

Création d'une image d'installation personnalisée d'Oracle® Solaris 11.2



Référence: E53754
Juillet 2014

Copyright © 2008, 2014, Oracle et/ou ses affiliés. Tous droits réservés.

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

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

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

U.S. GOVERNMENT END USERS:

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

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

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

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

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. 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.

Table des matières

Utilisation de la présente documentation	5
 1 Introduction à la création d'une image d'installation personnalisée	 7
A propos du constructeur de distribution	7
Types d'images Oracle Solaris	8
Processus de création d'images	9
Différences entre les archives SPARC et x86	10
 2 Création d'une image d'installation personnalisée	 11
Configuration système requise pour la création d'images	11
Personnalisation des images	12
Exemples de fichiers manifestes	12
▼ Création et génération d'une image personnalisée	13
Modification du contenu du fichier manifeste	14
Création et utilisation de scripts personnalisés	23
 3 Création d'une image	 27
Commande <code>distro_const</code>	27
▼ Création d'une image en une seule étape	28
▼ Création d'une image en plusieurs étapes	28
 Index	 31

Utilisation de la présente documentation

- **Présentation** : décrit comment créer des packages d'installation Oracle Solaris personnalisés à l'aide de l'outil du constructeur de distribution
- **Public visé** : les techniciens, aux administrateurs système et les fournisseurs de services agréés
- **Connaissances requises** : expérience de travail avec Oracle Solaris

Bibliothèque de documentation du produit

Les informations de dernière minute et les problèmes connus pour ce produit sont inclus dans la bibliothèque de documentation accessible à l'adresse : <http://www.oracle.com/pls/topic/lookup?ctx=E56338>.

Accès aux services de support Oracle

Les clients Oracle 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.

Commentaires

Faites part de vos commentaires sur cette documentation à la page suivante : <http://www.oracle.com/goto/docfeedback>.

Introduction à la création d'une image d'installation personnalisée

Les administrateurs système et les développeurs d'applications peuvent créer des images d'installation Oracle® Solaris personnalisées avec le constructeur de distribution.

- Si vous n'avez pas créé d'images d'installation personnalisée auparavant, reportez-vous à la section [“A propos du constructeur de distribution” à la page 7](#).
- Si vous êtes prêt à créer des images personnalisées, rendez-vous à la section [“Configuration système requise pour la création d'images” à la page 11](#).

A propos du constructeur de distribution

Le constructeur de distribution est un outil de ligne de commande permettant de créer des images Oracle Solaris préconfigurées. Il récupère le fichier manifeste XML en entrée et génère une image en fonction des paramètres spécifiés dans le fichier manifeste.

Le constructeur de distribution peut générer une image ISO (c'est-à-dire un fichier archive, également appelé image disque) d'un disque optique dans un format défini par l'Organisation internationale de normalisation (ISO). Vous pouvez également créer une image USB reposant sur une image ISO générée.

Remarques :

- Selon la configuration d'image, les images ISO ou USB peuvent être amorçables.
- Les images ISO et USB peuvent toutes deux être installées sur un système ou exécutées dans un environnement LiveMedia.
- Une image ISO peut être gravée sur un CD ou un DVD.
- Une image USB peut être copiée vers un lecteur flash.

Le constructeur de distribution crée une image USB compatible avec plusieurs types de périphériques de mémoire flash à condition que le système d'exploitation Oracle Solaris prenne en charge leur pilote. Il faut exécuter l'utilitaire `usbcopy` pour copier une image USB vers un lecteur flash USB. Cet utilitaire est disponible dans le package `distribution-creator`.

Types d'images Oracle Solaris

Vous pouvez créer les types d'images Oracle Solaris suivants à l'aide du constructeur de distribution.

- **Oracle Solaris x86 Live Media** : vous pouvez créer une image ISO x86 comparable à l'image Live Media distribuée avec chaque version d'Oracle Solaris. Vous pouvez également personnaliser le contenu de cette image ISO. Vous avez notamment la possibilité d'ajouter ou de supprimer des packages. Vous pouvez également modifier les paramètres par défaut pour l'environnement initialisé de manière à créer une image ISO ou USB personnalisée.

Pour plus d'informations sur les installations Live Media, reportez-vous au [Chapitre 3, “ Utilisation de Live Media ” du manuel “ Installation des systèmes Oracle Solaris 11.2 ”](#). Pour plus d'informations sur la personnalisation du contenu d'une image, reportez-vous à la section [“ Modification du contenu du fichier manifeste ” à la page 14](#).

- **Image d'installation en mode texte d'Oracle Solaris pour plates-formes x86 ou SPARC** : vous pouvez créer une image ISO SPARC ou x86 en vue de procéder à l'installation en mode texte du système d'exploitation Oracle Solaris. Le programme d'installation en mode texte peut être utilisé sur des systèmes ne nécessitant pas de carte graphique.

Remarque - Une installation en mode texte n'installe *pas* l'intégralité des packages logiciels inclus lors de l'installation à partir de l'image Live Media. Par exemple, le programme d'installation en mode texte n'installe pas de bureau. Après une installation en mode texte, vous pouvez ajouter des packages supplémentaires, tels que le package `solaris-desktop`.

Pour plus d'informations sur les installations en mode texte, reportez-vous au [Chapitre 4, “ Utilisation du programme d'installation en mode texte ” du manuel “ Installation des systèmes Oracle Solaris 11.2 ”](#).

- **Image ISO x86 ou SPARC en vue d'une installation automatisée** : le système d'exploitation Oracle Solaris inclut l'outil d'installation automatisée. Le programme d'installation automatisée (AI) permet d'automatiser l'installation du SE Oracle Solaris sur un ou plusieurs systèmes SPARC et x86 sur un réseau. L'architecture, les packages installés, la capacité du disque et d'autres paramètres peuvent varier d'une installation à une autre. Vous pouvez vous servir du constructeur de distribution pour créer une image ISO d'installation automatisée SPARC permettant d'installer le SE Oracle Solaris sur des clients SPARC, ou pour créer une image ISO d'installation automatisée x86 permettant d'installer le SE Oracle Solaris sur des clients x86.

Pour plus d'informations sur l'utilisation du programme d'installation automatisée, reportez-vous à la [Partie III, “ Installation à l'aide d'un serveur d'installation ” du manuel “ Installation des systèmes Oracle Solaris 11.2 ”](#).

Processus de création d'images

Le constructeur de distribution crée des images en fonction des paramètres indiqués dans des fichiers XML appelés *fichiers manifestes*. Les fichiers manifestes renferment les spécifications relatives au contenu et aux paramètres des images ISO créées à l'aide du constructeur de distribution. Le package du constructeur de distribution contient des exemples de fichiers manifestes par le biais desquels vous pouvez créer une image ISO x86 Live Media personnalisée, une image ISO d'installation automatisée pour systèmes x86 ou SPARC, ou encore une image ISO d'installation en mode texte pour plates-formes x86 ou SPARC. Reportez-vous à la section [“Exemples de fichiers manifestes” à la page 12](#).

Tous les champs des fichiers manifestes fournissent des valeurs par défaut prédéfinies qui créent le type d'image dont vous avez besoin. Vous pouvez modifier les champs dans le fichier manifeste afin de personnaliser davantage l'image obtenue. Par exemple :

- Vous pouvez modifier l'élément `target` du fichier manifeste afin d'indiquer un autre emplacement pour la zone de compilation dans laquelle créer l'image.
- Vous pouvez vérifier l'éditeur spécifié et confirmer que le système utilisé peut contacter cet éditeur pour télécharger les packages nécessaires à la création de l'image.
- Vous pouvez modifier l'élément `software name` afin de spécifier un autre éditeur, ainsi que l'emplacement du référentiel correspondant.

Pour plus d'instructions, reportez-vous à la section [“Personnalisation des images” à la page 12](#).

Vous pouvez également créer des *scripts personnalisés* pour modifier votre image d'installation. Vous pouvez ensuite ajouter des points de contrôle au fichier manifeste pour exécuter ces scripts personnalisés. Pour plus d'informations, reportez-vous à la section [“Création et utilisation de scripts personnalisés” à la page 23](#).

Le package du constructeur de distribution comprend également un utilitaire de ligne de commande (la commande `distro_const`) qui interprète les spécifications du fichier manifeste et génère l'image. Après avoir modifié l'empreinte de l'image en fonction de vos besoins dans un fichier manifeste, exécutez la commande `distro_const` pour construire l'image. Pour plus d'informations, reportez-vous au [Chapitre 3, Création d'une image](#).

Vous pouvez ajouter des options à la commande `distro_const` pour arrêter et redémarrer la compilation à différentes étapes du processus de génération, afin de vérifier l'image en cours de création et de suivre une procédure de débogage, le cas échéant. Ce processus d'arrêt et de redémarrage au cours de la génération est appelé *création de points de contrôle*. La création de points de contrôle est facultative. Les valeurs par défaut des points de contrôle sont indiquées dans chaque fichier manifeste.

Après avoir exécuté la commande `distro_const`, vous pouvez consulter les informations de compilation dans le fichier `journal` simple ou détaillé.

Pour plus d'informations, reportez-vous à la section “Création d'une image en plusieurs étapes” à la page 28 ou à la page de manuel [distro_const\(1M\)](#).

Différences entre les archives SPARC et x86

L'archive root pour les images x86 diffère de l'archive root pour les images SPARC. L'archive whole root (ou boot_archive) pour les images x86 est un système de fichiers UFS compressé à l'aide de lzma. La plate-forme SPARC ne prend pas en charge la compression de l'archive whole root de cette manière. Au lieu de cela, les archives SPARC utilisent DCFS, qui compresse chaque fichier individuellement. Ces fichiers compressés séparément peuvent nécessiter un traitement spécifique dans le fichier manifeste. Pour plus d'instructions, reportez-vous au champ <boot_archive_contents> de la page de manuel [dc_manifest\(4\)](#).

♦ ♦ ♦ C H A P I T R E 2

Création d'une image d'installation personnalisée

Ce chapitre indique la configuration système requise et décrit la procédure de création d'une image d'installation personnalisée.

Configuration système requise pour la création d'images

Pour pouvoir utiliser le constructeur de distribution, votre système doit satisfaire aux exigences répertoriées dans le tableau suivant.

TABLEAU 2-1 Configuration système requise

Configuration requise	Description
Espace disque	La taille minimale recommandée pour l'espace de travail du constructeur de distribution est de 8 Go. Vérifiez que vous disposez de suffisamment d'espace sur votre système pour utiliser le constructeur de distribution.
Système d'exploitation Oracle Solaris	Un système d'exploitation (SE) Oracle Solaris doit être installé sur votre système. Tenez compte des points suivants : ■ Votre système installé doit avoir accès au réseau. Le constructeur de distribution accède aux référentiels IPS (Image Packaging System, système d'emballage d'image) disponibles sur le réseau pour récupérer les packages pour l'image ISO. Vous devez disposer d'un accès réseau aux référentiels spécifiés dans le fichier manifeste. ■ Dans le constructeur de distribution, vous pouvez créer soit des images SPARC sur un système SPARC, soit des images x86 sur un système x86. ■ La version d'Oracle Solaris installée sur votre système doit être la même que celle de l'image que vous souhaitez créer avec le constructeur de distribution. Remarque - Vous devez vous connecter avec le rôle root pour exécuter le constructeur de distribution.
Packages requis	Le package <code>distribution-creator</code> , qui contient l'outil du constructeur de distribution.

Personnalisation des images

Le constructeur de distribution crée des images en fonction des paramètres indiqués dans des fichiers XML appelés *fichiers manifestes*. Les fichiers manifestes renferment les spécifications relatives au contenu et aux paramètres des images ISO créées à l'aide du constructeur de distribution. Le package `distribution-creator` contient des exemples de fichiers manifestes par le biais desquels vous pouvez créer une image ISO x86 Live Media personnalisée, une image ISO d'installation automatisée pour systèmes x86 ou SPARC, ou encore une image ISO d'installation en mode texte pour plates-formes x86 ou SPARC.

Les éléments présents dans chaque fichier manifeste fournissent des valeurs par défaut prédéfinies qui créent le type d'image ISO dont vous avez besoin. Vous pouvez modifier manuellement ces éléments prédéfinis dans un fichier manifeste afin de personnaliser davantage l'image obtenue. En outre, vous pouvez créer des scripts personnalisés pour modifier davantage votre image. Référez-vous ensuite les nouveaux scripts dans le fichier manifeste.

Exemples de fichiers manifestes

Le package `distribution-creator` fournit les exemples de fichiers manifestes décrits dans le tableau suivant.

TABEAU 2-2 Exemples de fichiers manifestes

Type de fichier manifeste	Emplacement du fichier manifeste	Description
Image ISO x86 Live Media	<code>/usr/share/distro_const/ dc_livecd.xml</code>	Créer une image ISO x86 comparable à l'image Oracle Solaris Live Media
Image d'installation en mode texte x86	<code>/usr/share/distro_const/ dc_text_x86.xml</code>	Créer une image ISO x86 pour procéder à l'installation en mode texte du système d'exploitation Oracle Solaris sur une plate-forme x86
Image d'installation en mode texte SPARC	<code>/usr/share/distro_const/ dc_text_sparc.xml</code>	Créer une image ISO SPARC en vue de procéder à l'installation en mode texte du système d'exploitation Oracle Solaris sur une plate-forme SPARC
Image ISO d'installation automatisée x86	<code>/usr/share/distro_const/ dc_ai_x86.xml</code>	Créer une image ISO d'installation automatisée x86 pour procéder à l'installation automatisée du SE Oracle Solaris sur des clients x86
Image ISO d'installation automatisée SPARC	<code>/usr/share/distro_const/ dc_ai_sparc.xml</code>	Créer une image ISO d'installation automatisée SPARC pour procéder à l'installation automatisée du SE Oracle Solaris sur des clients SPARC

▼ Création et génération d'une image personnalisée

Cette procédure décrit les étapes générales à suivre pour créer et compiler une image personnalisée.

1. Connectez-vous en tant qu'administrateur.

Pour plus d'informations, reportez-vous à la section “ [A l'aide de vos droits administratifs attribués](#) ” du manuel “ [Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2](#) ”.

2. Installez le package `distribution-creator`, qui contient l'outil du constructeur de distribution et des exemples de fichiers manifestes.

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

3. Copiez un des exemples de fichiers manifestes et créez un fichier manifeste personnalisé avec un nouveau nom.

Il faut référencer le fichier manifeste par son nom lorsque vous exécutez la commande `distro_const` pour créer une image.

Remarque - Sauvegardez toujours le fichier manifeste d'origine et les scripts par défaut avant de les copier.

4. Modifiez les éléments du fichier manifeste.

Vous pouvez par exemple modifier l'élément `target` du fichier manifeste afin d'indiquer un autre emplacement pour la zone de compilation dans laquelle créer l'image. Vous pouvez également vérifier l'éditeur pour confirmer que le système peut le contacter afin de télécharger les packages nécessaires à la création de l'image. Si nécessaire, vous pouvez modifier l'élément `software name` pour spécifier un éditeur et un emplacement de référentiel différents.

Pour plus d'informations, reportez-vous à la section “ [Modification du contenu du fichier manifeste](#) ” à la page 14 et à la page de manuel `dc_manifest(4)`.

5. (Facultatif) Créez des scripts personnalisés pour modifier davantage l'image.

Si vous rédigez de nouveaux scripts, mettez à jour leurs références dans la section `execution` du fichier manifeste.

Pour obtenir des instructions, reportez-vous à la section “ [Création et utilisation de scripts personnalisés](#) ” à la page 23.

6. Exécutez l'utilitaire `distro_const` pour créer une image.

Pour obtenir des instructions, reportez-vous au [Chapitre 3, Création d'une image](#).

Modification du contenu du fichier manifeste

Tous les champs des fichiers manifestes contiennent des valeurs par défaut prédéfinies qui créent le type d'image ISO dont vous avez besoin. Vous pouvez modifier manuellement ces champs prédéfinis dans un fichier manifeste afin de personnaliser davantage l'image obtenue.

Le tableau suivant décrit les principaux éléments des exemples de fichiers manifestes.

TABEAU 2-3 Eléments du fichier manifeste

Élément	Description
<code><distro name="Oracle_Solaris_Text_X86" add_timestamp="false"></code>	Spécifie le nom de l'image avec un horodatage facultatif
<code><boot_mods></code>	Spécifie les modifications du menu GRUB à appliquer à l'image
<code><target></code>	Définit le jeu de données ZFS dans lequel l'image est créée
<code><software name="transfer-ips-install" type="IPS"></code>	Spécifie la source des packages logiciels à installer
<code><software_data action="install"></code>	Répertorie les packages à installer
<code><software_data action="uninstall"></code>	Répertorie les packages à désinstaller
<code><software name="set-ips-attributes"></code>	Définit différents attributs pour l'IPS au terme de l'installation
<code><software name="ba-init"></code>	Spécifie le contenu de l'archive d'amorçage Attention - Modifiez cet élément avec précaution. Si l'archive d'amorçage est incorrecte, le système installé ne sera pas en mesure de s'initialiser.
<code><execution stop_on_error="true"></code> <code><checkpoint name="transfer-ips-install"/></code>	Répertorie les points de contrôle de création
<code><configuration name="pre-pkg-img-mod" type="sysconf"</code> <code>source="/etc/svc/profile/generic_limited_net.xml"></code>	Spécifie les services SMF à appliquer aux médias lors de la création Attention - Modifiez cet élément le plus rarement possible.

Indication d'un titre pour l'image

Utilisez l'élément suivant pour attribuer un nom personnalisé ou par défaut à l'image que vous allez créer :

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

Si vous envisagez de créer toute une série de versions d'une image et de conserver les images incrémentielles, vous pouvez définir la variable d'horodatage sur "true". Ainsi, la date et l'heure seront automatiquement ajoutées au nom de chaque image.

Si vous devez spécifier un proxy HTTP, supprimez les marques de commentaire de l'élément `distro name` incluant la variable de proxy, puis saisissez l'emplacement du proxy.

Modification du menu d'initialisation

L'élément `boot_mods` indique les modifications du menu d'initialisation à appliquer à l'image.

Dans l'exemple ci-dessous, un menu d'initialisation spécialisé portant le titre "boot1" sera appliqué à l'image. L'attribut de délai d'attente spécifie la durée avant l'activation automatique de l'entrée d'initialisation par défaut.

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

Pour ajouter une entrée de menu d'initialisation individuelle, il suffit d'insérer un nouvel élément `boot_entry` dans l'élément du menu d'initialisation. Les entrées sont ajoutées séquentiellement au menu d'initialisation, dans l'ordre défini par la valeur d'attribut `insert_at` "start" ou "end" de chaque entrée d'initialisation.

Remarque - Ajoutez les nouvelles entrées avant l'entrée "with magnifier" existante.

Reportez-vous à l'exemple ci-après d'un élément `boot_entry` individuel.

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

Pour plus d'informations, reportez-vous à la page de manuel [dc_manifest\(4\)](#).

Spécification de la zone de compilation

Vous pouvez personnaliser l'élément `target`. Cet élément définit le jeu de données ZFS à utiliser pour la création. Ce jeu de données correspond à la zone où l'image sera créée. Il faut indiquer un emplacement de jeu de données valide. Vérifiez la zone de compilation par défaut afin de garantir que la compilation ne détruira pas le contenu que vous devez conserver sur le système. Modifiez la zone de compilation si nécessaire.

Remarque - Le nom du système de fichiers ne doit pas inclure le nom du `zpool`.

L'exemple suivant illustre un élément `target`.

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

```
</zpool>
</logical>
</target>
```

Spécification de l'éditeur

L'élément suivant spécifie un éditeur auprès duquel le constructeur de distribution peut obtenir des packages à télécharger et à utiliser pour créer l'image.

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

Dans l'élément source imbriqué dans la section software name, modifiez les éléments publisher name et origin name afin de spécifier l'éditeur à utiliser, ainsi que l'emplacement du référentiel de packages. L'emplacement du référentiel peut prendre la forme d'un chemin NFS ou d'un répertoire local. Plusieurs éditeurs peuvent être répertoriés. Quand le constructeur de distribution tente de localiser les packages à installer, les éditeurs sont parcourus dans l'ordre dans lequel ils sont répertoriés ici.

S'il faut spécifier des copies miroir d'un éditeur, supprimez les marques de commentaire et modifiez l'élément mirror name.

L'exemple suivant illustre un élément source imbriqué dans l'élément software name.

```
<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"/>
  </publisher>
  <publisher name="publisher3.org">
    <origin name="http://example3.com/dev"/>
  </publisher>
  <publisher name="publisher4">
    <origin name="file:///net/myserver/publisher4/repo"/>
  </publisher>
</source>
```

Pour plus d'informations sur l'utilisation des éditeurs, reportez-vous à la section “ [Ajout et mise à jour de logiciels dans Oracle Solaris 11.2](#) ”.

Spécification des packages à installer

L'élément software_data avec l'attribut install répertorie l'ensemble des packages à installer afin de créer un type d'image particulier, en fonction du manifeste utilisé. Par exemple, le

fichier manifeste `dc_livecd.xml` répertorie les packages requis pour créer une image Live Media. Chaque balise `name` indique le nom d'un package individuel ou le nom d'un package de groupe contenant plusieurs packages.

```
<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>
```

Si vous souhaitez ajouter des packages à l'image, ajoutez leur nom en insérant une balise `name`.

Par défaut, la version la plus récente du package disponible dans le référentiel spécifié est installée. Si une autre version est requise, ajoutez le numéro de version à la référence du package en respectant le format suivant :

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

Remarque - La version d'Oracle Solaris installée sur votre système doit être la même que celle de l'image que vous souhaitez créer avec le constructeur de distribution.

En outre, les packages d'une version particulière spécifiée risquent de ne pas être installés si d'autres packages d'une version en conflit sont en cours d'installation, comme indiqué par le fichier manifeste d'un service d'installation automatisée. Reportez-vous au [Chapitre 9, “ Personnalisation des installations ”](#) du manuel “ [Installation des systèmes Oracle Solaris 11.2](#) ”.

EXEMPLE 2-1 Ajout de packages et d'éditeurs supplémentaires

Dans cet exemple, un deuxième éditeur (`mypublisher`) est spécifié. Des packages supplémentaires (`mypackage1` et `mypackage2`) sont spécifiés.

Au cours du processus de compilation, les éditeurs sont vérifiés dans l'ordre dans lequel ils sont répertoriés. Si les packages ne sont pas disponibles auprès du premier éditeur, l'éditeur suivant est parcouru à la recherche des packages spécifiés.

```
<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>
```

```
</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>
```

Spécification des packages à désinstaller

Il est possible de désinstaller un package individuel ou une définition de package de groupe par le biais de l'élément `software_data` avec l'attribut `uninstall`.

Remarque - Une **définition de package de groupe** lie tous les packages individuels qui font partie du groupe en une seule et même unité pouvant faire l'objet d'opérations spécifiques.

L'attribut `uninstall` s'avère particulièrement utile si vous souhaitez installer un package de groupe complet en excluant un ou plusieurs packages individuels. Vous pouvez utiliser l'attribut `uninstall` pour supprimer la définition de package de groupe en premier lieu. Vous pouvez ensuite désinstaller les packages précédemment installés dans le cadre d'un package de groupe.

Vous avez peut-être choisi de créer une image d'installation Live Media. L'image d'installation Live Media par défaut inclut un navigateur Firefox dans le package de groupe de bureau.

Si vous voulez exclure le navigateur Firefox de l'image à créer, effectuez les opérations suivantes :

1. Installez le package de groupe `solaris-desktop` qui inclut tous les logiciels du bureau Live Media classique. Reportez-vous à la section [“Spécification des packages à installer” à la page 16](#).
2. Désinstallez la définition du package de groupe `solaris-desktop` à l'aide de l'attribut `uninstall`, comme suit :

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

Remarque - L'exécution de l'attribut `uninstall` ne fait que supprimer la définition du package de groupe. Tous les packages individuels que le groupe englobe restent installés.

3. Maintenant que les packages individuels ne sont plus liés ensemble dans une définition de groupe, vous pouvez exécuter à nouveau l'attribut `uninstall` pour désinstaller le package Firefox.

```
<software_data action="uninstall">
  <name>pkg:/web/browser/firefox</name>
</software_data>
```

Vous pouvez éventuellement combiner les étapes 2 et 3 dans une seule entrée, comme suit :

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

Ajoutez les packages supplémentaires à désinstaller à la fin de la section `uninstall`.

Spécification d'un éditeur pour un système installé

L'élément `software name` s'applique à un système une fois qu'il a été installé avec l'image générée à l'aide du constructeur de distribution.

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

Fournissez le nom de l'éditeur et des balises de nom de copie miroir facultatives pour spécifier l'endroit où le système installé peut accéder à d'autres packages à télécharger et à installer.

Vous pouvez également définir des attributs IPS dans cet élément. Reportez-vous aux informations de la propriété IPS de la page de manuel [pkg\(1\)](#).

Définition de points de contrôle de compilation

L'élément `execution` du fichier manifeste répertorie une série de points de contrôle exécutés lors de la compilation d'une image. Les points de contrôle sont exécutés dans l'ordre dans lequel ils sont répertoriés dans cette section. Les points de contrôle par défaut nécessaires à la compilation de l'image d'installation par défaut sont inclus dans chaque fichier manifeste.

Au cours du processus de génération de l'image, les points de contrôle modifient le contenu de la zone de compilation spécifiée dans le fichier manifeste.

La zone de compilation contient les répertoires suivants :

- `ZFS dataset/build_data/pkg_image`
- `ZFS dataset/build_data/boot_archive`

La variable `ZFS dataset` est spécifiée par l'élément `target` dans le fichier manifeste.

Au cours du processus de compilation, toutes les entités incluses dans l'image finale sont ajoutées au répertoire à l'annuaire `pkg_image`. Les fichiers stockés dans le répertoire séparé `boot_archive` servent à créer une archive d'amorçage pendant le processus de création, qui est également ajoutée au répertoire `pkg_image`.

La liste suivante décrit brièvement chaque point de contrôle par défaut, dans l'ordre dans lequel ils sont exécutés dans la plupart des fichiers manifestes.

- `transfer-ips-install` : au niveau de ce point de contrôle, le constructeur de distribution contacte les éditeurs IPS et ajoute à l'image les packages répertoriés dans l'élément `software_data` du fichier manifeste.
- `set-ips-attributes` : au niveau de ce point de contrôle, le constructeur définit l'éditeur que le système installé doit utiliser. Les valeurs définies par ce point de contrôle n'ont aucune incidence si vous créez une image d'installation automatisée.
- `pre-pkg-img-mod` : au niveau de ce point de contrôle, le constructeur importe dans l'image les fichiers de service SMF spécifiés dans l'élément `configuration` du fichier manifeste. Le constructeur modifie également certains fichiers pour optimiser l'image.

Toutes les modifications apportées jusqu'ici sont incluses à la fois dans l'image en cours de création et dans l'archive root. Dans les scripts personnalisés, il faut ajouter les nouveaux points de contrôle qui doivent modifier aussi bien l'archive root que l'image avant ou juste après le point de contrôle `pre-pkg-img-mod`.

- `ba-init` : au niveau de ce point de contrôle, le constructeur place les fichiers répertoriés dans la section `ba-init` du fichier manifeste dans l'archive root. Ces fichiers sont copiés de la zone `pkg_image` vers la zone `root_archive`.
- `ba-config` : au niveau de ce point de contrôle, le constructeur apporte d'autres modifications aux fichiers précédemment copiés dans l'archive root. Le constructeur crée des liens symboliques pointant vers d'autres fichiers requis à une étape ultérieure du processus d'initialisation afin de réduire la taille de l'archive root.
- `ba-arch` : au niveau de ce point de contrôle, le constructeur compresse l'archive root avant de la placer dans le répertoire `pkg_image`. Le constructeur effectue également les opérations possibles pour optimiser l'archive root en fonction du type de système en cours de création. Après ce point de contrôle, aucune modification définie dans des scripts personnalisés n'est intégrée dans l'archive root dans la mesure où celle-ci a déjà été compressée.
- `boot-setup` : au niveau de ce point de contrôle, le constructeur définit le menu GRUB2 en fonction des entrées spécifiées dans la section `boot_entry` du fichier manifeste. Ce point de contrôle ne s'applique qu'aux images destinées aux systèmes x86.
- `pkg-img-mod` : au niveau de ce point de contrôle, le constructeur crée les principales archives pour l'image en cours de création et optimise la zone `pkg_image`. Le constructeur déplace des fichiers dans le répertoire `pkg_image` en créant l'archive de l'image. Tous les fichiers stockés dans le répertoire `pkg_image` sont inclus dans l'image. Aucun fichier ajouté après ce point de contrôle ne sera inclus dans l'image.
- `create-iso` : ce point de contrôle génère les fichiers `.iso` en incluant toutes les entités stockées dans le répertoire `pkg_image`.

- `create-usb` : ce point de contrôle génère le fichier `.usb` à partir du fichier `.iso` généré.

Dans les champs spécifiques des sections de points de contrôle, chaque balise checkpoint name inclut l'attribut `mod-path` qui spécifie l'emplacement du script correspondant.

Certaines balises de points de contrôle par défaut incluent des arguments avec des valeurs par défaut fournies. L'exemple de point de contrôle suivant (extrait de l'exemple de fichier manifeste `dc_ai_sparc.xml`) crée l'archive d'amorçage pour la compilation de l'image et renvoie au script chargé d'accomplir cette tâche. L'exemple de point de contrôle inclut également des champs `argitem` avec des valeurs spécifiques fournies pour chaque 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>
```

Comme indiqué dans cet exemple, l'élément `kwargs` contient des arguments de type mot-clé à transmettre au point de contrôle au cours de la compilation. L'élément `kwargs` contient des éléments `arg name` qui permettent de spécifier des mots-clés individuels à transmettre au point de contrôle. Enfin, l'élément `arglist` contient une liste de valeurs `argitem` à transmettre au point de contrôle. Cet exemple inclut la liste des fichiers décompressés dans l'élément `arglist`.

Chaque élément de liste `kargs` est indiqué entre guillemets. En l'absence de guillemets ou si l'intégralité de la chaîne est indiquée entre guillemets, celle-ci (espaces et retours inclus) est interprétée comme un argument. N'insérez pas de virgules pour séparer les arguments.

Si vous rédigez un script personnalisé à utiliser lors de la génération d'une image, il faut ajouter un élément de point de contrôle renvoyant à l'emplacement du script. Le point de contrôle nécessite un seul élément `args` pointant vers le script personnalisé. Pour plus d'informations et d'exemples, reportez-vous à la section [“Création et utilisation de scripts personnalisés” à la page 23](#).

Utilisez les options de la commande `distro_const` pour contrôler l'interruption et le redémarrage du processus de compilation au niveau de points de contrôle particuliers. Reportez-vous à la section [“Création d'une image en plusieurs étapes” à la page 28](#).

EXEMPLE 2-2 Ajout de packages SVR4

Dans cet exemple, un nouveau point de contrôle est ajouté au fichier manifeste. Ce nouveau point de contrôle répertorie les packages SVR4 à ajouter à l'image, ainsi que leur emplacement. Ensuite, ce nouveau point de contrôle est référencé dans la section `execution`.

Tout d'abord, le nouveau point de contrôle est créé par l'ajout d'un nouvel élément `software`. Ce point de contrôle indique SVR4 comme type de logiciel, l'emplacement des packages et la destination d'installation.

En outre, les packages SVR4 spécifiques à installer sont répertoriés dans l'élément `software_data`.

```
<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>
```

Si les valeurs `{PKG_IMAGE_PATH}` et `{BOOT_ARCHIVE}` sont incluses dans le point de contrôle, l'utilitaire `distro_const` les remplace respectivement par `ZFS dataset/build_data/pkg_image` et `ZFS dataset/build_data/boot_archive`. Dans cet exemple, les packages SVR4 sont installés sous `ZFS dataset/build_data/pkg_image`.

Enfin, ce nouveau point de contrôle est référencé dans la section `execution`. Le nom du point de contrôle peut être n'importe quelle chaîne, mais pour cet exemple `checkpoint_class` doit être `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"/>
```

Notez que le nom du logiciel doit correspondre au nom du point de contrôle. Dans cet exemple, les deux sont "transfer-svr4-install".

EXEMPLE 2-3 Création de hachages d'un média

Ce point de contrôle permet aux utilisateurs de générer automatiquement des hachages du média généré par `distro_const` à l'aide des sommes de contrôles du manifeste DC.

```
<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>
```

`arglist` inclut tous les algorithmes utilisés pour générer des hachages pour le média généré. Chaque `argitem` spécifie un algorithme. Les algorithmes valides peuvent être déterminés en exécutant `/usr/bin/digest -l`. Chaque `argitem` peut avoir un attribut `path` qui spécifie le chemin absolu d'un fichier supplémentaire ajouté avec les hachages produits par cet algorithme. Si aucun algorithme n'est spécifiée, la valeur par défaut est `md5`.

Lors de l'image est créée, les fichiers seront générés pour chaque algorithme contenant les sommes de contrôle de chaque média.

Création et utilisation de scripts personnalisés

Le constructeur de distribution vous permet de spécifier des scripts supplémentaires afin de personnaliser l'image selon son type au cours du processus de création. Les fichiers manifestes renvoient aux scripts, et ceux-ci transforment l'image générique en distribution propre au média. Ces scripts sont référencés dans la section `execution` des fichiers manifestes. Vous pouvez spécifier le nombre de points de contrôle de scripts personnalisés de votre choix.

Remarque - La prise en charge des scripts est limitée aux scripts par défaut (non modifiés) fournis avec les packages d'application. Si vous décidez de personnaliser ces scripts, commencez par sauvegarder les scripts d'origine.

D'autre part, notez que les scripts spécifiés dans la section execution du fichier manifeste sont exécutés au cours du processus de création de l'image. La section execution ne référence pas de scripts de pré-installation ou de post-installation.

Lorsque vous créez vos propres scripts personnalisés, tenez compte des points suivants :

- Les scripts peuvent être des programmes Python, des scripts shell ou des fichiers binaires.
- Les scripts sont exécutés dans leur ordre d'apparition dans la section execution du fichier manifeste.
- La sortie standard (`stdout`) et la sortie d'erreur (`stderr`) des commandes exécutées dans les scripts (modules shell et python) sont consignées dans des fichiers journaux qui établissent des rapports sur les compilations terminées ou tentées.

▼ Création et utilisation d'un script personnalisé

1. **Créez un nouveau script.**
2. **Ajoutez les nouveaux scripts à votre répertoire personnel ou ailleurs sur le système ou réseau.**

Assurez-vous qu'un utilisateur connecté avec le rôle root peut exécuter ces scripts.

3. **Référez le nouveau script en ajoutant un point de contrôle dans la section execution du fichier manifeste approprié.**

Pour déterminer où ajouter le nouveau point de contrôle, consultez la description des points de contrôle par défaut à la section [“Définition de points de contrôle de compilation” à la page 19](#).

Assurez-vous de spécifier le chemin d'accès complet à vos scripts. Les points de contrôle sont exécutés dans leur ordre d'apparition dans la section execution du fichier manifeste.

Lorsque vous ajoutez une référence à un nouveau script dans la section execution d'un fichier manifeste, il faut spécifier le nom d'un point de contrôle permettant d'interrompre la création de l'image avant que ce script n'ait effectué sa tâche ou bien après. Vous pouvez également inclure un message personnalisé associé au nom du point de contrôle. Si vous n'ajoutez pas de message, le chemin du script est utilisé comme message de point de contrôle par défaut. Un message relatif au point de contrôle s'affiche lorsque celui-ci est exécuté au cours du processus de compilation.

Remarque - Attribuez des noms explicites aux points de contrôle, plutôt que des nombres. Si de nouveaux scripts sont ajoutés, les nouveaux points de contrôle correspondants perturbent l'ordre des points de contrôle numérotés.

L'exemple de point de contrôle suivant référence un script personnalisé nommé "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. (Facultatif) Spécifiez un paramètre de compilation en tant que partie intégrante du point de contrôle, comme suit.

{PKG_IMAGE_PATH} est spécifié en tant que paramètre de compilation dans la section des arguments.

```
<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>
```

Si les valeurs {PKG_IMAGE_PATH} et {BOOT_ARCHIVE} sont incluses dans le point de contrôle, l'utilitaire `distro_const` les remplace respectivement par *ZFS dataset/build_data/pkg_image* et *ZFS dataset/build_data/boot_archive*.

5. Création de l'image.

Vous pouvez créer une image en une seule étape. Pour vérifier l'état de la compilation, vous pouvez également arrêter et redémarrer le processus au niveau de différents points de contrôle.

Pour obtenir des instructions, reportez-vous au [Chapitre 3, Création d'une image](#).

6. (Facultatif) Au terme de la compilation, vous pouvez consulter un fichier journal contenant un rapport sur le processus.

La sortie de compilation affiche l'emplacement des fichiers journaux.

Création d'une image

Après avoir configuré le fichier manifeste que vous avez choisi d'utiliser et, éventuellement, personnalisé les scripts de finalisation, vous êtes prêt à créer une image en exécutant la commande `distro_const`.

Vous pouvez créer une image en une seule étape par le biais de la commande `distro_const`. Vous pouvez également suspendre puis relancer la compilation afin d'examiner le contenu de l'image et de déboguer les scripts au cours du processus, le cas échéant.

Commande `distro_const`

La syntaxe complète de la commande `distro_const` est la suivante :

```
distro_const build [-v] [-r checkpoint] [-p checkpoint] [-l] manifest
```

Le tableau suivant répertorie les options de la commande `distro_const`.

TABLERAU 3-1 Options de la commande `distro_const`

Options de commande	Description
<code>distro_const build <i>Manifeste</i></code>	Crée une image en une seule étape à l'aide du fichier manifeste spécifié
<code>distro_const build -v <i>Manifeste</i></code>	Mode détaillé
<code>distro_const build -l <i>Manifeste</i></code>	Répertorie tous les points de contrôle valides à partir desquels vous pouvez suspendre et reprendre la compilation d'une image
<code>distro_const build -p <i>Point de contrôle Manifeste</i></code>	Suspend la compilation d'une image à un point de contrôle spécifié
<code>distro_const build -r <i>Point de contrôle Manifeste</i></code>	Reprend la compilation d'une image au niveau d'un point de contrôle spécifié
<code>distro_const build -h</code>	Affiche l'aide de la commande

Remarque - Vous devez prendre le rôle `root` pour exécuter la commande `distro_const`.

▼ Création d'une image en une seule étape

1. **Connectez-vous en tant qu'administrateur.**

Pour plus d'informations, reportez-vous à la section “ [A l'aide de vos droits administratifs attribués](#) ” du manuel “ [Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2](#) ”.

2. **Téléchargez le package `distribution-creator`.**

3. **Sélectionnez un fichier manifeste adapté à votre image.**

4. **(Facultatif) Si nécessaire, personnalisez le fichier manifeste en ajoutant les références aux scripts personnalisés.**

5. **Exécutez la commande de base `distro_const` sans options.**

```
# distro_const build manifest.xml
```

Remplacez *manifest* par le nom du fichier manifeste à utiliser comme empreinte pour votre image.

Par exemple :

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

Le constructeur de distribution extrait les packages requis et génère l'image selon les spécifications définies dans le fichier manifeste.

6. **(Facultatif) Au terme de la compilation, vous pouvez consulter un fichier journal contenant un rapport sur le processus.**

La sortie de compilation affiche l'emplacement des fichiers journaux.

▼ Création d'une image en plusieurs étapes

Vous pouvez utiliser les options offertes par la commande `distro_const` pour arrêter et redémarrer le processus de génération d'image au niveau de différents points de contrôle, afin de vérifier la sélection de fichiers, packages et scripts, et de suivre une procédure de débogage le cas échéant.

1. **Connectez-vous en tant qu'administrateur.**

Pour plus d'informations, reportez-vous à la section “ [A l'aide de vos droits administratifs attribués](#) ” du manuel “ [Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2](#) ”.

2. **Téléchargez le package `distribution-creator`.**
3. **Sélectionnez un fichier manifeste adapté à votre image.**
4. **(Facultatif) Si nécessaire, personnalisez le fichier manifeste en ajoutant les références aux scripts personnalisés.**
5. **Vérifiez les points de contrôle valides au niveau desquels vous pouvez choisir de suspendre ou relancer la compilation.**

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

Cette commande affiche les points de contrôle valides au niveau desquels vous pouvez interrompre ou reprendre la création d'une image. Utilisez les noms de points de contrôle fournis par cette commande comme valeurs correctes pour les autres options de commande de création de points de contrôle.

Par exemple, la commande suivante confirme les points de contrôle disponibles pour un fichier manifeste nommé `dc_livecd.xml`.

```
# distro_const build -l /usr/share/distro_const/dc_livecd.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
boot-setup		Setup LiveCD boot menu
pkg-img-mod		Package image area modifications
create-iso		ISO image creation
create-usb		USB image creation

Remarque - Dans cet exemple de sortie de commande, une marque X dans le champ de reprise indique que vous pouvez relancer la création à partir de ce point de contrôle.

6. **Créez l'image et suspendez la compilation au niveau du point de contrôle spécifié.**

```
# distro_const build -p checkpoint manifest
```

Par exemple, la commande suivante lance la compilation d'une image et l'interrompt avant que `im-mod` ne modifie la zone d'image :

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

7. **Reprenez la compilation de l'image à partir d'un point de contrôle spécifié.**

```
# distro_const build -r checkpoint manifest
```

Remarque - Le point de contrôle spécifié doit correspondre au point au niveau duquel la compilation antérieure a été interrompue ou à un point de contrôle précédent. Un point de contrôle ultérieur n'est pas valide.

Par exemple, la commande suivante permet de reprendre la compilation de l'image à l'étape ba-arch.

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

Remarque - Vous pouvez combiner les options de suspension et de reprise dans une commande build.

8. **(Facultatif) Au terme de la compilation, vous pouvez consulter un fichier journal contenant un rapport sur le processus.**

La sortie de compilation affiche l'emplacement des fichiers journaux.

Index

A

- Archive root
 - Différences SPARC et x86, 10
- Attribution d'un nom
 - Image d'installation, 14
 - Points de contrôle, 24

B

- boot_entry, élément de fichier manifeste, 15
- boot_mods, élément de fichier manifeste, 15

C

- Cible Élément de fichier manifeste, 15
- Configuration système requise pour créer des images, 11
- Constructeur de distribution
 - Présentation, 7
- Création d'images d'installation, 13
 - en une étape, 28
 - Images d'installation en plusieurs étapes, 28
 - Présentation, 27

D

- Désinstallation de packages, 18
- distro_name, élément de fichier manifeste, 14
- distro_const , commande
 - Syntaxe et options, 27
- distro_const, commande
 - Utilisation pour créer une image en plusieurs étapes, 28

E

- Editeur
 - Ajout à l'image d'installation, 17
 - Modifications de systèmes installés, 19
 - Spécifier pour les packages, 16
- Eléments de fichier manifeste
 - boot_entry, 15
 - boot_mods, 15
 - Cible, 15
 - distro_name, 14
 - Exécution, 19
 - Liste de, 14
 - Modification
 - Archive root, 10
 - Points de contrôle de compilation, 19
 - Titre de l'image, 14
 - Zone de compilation, 15
 - Modifications
 - Editeur pour les systèmes installés, 19
 - Spécifier l'éditeur à utiliser pendant la création, 16
 - Source, 16
- Eléments de manifeste
 - Modification
 - Liste des packages, 16
 - Menu d'initialisation, 15
- Exécution Élément de fichier manifeste, 19
- Exemples de fichiers manifestes, 12

F

- Fichiers manifeste
 - Personnalisation, 14
- Fichiers manifestes
 - Définition, 9
 - Exemple, 12

I

Images d'installation

- Ajout d'éditeurs supplémentaires à, 17
- Ajout de packages à, 17
- Ajout des packages SVR4 à, 22
- Attribution d'un nom, 14
- Création, 13, 27
 - Configuration système requise, 11
 - En plusieurs étapes, 28
 - en une étape, 28
 - Présentation, 9
- Personnalisation, 12, 23
- Types Oracle Solaris, 8
 - Différences entre ISO et USB, 7
 - ISO pour installation en mode texte, 8
 - ISO pour installation Live Media, 8
- Types Oracle Solaris
 - ISO pour installations automatisées, 8
 - USB, 7

Images d'installation USB, 7

Images ISO, 8

Installation automatisée

- Création d'une image ISO, 8

Installation en mode texte

- Création d'une image ISO, 8

Installation Live Media

- Création d'une image ISO, 8

M

Menu d'initialisation

- Personnalisation, 15

P

Package de groupe

- Définition de, 18

Package de groupes

- Omission de packages dans, 18

Packages

- A installer, 16
- Ajout à l'image d'installation, 17
- Désinstallation, 18

Packages SVR4

- Ajout à une image d'installation, 22

Périphériques de mémoire flash, images d'installation USB, 7

Personnalisation

- Fichiers manifeste, 14
- Images d'installation à l'aide des fichiers manifeste, 12
- Images d'installation à l'aide des scripts, 23
- Menu d'initialisation, 15

Points de contrôle

- Ajout, 19
- Attribution d'un nom, 24
- Champs, 21
- Définition, 9
- Scripts personnalisés et, 21
- Utilisation pour créer une image en plusieurs étapes, 28
- Utilisation pour installer les packages SVR4, 22
- Utilisation pour référencer les scripts personnalisés au cours d'une création, 23

S

Scripts *Voir* Scripts personnalisés

Scripts personnalisés

- Création et utilisation, 23
- Points de contrôle et, 21

Source Elément de fichier manifeste, 16