双引号括住整个字符串，则包括空格和换行符的整个字符串将被视为一个参数。请勿在参数之间使用逗号。

如果创建将在构建映像期间使用的定制脚本，必须添加指向该脚本位置的检查点元素。定制脚本的检查点只需要一个指向定制脚本位置的 <args> 元素。有关详细信息和示例，请参见第 21 页中的“创建和使用定制脚本”。

使用 distro_const 命令选项可控制在特定的检查点停止和重新启动构建过程。请参见第 24 页中的“如何分阶段构建映像”。

示例 2-2 添加 SVR4 软件包

在此示例中，一个新检查点将被添加到清单中。该新检查点会列出要添加到映像的 SVR4 软件包及其位置。然后，会在执行部分引用该新检查点。

首先，通过添加新的 software 元素创建新检查点。该检查点指定 SVR4 作为软件类型，并指定查找软件包以及安装软件包的位置。

此外，要安装的特定 SVR4 软件包会列在 software_data 元素中。

```

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

```

示例 2-2 添加 SVR4 软件包 (续)

如果包括在检查点中，{PKG_IMAGE_PATH} 和 {BOOT_ARCHIVE} 的值将由 distro_const 实用程序分别替换为 <ZFS Dataset>/build_data/pkg_image 和 <ZFS Dataset>/build_data/boot_archive。在该示例中，SVR4 软件包将被安装到 <ZFS Dataset>/build_data/pkg_image 中。

最后，会在执行部分引用该新检查点。

```
<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"/>
```

请注意，软件名称必须与检查点名称匹配。在该示例中，这两个名称均为 "transfer-svr4-install"。

创建和使用定制脚本

分发构造器允许您指定可用于基于要构建的映像类型进行定制的其他脚本。清单文件指向脚本，脚本会将通用映像转换为特定于介质的分发。这些脚本在清单文件的执行部分中引用。可以指定任意数量的定制脚本检查点。

注-脚本的支持限于随应用程序软件包一起提供的任何未修改的缺省脚本。如果选择定制这些脚本，请先备份原始脚本。

▼ 如何创建和使用定制脚本

开始之前 创建您自己的定制脚本时，请注意以下内容：

- 脚本可以是 Python 程序、shell 脚本或二进制文件。
- 脚本以其在清单文件的执行部分中列出的顺序执行。
- 在脚本（shell 和 python 模块）内执行的命令的标准输出 (stdout) 和错误输出 (stderr) 记录在用于报告已完成构建或尝试构建的日志文件中。

1 创建新脚本。

2 将新脚本添加到起始目录或者系统或网络上的其他位置。

确保承担 root 角色的用户可以执行这些脚本。

3 通过在相应清单文件的执行部分中添加检查点来引用新脚本。

确保为脚本指定完整路径。检查点以其在清单的执行部分中列出的顺序执行。

在清单文件的执行部分中为新脚本添加引用时，必须指定可用于在该脚本执行其任务之前或之后暂停映像构建的检查点名称。或者，可以包括与检查点名称关联的定制消息。如果忽略该消息，脚本的路径将用作缺省检查点消息。在构建过程中，检查点运行时将显示检查点消息。

注 - 将有意义的名称用于检查点名称，而不是使用数字。如果添加了新脚本，这些新脚本对应的新检查点将扰乱已编号的检查点顺序。

以下示例检查点引用名为 "my-script" 的定制脚本。

```
<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 可选指定作为检查点一部分的构建参数，如下所示。

此处，{PKG_IMAGE_PATH} 被指定为参数部分中的构建参数。

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

如果包括在检查点中，{PKG_IMAGE_PATH} 和 {BOOT_ARCHIVE} 的值将由 distro_const 实用程序分别替换为 <ZFS Dataset>/build_data/pkg_image 和 <ZFS Dataset>/build_data/boot_archive。

5 构建映像。

可以一步构建映像。或者，要检查构建状态，可以在各个检查点上停止然后重新启动构建。

有关说明，请参见第 3 章，[构建映像](#)。

6 可选构建完成后，可以查看用于报告构建过程的日志文件。

构建输出会显示日志文件的位置。

构建映像

设置计划使用的清单文件后，如果需要，可以定制终结器脚本，运行 `distro_const` 命令即可创建映像。

可以按以下任一方式使用 `distro_const` 命令构建映像：

- 一步完成
- 根据需要暂停然后重新启动构建，以在构建过程中检查映像的内容并对脚本进行调试。

distro_const 命令

`distro_const` 命令的完整语法如下所示：

```
Syntax: distro_const build [-v] [-r checkpoint_name] [-p checkpoint_name] [-l] manifest
```

查看以下命令选项。

表 3-1 `distro_const` 命令选项

命令选项	描述
<code>distro_const build <i>manifest</i></code>	使用指定的清单文件一步构建映像
<code>distro_const build -v</code>	详细模式
<code>distro_const build -l <i>manifest</i></code>	列出您可以暂停和恢复构建映像的所有有效检查点
<code>distro_const build -p <i>checkpoint_name</i> <i>manifest</i></code>	在指定的检查点暂停构建映像
<code>distro_const build -r <i>checkpoint_name</i> <i>manifest</i></code>	从指定的检查点恢复构建映像
<code>distro_const build -h</code>	显示命令的帮助

注 - 必须承担 root 角色才能使用 `distro_const` 命令。

▼ 如何一步构建映像

开始之前 下载 `distribution-creator` 软件包并为映像选择清单。如果需要，对清单进行定制并添加定制脚本。

- 1 成为 root 角色。
- 2 要运行映像的完整创建而不暂停，请使用不带选项的基本 `distro_const` 命令，如下所示：

```
# distro_const build manifest
```

注 - 需要 `build` 子命令。

使用要用作映像蓝图的清单文件的名称替换 `manifest`。

例如，键入以下命令：

```
# distro_const build /usr/share/distro_const/dc_livecd.xml
```

- 3 分发构造器为映像获取所需的软件包。
- 4 分发构造器根据您在清单文件中设置的规范构建映像。
- 5 可选构建完成后，可以查看用于报告构建过程的日志文件。
构建输出会显示日志文件的位置。

▼ 如何分阶段构建映像

您可以使用 `distro_const` 命令中提供的选项停止和重新启动处于映像生成过程中各个阶段的创建过程，以便检查和调试针对要创建的映像选择的文件、软件包和脚本。此过程使用 `distro_const` 命令中提供的检查点选项，如以下基本说明中所述。

- 1 成为 root 角色。
- 2 构建映像之前，检查可以选择暂停或恢复构建映像的有效检查点。

```
# distro_const build -l manifest.xml
```

注 - 需要 `build` 子命令。

该命令显示可以暂停或恢复构建映像的有效检查点。对于其他检查点命令选项，将该命令提供的检查点名称用作有效值。

例如，以下命令确认在给定名为 `dc_livecd.xml` 的清单文件的条件下，哪些检查点可用。

```
# distro_const build -l /usr/share/distro_const/dc_livecd.xml
```

运行该命令后，将显示有效的检查点。例如，检查点可能包括以下内容。

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		Boot archive initialization
ba-config		Boot archive configuration
ba-arch		Boot archive archiving
grub-setup		Set up the GRUB menu
pkg-img-mod		Package image area modifications
create-iso		ISO image creation

注 - 在该命令输出样例中，可恢复字段中的 "X" 表示可以从该检查点重新启动构建。

3 构建映像，然后在指定的检查点上暂停构建映像。

```
# distro_const build -p checkpoint_name manifest
```

注 - 需要使用 `build` 子命令以及 `checkpoint_name` 和 `manifest` 字段。

例如，以下命令启动构建映像，并在 `ba-arch` 修改映像区域之前暂停构建：

```
# distro_const build -p ba-arch /usr/share/distro_const/dc_livecd.xml
```

4 从指定的检查点恢复构建映像。

```
# distro_const build -r checkpoint_name manifest
```

注 - 指定的检查点必须是上一次构建停止执行的检查点，或者是更早的检查点。以后的检查点无效。需要使用 `checkpoint_name` 和 `manifest` 字段以及 `build` 子命令。

例如，以下命令在 `ba-arch` 阶段恢复构建映像。

```
# distro_const build -r ba-arch /usr/share/distro_const/dc_livecd.xml
```

注 – 可以在 `build` 命令中组合使用暂停和恢复选项。

- 5 可选构建完成后，可以查看用于报告构建过程的日志文件。
构建输出会显示日志文件的位置。