

Gestion des services système dans Oracle® Solaris 11.2



Référence: E53839
Juillet 2014

Copyright © 2013, 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	13
 1 Introduction à l'utilitaire de gestion des services	15
Capacités SMF	15
Nouvelles fonctionnalités de cette version	16
Concepts et composants SMF	17
Service SMF	18
Modèles de service	20
Noms des services	20
Etats des services	21
Dépendances de service	21
Agents de redémarrage du service	22
Propriétés et groupes de propriétés du service	23
Référentiel de configuration de service	24
Fichiers de configuration et services SMF	28
Privilèges de gestion de service	30
 2 Obtention d'informations sur les services	31
Création de listes de services sur le système	31
Affichage de l'état de service	31
Présentation d'informations supplémentaires à propos des services	32
Affichage des informations sur le service sélectionné	34
Affichage des dépendances de service	34
Regroupements de dépendance	35
Présentation des instances dont un service dépend	36
Présentation des instances dépendant d'un service	37
Redémarrage automatique d'un service	38
Plus d'informations sur les états de service	39
Affichage des fichiers journaux d'un service	41
Inspection de la configuration du service	42

Affichage des descriptions de propriétés et de groupes de propriétés	42
Affichage des valeurs de propriété d'instance et de services	43
Affichage des propriétés dans un type de groupe de propriétés	46
Affichage de la couche de définition de la valeur	47
Affichage des valeurs d'un instantané particulier	48
Affichage des personnalisations de configuration	49
Affichage des paramètres de notification d'événement	49
3 Administration de services	53
Gestion des instances de service SMF	53
Démarrage d'un service	53
Arrêt d'un service	56
Redémarrage d'un service	58
Relecture de la configuration d'un service	59
Suppression d'un service	60
Configuration la notification des événements de transition d'état et FMA	62
4 Configuration des services	65
Utilisation de la commande de configuration de service	65
Appel d'un éditeur de propriétés	66
Appel de svccfg de manière interactive ou avec un fichier	67
Définition des valeurs de propriété	68
▼ Modification d'une valeur de propriété ttymon	70
▼ Procédure de modification d'une variable d'environnement pour un service de processus d'environnement	72
Ajout de groupes, des propriétés et valeurs de propriété	73
Suppression de groupes de propriétés, de propriétés et de valeurs de propriétés	75
Suppression de la configuration de l'administration	75
Suppression de la configuration non administrative	77
Ajout d'instances de service	79
Rétablissement des instantanés	79
Importation et application de manifestes et de profils	80
Configuration de plusieurs systèmes	80
▼ Création d'un profil à l'aide de svcbundle	81
▼ Création d'un profil à l'aide de svccfg	82
Modification des services contrôlés par inetd	82
▼ Modification d'une valeur de propriété pour un service contrôlé par inetd	83
Modification des services configurés par un fichier	84

5 Utilisation de SMF pour contrôler votre application	87
Création d'un service SMF	87
▼ Procédure de création d'un service SMF à l'aide de l'outil de génération de bundles de services	88
Création d'un service de démarrage ou d'arrêt d'une instance de base de données Oracle	89
Services de noms, instances, groupes de propriétés et propriétés	96
▼ Procédure de conversion d'un script de contrôle d'exécution en service SMF	97
Création d'un fichier de configuration à l'aide d'un pochoir	98
▼ Création d'un service de pochoir	98
Exemple de service de pochoir dans Oracle Solaris	99
 A Pratiques recommandées dans le cadre de SMF et du dépistage des pannes	105
Pratiques recommandées dans le cadre de SMF	105
Dépistage des pannes de services	106
Compréhension des modifications de la configuration	106
Réparation d'une instance endommagée, hors ligne ou en cours de maintenance	107
Marquage d'une instance comme dégradée ou en maintenance	111
Diagnostic et réparation des problèmes de référentiel	111
Spécification de la quantité de messages au démarrage	114
Spécification du jalon SMF sur lequel initialiser	115
Utilisation de SMF pour rechercher les problèmes d'initialisation du système	117
Conversion des services inetd en services SMF	119
 Index	121

Liste des figures

FIGURE 1-1	Utilitaire de gestion des services Framework	18
FIGURE 2-1	Les relations de dépendance de service	35

Liste des tableaux

TABLEAU 2-1	Après une automatiquement de dépendance Redémarrage d'une Arrêter service	39
TABLEAU A-1	Niveaux de journalisation des messages de démarrage de SMF	115
TABLEAU A-2	Jalons d'initialisation SMF et niveaux d'exécution correspondant	116

Liste des exemples

EXEMPLE 2-1	Etablissement de la liste de tous les services Activé	31
EXEMPLE 2-2	Etablissement de la liste de tous les services installés	32
EXEMPLE 2-3	Service d'une liste de toutes les instances d'une	32
EXEMPLE 2-4	Processus démarrés par présentant un contrat de service	33
EXEMPLE 2-5	Redémarrage d'automatiquement présentant après un contrat de service Arrêt du processus	33
EXEMPLE 2-6	Actuellement l'établissement de la liste de l'instance en cours d'utilisation et les propriétés héritées	43
EXEMPLE 2-7	Groupes de propriétés liste spécifié par l'actuellement Propriétés ou en cours d'utilisation	44
EXEMPLE 2-8	Les valeurs d'instance d'répertoriant dans le Service et Modification de la vue	44
EXEMPLE 2-9	Groupes de propriétés liste spécifiée dans le Propriétés ou Modification de la vue	45
EXEMPLE 2-10	Groupes de propriétés et leurs types présentant	46
EXEMPLE 2-11	Un groupe de propriétés répertoriant Type Properties of a	46
EXEMPLE 3-1	Activation définitivement d'une instance de service	55
EXEMPLE 3-2	Activation temporaire d'instance de service	55
EXEMPLE 3-3	Désactivation d'une instance de service	58
EXEMPLE 3-4	Configuration d'une notification générale pour un événement d'état de service	62
EXEMPLE 3-5	Configuration d'une notification pour une instance de service spécifiée	63
EXEMPLE 3-6	Configuration d'une notification pour un événement FMA	63
EXEMPLE 3-7	Suppression de paramètres de notification	63
EXEMPLE 4-1	La définition d'une valeur simple	68
EXEMPLE 4-2	Une Valeur qui contient un paramètre deux points de caractères	69
EXEMPLE 4-3	Embedded une Valeur qui contient la définition de Spaces	69
EXEMPLE 4-4	Une valeur que l'option Est la définition d'un jeu de valeurs (LOV)	69
EXEMPLE 4-5	Ajout d'une valeur	70
EXEMPLE 4-6	Utilisation de addpg pour créer un nouveau groupe de propriétés	73
EXEMPLE 4-7	Utilisation de setprop pour créer une nouvelle propriété	74

EXEMPLE 4-8	Utilisation d'addpropvalue pour créer une nouvelle propriété	74
EXEMPLE 4-9	Propriétés de la suppression de toutes les valeurs d'un	75
EXEMPLE 4-10	De la suppression d'une propriété Toutes les valeurs concordantes	76
EXEMPLE 4-11	Suppression d'une propriété	76
EXEMPLE 4-12	La suppression d'un groupe de propriétés	76
EXEMPLE 4-13	Suppression de personnalisations	77
EXEMPLE 4-14	Prise en charge la suppression d'un lot d'offres configuration qui a	78
EXEMPLE 4-15	Annulation de masquage Configuration	78
EXEMPLE 4-16	Installation automatique d'un profil à l'aide de svcbundle	82
EXEMPLE 4-17	Modification de la commande à exécuter au démarrage d'un service contrôlé par inetd	84
EXEMPLE 5-1	Installation automatique d'un manifeste généré	89

Utilisation de la présente documentation

- **Présentation** : décrit comment utiliser la fonction SMF (Service Management Facility, utilitaire de gestion des services) d'Oracle Solaris. SMF est l'un des composants de l'autorétablissement prédictif Oracle Solaris le plus étendu.
- **Public visé** : administrateurs système gérant les services système
- **Connaissances requises** : expérience avec l'administration des systèmes Oracle Solaris

Bibliothèque de la documentation des produits

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 l'utilitaire de gestion des services

Le Oracle Solaris SMF (Service Management Facility, utilitaire de système et de coordination de reconfiguration), structure permet de gérer des services d'application. Les services SMF dont on doit absolument disposer de gérer le système critiques du système et de travail permet de gérer opération des services d'application tel qu'une base de données ou sur le serveur Web. Améliore la disponibilité d'un SMF système, vous devez vous assurer que les services du système et des applications essentielles s'exécuter en continu, même dans le cas de pannes matérielles ou logicielles.

SMF remplace les fichiers de configuration pour la gestion de services ; c'est le mécanisme recommandé pour démarrer les applications. SMF remplace le mécanisme de démarrage par script `init`, les configurations `inetd.conf` et la plupart des scripts `rc?.d`. SMF conserve la compatibilité des pratiques administratives existantes, à chaque fois que cela est possible. Par exemple, la plupart des scripts `rc` fournis par les clients et les éditeurs de logiciels fonctionnent toujours de la même manière que sans SMF.

Capacités SMF

La structure SMF est actif en permanence sur un système Oracle Solaris 11. SMF offre les fonctionnalités suivantes :

- D'initialisation plus rapidement. SMF accélère l'initialisation des systèmes de grande taille grâce au démarrage des services en parallèle en fonction des dépendances des services.
- Echec du redémarrage des services. Services ont bien définis SMF des relations de dépendance avec d'autres services. Si un service signale les échoue, les services dépendants, SMF. Tente automatiquement de redémarrer SMF des services après échec par ordre de dépendance.
- Inspectez les services. Afficher les relations entre services et processus. Voir les valeurs de propriétés de service.
- Gérer les services. Activer, désactiver et de redémarrer les services. Ces modifications peuvent être conservées après les mises à niveau et les réinitialisations, ou vous pouvez spécifier des modifications temporaires.
- Configurer les services.
 - Modifier les valeurs des propriétés de service.

- Ajouter et supprimer des propriétés personnalisées.
- Service d'audit des modifications. Solaris SMF écrit les enregistrements d'audit pour chaque modification d'administration à un service ou ses propriétés. SMF valeur de la propriété est possible d'afficher ou de service si un état a été défini par un administrateur.
- Déléguer les tâches de manière sécurisée aux utilisateurs non-root , y compris la possibilité de modifier les propriétés, et d'activer, désactiver ou redémarrer les services.
- Créer des services. Créer facilement une nouvelle instance d'un service existant, copiez-les et modifiez un service existant, ou utiliser un service des outils de création.
- Des problèmes liés au service de débogage. Afficher une explication indiquant la raison pour laquelle facilement un service activé n'est pas en cours d'exécution ou le motif pour lequel un service empêche un autre service dans en cours d'exécution.
- Configurer la manière dont vous recevrez une notification de particulière. les événements du logiciel ou les pannes de matériel
- Conversion de configurations `inetd.conf` en services SMF.
- Les propriétés du service pour convertir les fichiers de configuration SMF. Ce mécanisme offre une passerelle pour les services qui sont gérés par les applications qui SMF mais nécessitent tout de même interagir avec les fichiers de configuration.

Nouvelles fonctionnalités de cette version

Nouvelles fonctionnalités relatives à ce qui suit SMF dans cette version :

Opérations de redémarrage et d'actualisation synchrones

Une nouvelle option `-s` des sous-commandes `restart` et `refresh` de la commande `svcadm` spécifie une opération synchrone similaire à celle de l'option `-s` actuelle pour les sous-commandes `enable` et `disable`.

Afficher les fichiers journaux

Une nouvelle option `-L` pour la commande `svcs` contient l'intégralité du fichier journal, les dernières lignes du fichier journal ou le chemin d'accès complet du fichier journal de l'instance de service spécifiée.

Opérations par lot sur un ensemble d'instances de service

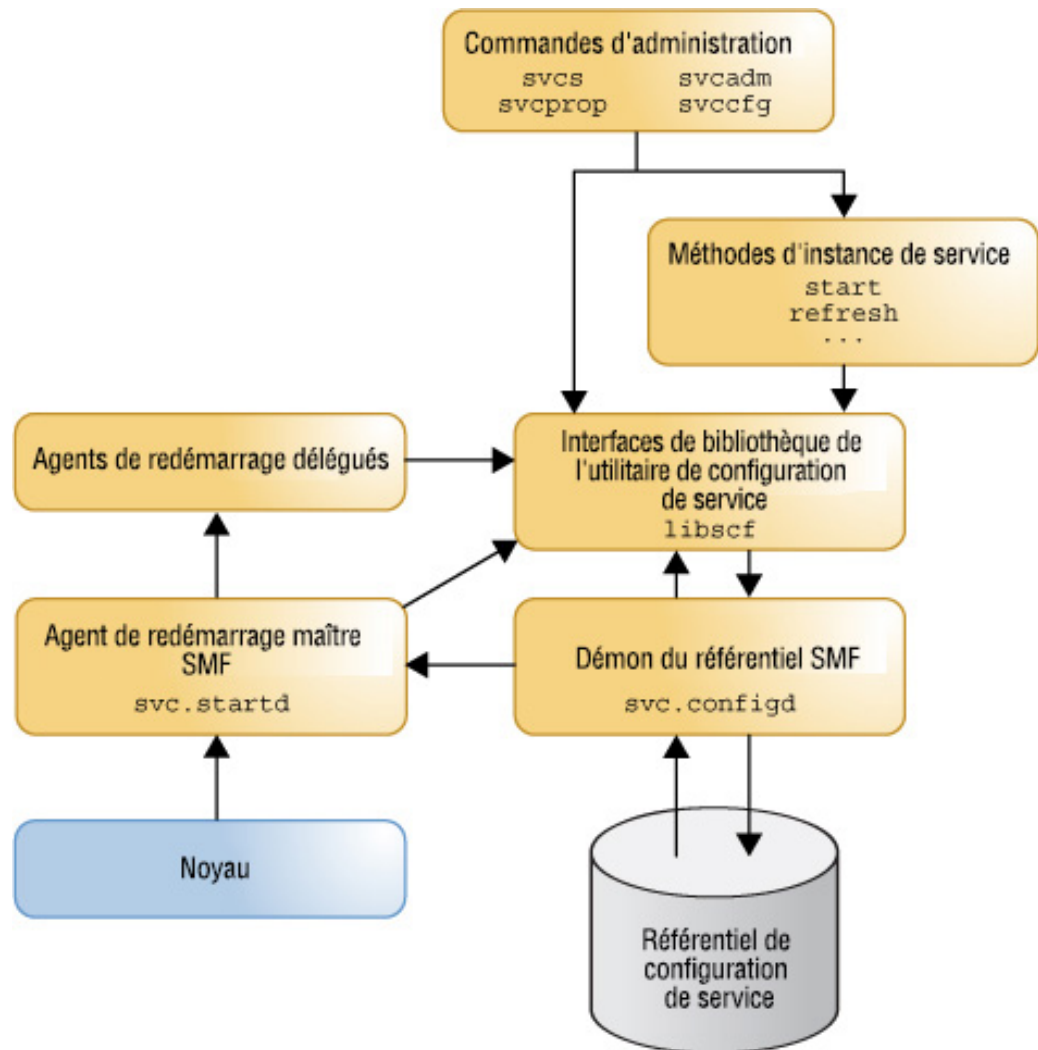
Les nouveaux appels de fonction `libscf(3LIB)` permettent de regrouper les opérations administratives courantes et de les exécuter par lots, de manière synchrone ou non. Vous pouvez activer et désactiver la fonction, cochez la case Terminé, puis, redémarrez, actualiser ou ressources dégradés, effacez le département et de maintenance transition entre les jalons d'un ensemble d'instances de service.

Conversion de propriétés du service SMF en fichiers de configuration

Pour les services qui sont gérés par SMF qui mais nécessitent tout de même interagissent avec les applications, un nouveau fichier de configuration est créé et mis à jour un stenciling de capacité dans les valeurs de propriété définies fichier de configuration pour le service dans le référentiel de configuration de service. Stencil un nouveau fichier appelé définir les structures de votre fichier de la contient le fichier de configuration qui sera créé par SMF. Un nouvel utilitaire `svcio` génère le fichier de configuration à partir des définitions du fichier pochoir et des propriétés du service SMF. Cette fonctionnalité permet de bénéficier de modification de gestion de la configuration avec un minimum de SMF pour l'application existante.

Concepts et composants SMF

Cette section définit les termes utilisés dans le reste du guide. La figure suivante illustre la structure SMF principaux composants du. Lorsque vous initialisez une image, les mises à jour le référentiel de configuration de service SMF si nécessaire, lit les données du référentiel, puis il démarre ce service instances qui sont activées. Dans la figure suivante, `libscf` est l'interface de bibliothèque utilisée par les agents de redémarrage pour interagir avec le référentiel de configuration de service. L'interaction entre le référentiel de configuration de service et les interfaces de bibliothèque `libscf` est gérée par le démon `svc.configd`. Les commandes `svcs`, `svcprop`, `svcadm` et `svccfg` constituent l'interface utilisé par les administrateurs pour interagir avec le référentiel de configuration de service.

FIGURE 1-1 Utilitaire de gestion des services Framework

Service SMF

Un *service* SMF est une application qui s'exécute de manière permanente et représente une entité système telle que les suivantes :

- Les services applicatifs telle qu'une base de données ou un serveur Web
- Essentielles des services système
- Le logiciel état d'un périphérique
- Informations sur la configuration du noyau
- Des jalons correspondant à l'état init du système

Une *instance de service* est un enfant d'un service et fournit des fonctionnalités et des relations de dépendance aux applications et aux autres instances de service. Seules les instances ont un état et peut être démarré et arrêté. Si une instance échoue pour quelque raison que ce soit, par exemple, une panne matérielle ou logicielle détecte l'erreur et SMF redémarre automatiquement instance ainsi que toutes dépendants les instances.

Autoriser plusieurs instances des configurations d'un service d'un service de s'exécuter simultanément. Le nombre d'instances du service de configuration et de personnaliser de service communs héritent. Par exemple, vous pouvez définir un service à l'aide d'une seule instance de serveur Web configuré pour écouter sur le port 80 et une autre instance configuré pour écouter sur le port 1008. La plupart des services ont une instance default. Ne pas certains services, tels que certains contiennent déjà des instances, SMF pour stocker les services utilisant mais pas d'exécuter des programmes des configurations. Par exemple, le service x11/x11-server ne dispose d'aucune instance.

Un service SMF est décrit dans un fichier appelé un *manifeste* de service. Paramétrage des instances de service, le fichier manifeste décrit certaines d'entre elles, les propriétés de configuration, et les méthodes. Les *méthodes* de service peuvent démarrer, arrêter et actualiser des instances de service. Une méthode peut être un démon ou d'autres binaire exécutable, un script exécutable. Un fichier de *profil* de service permet de personnaliser n'importe quel service, principalement en ajoutant des propriétés et en ajoutant et remplaçant les valeurs des propriétés. Les nouvelles propriétés et valeurs sont superposées aux valeurs attribuées dans le manifeste, comme décrit dans [“Référentiel Couches” à la page 26](#). Reportez-vous à la section [“Les lots d'offres de service” à la page 25](#) pour plus information sur les manifestes et les profils. Un profil est aussi un excellent outil pour appliquer la même configuration personnalisée à plusieurs systèmes, comme décrit dans [“Configuration de plusieurs systèmes” à la page 80](#).

Les informations de service sont stockées dans le *référentiel de configuration de service*, aussi appelé *base de données SMF*. Le référentiel de configuration de service stocke l'état actuel de chaque instance de service sur le système et les données de configuration pour chaque service et de l'instance de service. Les données sont stockées dans les *couches* en fonction de la manière dont les valeurs ont été modifiées, comme décrit dans [“Référentiel Couches” à la page 26](#).

SMF fournit des *actions* que vous pouvez appeler sur une instance de service, dont les actions activer, désactiver, actualiser, et redémarrer. Chaque instance de service est gérée par un *agent de redémarrage*, qui effectue ces actions d'administration. En général, les agents de redémarrage réalisent les actions en exécutant des *méthodes* pour faire passer l'instance de service d'un état à un autre. Pour plus d'informations sur les agents de redémarrage, reportez-vous à [“Agents de redémarrage du service” à la page 22](#)

Un *service de jalon* est un type de service particulier qui représente un niveau de préparation du système. Un jalon se compose d'un service qui les autres instances de service dépendent de données puisse démarrer. Par exemple, les niveaux d'exécution sont représentés par des services de jalon tels que `svc:/milestone/multi-user-server`. Les jalons peuvent également être utilisés pour indiquer l'état de préparation d'un groupe de services, tel que `svc:/milestone/devices`, `svc:/milestone/network` ou `svc:/milestone/name-services`.

Modèles de service

L'un des services SMF trois sont des modèles, procédez comme suit :

Service transitoire

Le service réalise certaines tâches et s'arrête, sans démarrer de processus à longue durée d'exécution.

Service enfant ou d'attente

Le service est redémarré une interruption en toute sécurité chaque fois que son arrêt du processus enfant. Réseau qui quitterait le réseau une interruption en toute sécurité d'un processus fils qui n'est pas considérée comme une erreur.

Service de contrat ou démon

Le service démarre un démon à longue durée d'exécution ou plusieurs processus apparentés reliés entre eux par un *contrat de service*. Le contrat de service et gère des processus du mois que sa date de début ainsi que leur date de début n'importe quel ordre les services qui en dépendent. Vous n'avez besoin pour gérer le service de haut niveau.

Noms des services

Chaque service et instance de service est représentée par un FMRI (Fault Management Resource Identifiers, identificateurs). Le FMRI complet pour une instance de service a le format suivant :

`svc:/service_name:instance_name`

Le *service_name* est un nom hiérarchique, tel que `network/dns/client` ou `application/pkg/server`. Les composants du *service_name* qui précèdent la barre oblique finale (/) constituent la *catégorie* du service. La catégorie, par exemple `application`, `device`, `milestone`, `network` ou `system` aide à identifier l'usage du service.

La catégorie `site` est réservée pour vous aider à éviter les conflits de noms lorsque vous créez vos propres services SMF. Par exemple, un service spécifique au site nommé `svc:/site/tool` n'entrera pas en conflit avec un service Oracle Solaris nommé `svc:/tool`.

Noms d'instance de service sont ajoutés à la fin de service parent. nom une fois que le signe deux-points Par exemple, `svc:/system/identity:node` et `svc:/system/identity:domain` sont des instances du service `svc:/system/identity`.

Dans les scripts, nous vous recommandons d'utiliser le nom d'instance de service complet. En mode interactif, les noms peuvent être raccourci à la plus à droite des parties du nom qui ont pour résultat un nom unique. Par exemple, `svc:/system/identity` peut être abrégé en `identity` et `svc:/system/identity:domain` en `identity:domain`. Les noms d'instance doit être précédée d'un composant du nom de service, suivi du symbole.

Etats des services

A une heure donnée, est une instance de service SMF dans l'un des états suivants :

- `degraded` – L'instance est en cours d'exécution ou capable de s'exécuter, mais fonctionne à capacité limitée.
- `disabled` – L'instance n'est pas activée et n'est pas en cours d'exécution ni capable de s'exécuter.
- `maintenance` – L'instance est activée, mais incapable de s'exécuter. L'instance peut passer par l'état `maintenance` parce qu'une action administrative n'est pas encore terminée. Dans le cas contraire, une action d'administration est obligatoire pour résoudre le problème.
- `offline` – L'instance est activée, mais pas en cours d'exécution, ou incapable de s'exécuter. Par exemple, si les dépendances d'un service activé ne sont pas satisfaites, le service est conservé dans l'état `offline`.
- `online` – L'instance est activée et en cours d'exécution, ou capable de s'exécuter. L'état `online` est l'état de fonctionnement attendu pour une instance de service correctement configurée et dont toutes les dépendances sont satisfaites.
- `uninitialized` – Cet état est l'état initial pour tous les services.

Une instance de service passe d'un état à l'autre en fonction de conditions telles que les actions administratives ou l'état des services dépendant d'elle. Par exemple, lorsque vous activez une instance qui se trouvait dans l'état `disabled`, elle passe d'abord à l'état `offline`, puis à l'état `online` une fois que toutes ses dépendances sont satisfaites. Reportez-vous à la page de manuel [smf\(5\)](#) pour plus d'informations sur ces états de service et sur la façon dont les instances de service changent d'état.

Dépendances de service

Un service peut avoir une *dépendance* par rapport à un service, une instance de service ou un fichier. Définir des relations entre les services de dépendances de service.

Déterminer le moment où un service des relations de dépendance est lancée et automatiquement arrêté. Lorsque les dépendances d'un service ne sont pas satisfaites, le service se trouve dans l'état `offline`. Les dépendances d'un service compatible lorsque n'est satisfaite, le service est démarré. Si le service démarre avec succès, il passe à l'état `online`.

Les services sont validés de nouveau en tant que dépendances de service passent par différents états. Dépendances qui sont satisfaites de service peuvent ensuite devenir non satisfaites. Dépendances de fichier sont évaluées une seule fois.

Les dépendances peuvent être obligatoires ou facultatives. Dépendances de service peut être nécessaire pour être en cours d'exécution ou désactivé. Une personne à charge service peut être configuré de manière à redémarrer ou non lorsque l'un de ses dépendances de service est arrêté ou que vous l'avez actualisée.

Autoriser les éléments suivants des relations de dépendance une fonction, procédez comme suit :

- Processus d'initialisation évolutifs et reproductibles
- Plus rapide du système parallèle de démarrage des fonctions sur des ordinateurs possédant des services en parallèle en démarrant indépendant
- En matière de récupération d'erreurs, le confinement précis des erreurs en redémarrant uniquement et services directement affectés par une erreur, et de redémarrer ces services avec le bon ordre de dépendance

Agents de redémarrage du service

Chaque instance de service SMF est gérée par un *agent de redémarrage*. Configuration d'instance et l'agent de redémarrage extrait fournit un environnement d'exécution. Reportez-vous à la page de manuel [smf_restarter\(5\)](#) pour des informations communes à tous les agents de redémarrage.

Démon d'agent de redémarrage principal

Le démon `svc.startd` est le démon d'*agent de redémarrage principal* pour SMF et l'*agent de redémarrage principal* pour toutes les instances de service. Le démon `svc.startd` gère les états pour toutes les instances de service et leurs dépendances. Instances en tant que dépendances est satisfait lorsque passer à l'état `online` (en ligne), l'agent de redémarrage maître des autres instances appelle les méthodes de démarrage ou dirige l'agent de redémarrage délégué pour appeler la méthode `start`. L'agent de redémarrage maître arrête un service de l'instance lors de l'instance les dépendances ne sont plus respectées. L'agent de redémarrage tente de redémarrer une instance en cas d'échec de l'instance. Ne peut pas être en ligne : une instance tant que toutes ses dépendances sont vérifiées, les dépendances d'une instance de vous aider à déterminer

le comportement de redémarrage de l'instance. Déclaration propriétés définies dans chaque dépendance requis définir si cette dépendance est instance et dans quel cas, il sera redémarrée si la dépendance est redémarré.

Entre autres tâches, le démon `svc.startd` démarre les scripts `/etc/rc*.d` adéquats aux niveaux d'exécution appropriés, ce qui était auparavant du ressort de `init`.

L'exemple suivant montre que `svc.startd` est l'agent de redémarrage de l'instance de service `network/impd:default`. Associée n'a pas été omise que d'autres types de sorties à partir de cet exemple.

```
$ svcs -l impd:default
restarter    svc:/system/svc/restarter:default
```

Si la propriété `restarter` est vide ou définie sur `svc:/system/svc/restarter:default`, l'instance de service est gérée par `svc.startd`. Pour plus d'informations sur le démon `svc.startd`, reportez-vous à la page de manuel [svc.startd\(1M\)](#).

Redémarrage Agents de redémarrage délégués

Certains services présentent un ensemble de comportements communs au démarrage. Un *agent de redémarrage délégué* peut fournir un environnement d'exécution spécifique et un comportement de redémarrage spécifique à l'application pour ces services.

`inetd` est un exemple d'agent de redémarrage délégué, pouvant démarrer les services Internet à la demande pour éviter qu'ils soient constamment en cours d'exécution. L'agent de redémarrage `inetd` fournit à ses instances de service un environnement composé d'une connexion réseau en entrée et de descripteurs de fichiers en sortie. Pour plus d'informations sur le démon `inetd`, reportez-vous à la page de manuel [inetd\(1M\)](#). L'exemple suivant montre qu'`inetd` est l'agent de redémarrage de l'instance de service `cups/in-lpd:default`. Associée n'a pas été omise que d'autres types de sorties à partir de cet exemple.

```
$ svcs -l cups/in-lpd:default
restarter    svc:/network/inetd:default
```

L'agent de redémarrage délégué propriété spécifié par la propriété `restarter` est responsable de la gestion de l'instance de service dès qu'il est disponible.

Propriétés et groupes de propriétés du service

Les informations sur les services, y compris les dépendances, les méthodes, l'état et les données d'application, sont enregistrées dans le référentiel de configuration de service sous forme

d'ensemble de *propriétés*. Les propriétés peuvent être définies sur le service ou une instance du service. Les propriétés définies sur le service sont héritées par toutes les instances de ce service. Les propriétés qui sont définies sur une instance sont uniquement utilisées par cette instance. Le nombre d'instances du service hérités ont la possibilité de personnaliser les valeurs des propriétés et pouvez définir d'autres propriétés qui ne sont pas définies pour le service parent.

Les propriétés sont organisées en *groupes de propriétés*. Certaines propriétés courant aux groupes sont les suivantes :

- `general` – Contient des informations telles que si l'instance est activée
- `restarter` – Contient des informations d'exécution stockées par l'agent de redémarrage pour le service, y compris l'état actuel de l'instance
- `start`, `refresh`, `stop` – Contiennent des informations telles que le programme à exécuter pour démarrer, actualiser ou arrêter le service
- `config` – Utilisé par les développeurs de services pour conserver les données d'application.

Reportez-vous à la page de manuel [smf\(5\)](#) pour plus d'informations sur les propriétés et les groupes de propriétés.

Référentiel de configuration de service

Les informations sur chaque service sont stockées dans le *référentiel de configuration de service*, aussi appelé *base de données SMF*. Le référentiel de configuration de service stocke des informations telles que l'état actuel de chaque instance de service sur le système, ainsi que les propriétés de chaque service et de l'instance de service.

Le référentiel stocke des informations de configuration permanentes, ainsi que les données d'exécution pour les services SMF.

- Les informations de configuration permanentes sont stockées dans les couches en fonction de la source des données. Reportez-vous à “[Référentiel Couches](#)” à la page 26.
- Les données d'exécution, ou informations de configuration non permanentes, ne sont pas conservées après la réinitialisation et le référentiel ne stocke pas les informations de couche des données non permanentes. En général, les groupes de propriétés non permanents définissent un état de programme actif.

Le référentiel stocke également des données de modèle de service tels que les types, les contraintes de valeurs et les descriptions des propriétés. Modèle de données sont définies dans le manifeste du service. Reportez-vous à la page de manuel [smf_template\(5\)](#) pour plus d'informations sur les données modèles.

Le référentiel de configuration de service peut uniquement être manipulé ou interrogé à l'aide des interfaces SMF. Utilisez les commandes `svcs`, `svccprop`, `svcadm` et `svccfg` ou les fonctions de bibliothèque de l'Utilitaire de configuration de services répertoriées à la page de

manuel [libscf\(3LIB\)](#). Vous pouvez lire et écrire la propriété des valeurs et d'afficher les valeurs de propriété de couches spécifiées et aux instantanés. Pour plus d'informations sur les couches, reportez-vous à [“Référentiel Couches” à la page 26](#). Pour plus d'informations sur les clichés, reportez-vous à la section [“Référentiel Clichés” à la page 27](#). Vous pouvez n'afficher que les propriétés de l'instance de service ou du service parent sélectionné, ou un affichage *composé* des propriétés. Dans une vue composée, les deux propriétés définies dans le parent de service et de propriétés définies dans l'instance de service ne s'affiche. les valeurs indiquées sont les valeurs définies sur l'instance de service.

Les lots d'offres de service

Un service le groupe est un fichier contenant les informations XML qui est stocké dans le référentiel de configuration de service pour un service ou d'une instance de service. Lots d'offres informations fournies dans de service sont stockées dans le référentiel de configuration de service et peut être exporté à partir du référentiel. Lots d'offres dans des emplacements standard de service sont importés dans le référentiel au cours de l'initialisation du système.

Il existe deux types de service. les lots sont des manifestes et des profils

Manifestes	Contiennent l'ensemble des manifestes associées à d'autres propriétés à l'aide d'un ensemble de services ou instances de service.
Profils	La personnalisation d'un des profils de service fournissent généralement ou une instance de service qui augmente ou remplace les informations fournies dans le fichier manifest. Les exemples de personnalisations sont notamment les propriétés supplémentaires et les valeurs de propriété modifiée.

L'*emplacement standard* des manifestes est `/lib/svc/manifest`. L'emplacement standard des profils est `/etc/svc/profile`.

Lors de l'initialisation du système ou le manifeste du service d'import n'a pas été redémarré, les manifestes sont importés et tous les profils sont appliquées si elles ont été ajoutées ou améliorées. Un package IPS qui offre des fonctionnalités d'un lot d'offres peut indiquer que le service du service d'import doit être redémarré manifeste lorsque le package est installé.

Les personnalisations locales peuvent être fournies dans des fichiers de profil avec un suffixe `.xml` dans le répertoire `/etc/svc/profile/site`. Si la même propriété dans la couche de référentiel du même service ou de la même instance est définie par plusieurs manifestes ou profils, SMF ne peut pas déterminer la valeur à utiliser. Lorsque ce type de conflit est détecté, l'instance est placée en état de maintenance. Pour plus d'informations sur les couches, reportez-vous à [“Référentiel Couches” à la page 26](#).

Services particuliers en outre Oracle Solaris de transmettre les lots peuvent également mise à disposition, de service à travers une variété de configuration personnalisé de systèmes.

Un profil système, `/etc/svc/profile/generic.xml`, est appliqué au cours de l'installation. Ne modifiez pas ce profil. Toutes les modifications apportées à ce profil système seront écrasés lors de la mise à niveau. Pour plus d'informations, reportez-vous à la page de manuel [smf_bootstrap\(5\)](#).

Référentiel Couches

Le référentiel de configuration de service peut stocker des valeurs différentes pour une seule propriété. Le référentiel stocke les données dans les *couches* en fonction de leur source. Manifestes toutes les origines peuvent être des profils système, les profils de site, et les personnalisations apportées à l'aide, commandes et de bibliothèque Interface de SMF. Vous pouvez afficher les valeurs dans les différentes couches pour comprendre la source de la valeur dans la configuration en cours d'exécution a été affecté : si une valeur dans le fichier manifest, dans un profil, ou a été modifié par un administrateur.

Les modifications de configuration apportées à l'aide des commandes SMF et des interfaces de bibliothèque n'apparaissent que dans la couche `admin`. La configuration dans d'autres couches manifeste est défini dans le profil et des fichiers dans des emplacements standard. Lorsqu'une propriété est ajoutée dans le référentiel à partir d'un fichier, les informations relatives à cette propriété incluent le nom des fichiers en question.

Couche	Contenu
admin	Toutes les modifications qui ont été apportées à l'aide des commandes ou de bibliothèque Interface de SMF, par un administrateur ou par une application. La couche <code>admin</code> inclut également toutes les modifications été apportées par l'importation d'un profil ou l'application d'un manifeste à partir d'un emplacement non standard. Reportez-vous à la section “ Importation et application de manifestes et de profils ” à la page 80 pour des avertissements sur l'utilisation d'emplacements standard.
site-profile	Toutes les valeurs des fichiers de profil du répertoire <code>/etc/svc/profile/site</code> ou les profils hérités <code>/etc/svc/profile/site.xml</code> et <code>/var/svc/profile/site.xml</code> . Notez que <code>/var/svc/profile</code> est désapprouvé en tant qu'emplacement standard et ne doit pas être utilisé pour les nouveaux profils.
system-profile	Toutes les valeurs des profils système <code>/etc/svc/profile/generic.xml</code> et <code>/etc/svc/profile/platform.xml</code> .
manifest	Les valeurs des manifestes dans les répertoires <code>/lib/svc/manifest</code> et <code>/var/svc/manifest</code> . Notez que <code>/var/svc/manifest</code> est désapprouvé en tant qu'emplacement standard et ne doit pas être utilisé pour les nouveaux manifestes.

Vous ne pouvez pas saisir propriété au sein de n'importe quel des conflits couche. Une propriété en conflit dans la couche `admin` remplace la propriété précédente. Si la même propriété dans, est concédé sous licence par plusieurs fichiers et que tous les autres n'est pas défini au, une version plus récente de la couche, l'instance est marquée comme étant en conflit et ne pourra pas démarrer tant que la définition est désactivé et / ou le en conflit est définie sur une plus grande part de la propriété couche.

Vous pouvez spécifier la couche de et, par conséquent, les données de configuration identifier pour visualiser les données qui correspondent à des personnalisations administratives et celles qui ont été fournies avec le logiciel. Lorsqu'un client ne spécifie pas la couche dans laquelle vous pouvez extraire les données de configuration des données, la couche la plus haute est fourni. La couche supérieure est déterminée par l'ordre de priorité suivant, par ordre descendant : couche admin, couche site-profile, couche system-profile, couche manifest. Si une propriété possède une valeur dans la couche admin, le référentiel fournira cette valeur. Les personnalisations locales de cette manière sont préférés aux, telles qu'elles étaient fournies lors de l'installation du système.

Référentiel Clichés

Le référentiel capture un *cliché* en lecture seule de chaque service à chaque fois que le service démarre avec succès. Ces instantanés vous permettent de facilement revenir à l'un de ces fonctionne. si nécessaire. Les clichés suivants peuvent être mis à toute nouvelle instance donnée, procédez comme suit :

<code>initial</code>	La configuration initiale lorsque le service et de ses instances ont été importées pour la première fois. Le cliché <code>initial</code> n'est pas créé si un profil démarre le service ou l'instance avant l'importation du manifeste.
<code>previous</code>	Saisie lors de la de configuration en cours est effectuée pour un manifeste de service d'import qui a déjà été distribués. Le le service a peut - être déjà été fournies par le manifeste en cours d'import ou par un autre manifeste.
<code>running</code>	La configuration en cours d'exécution de l'instance de service. Lorsque vous modifiez les données de configuration, utilisez la commande <code>svcadm refresh</code> ou <code>svccfg refresh</code> pour envoyer les nouvelles valeurs au cliché en cours d'exécution.
<code>start</code>	Configuration capturée pendant une transition réussie à l'état <code>online</code> .

Sauvegardes du référentiel

SMF effectue automatiquement les *sauvegardes* suivantes du référentiel de configuration de service :

- La sauvegarde boot est effectuée juste avant la première modification du référentiel pendant chaque démarrage du système.
- Les sauvegardes `manifest_import` se produisent avant l'achèvement de `svc:/system/early-manifest-import:default` ou de `svc:/system/manifest-import:default` si le service a importé de nouveaux manifestes ou exécuté des scripts de mise à niveau.

Quatre sauvegardes de chaque type sont mises à jour par le système supprimé par les plus anciens les sauvegardes, selon vos besoins.

Vous pouvez restaurer le référentiel appartenant à l'un des deux sauvegardes. Reportez-vous à la section [“Restauration d'un référentiel à partir d'une sauvegarde” à la page 112.](#)

Fichiers de configuration et services SMF

Il est conseillé d'utiliser le mécanisme SMF à utiliser pour démarrer des applications. Dans la plupart des cas, l'utilisation de SMF des fichiers de configuration remplace qui permet de gérer les services. Cette section décrit la procédure à suivre les plus utilisées les scripts et fichiers de configuration héritées sont gérées.

scripts /etc/rc?.d

Les répertoires /etc/rc?.d, où ? représente un niveau d'exécution, contiennent les scripts d'initialisation et d'arrêt hérités pour la gestion des services qui s'exécutent lors des transitions de niveau d'exécution. La plupart des services auparavant mis en oeuvre par les scripts /etc/rc?.d sont gérés par SMF. Certains scripts /etc/rc?.d sont conservés pour vous permettre d'utiliser des applications tierces qui n'admettent que ces services comme scripts /etc/rc*.d. Ces scripts sont physiquement liés aux fichiers du répertoire /etc/init.d. Pour plus d'informations sur les scripts /etc/rc?.d et les niveaux d'exécution, reportez-vous au fichier /etc/init.d/README, aux fichiers README des répertoires /etc/rc?.d et à la page de manuel [inittab\(4\)](#). Pour savoir comment convertir un script de contrôle d'exécution, reportez-vous à [“Procédure de conversion d'un script de contrôle d'exécution en service SMF” à la page 97.](#) Après avoir converti un script rc?.d, renommez-le de *Sscript* en *sscript* pour le supprimer.

scripts /etc/init.d

Le répertoire /etc/init.d contient les scripts d'initialisation et de fin pour la modification des états init. Certains de ces scripts sont physiquement liés aux fichiers des répertoires /etc/rc?.d. Pour plus d'informations sur les scripts /etc/init.d, reportez-vous à /etc/init.d/README et à la page de manuel [init.d\(4\)](#).

Les scripts de contrôle d'exécution hérités init.d sont représentés par des FMRI SMF commençant par lrc au lieu de svc. Par exemple, le script de configuration PPP /etc/rc2.d/S47pppd est représenté par le service lrc:/etc/rc2_d/S47pppd. L'état de ces services lrc est legacy_run. Comme l'illustre l'exemple suivant, vous pouvez obtenir la liste de noms et les services hérités les heures de début, mais il n'est pas possible d'administrer ces services à l'aide de SMF.

```
$ svcs lrc:\*
STATE          STIME          FMRI
```

```
legacy_run      9:34:54  lrc:/etc/rc2_d/S47pppd
legacy_run      9:34:54  lrc:/etc/rc2_d/S89PRESERVE
$ svcs -l lrc:/etc/rc2_d/S47pppd
svcs: Operation not supported for legacy service 'lrc:/etc/rc2_d/S47pppd'
$ svccfg -s lrc:/etc/rc2_d/S47pppd listprop
svccfg: Operation not supported for legacy service 'lrc:/etc/rc2_d/S47pppd'
```

entrées /etc/inittab

Entrées du processus de contrôle de fichiers /etc/inittab, répartition par init. Vous ne devez en aucun cas modifier le fichier /etc/inittab directement. Modifiez plutôt les services SMF. Reportez-vous à la section [“Modification d'une valeur de propriété ttymon” à la page 70](#) pour obtenir un exemple de la procédure à suivre pour modifier un paramètre spécifié pour ttymon.

Pour plus d'informations sur le format des entrées de fichiers /etc/inittab, reportez-vous à la page de manuel [inittab\(4\)](#). Pour plus d'informations sur les niveaux d'exécution reportez-vous à la page de manuel [inittab\(4\)](#) et à /etc/init.d/README.

fichier /etc/inetd.conf

Les services auparavant configurés à l'aide du fichier `inetd.conf` le sont maintenant au moyen de SMF. Les configurations dans le fichier `inetd.conf` doivent être converties en services SMF pour pouvoir être utilisées. Reportez-vous à la rubrique [“Conversion des services inetd en services SMF” à la page 119](#). Pour les services `inetd` qui sont déjà convertis en services SMF, reportez-vous à la section [“Modification des services contrôlés par inetd” à la page 82](#).

fichier /etc/nscd.conf

fichier /etc/nsswitch.conf

fichier /etc/resolv.conf

Il est fortement déconseillé de modifier ces fichiers. Les modifications seront perdues. Ces fichiers sont automatiquement générés à partir de données à des fins de compatibilité descendante avec SMF application pouvant d'analyser le fichier. Utilisez la commande `svccfg setprop` pour modifier les valeurs de propriété comme indiqué dans la section [“Définition des valeurs de propriété” à la page 68](#).

La fonction du fichier `nscd.conf` est remplacée par le service SMF `svc:/system/name-service-cache`. Reportez-vous à la page de manuel [nscd.conf\(4\)](#) pour voir quelles propriétés de `name-service-cache` configurer au lieu de modifier le fichier `nscd.conf`.

La fonction du fichier `nsswitch.conf` est remplacée par le service SMF `svc:/system/name-service/switch`. Reportez-vous à la page de manuel [nsswitch.conf\(4\)](#) pour connaître les propriétés de `name-service/switch` à configurer au lieu de modifier le fichier `nsswitch.conf`.

La fonction du fichier `resolv.conf` est remplacée par le service SMF `svc:/network/dns/client`. Reportez-vous à la page de manuel [resolv.conf\(4\)](#) pour connaître les propriétés de `dns/client` à modifier au lieu de modifier le fichier `resolv.conf`.

Ces fichiers sont consulter des exemples de fichiers de configuration que vous ne devez pas modifier. Autres fichiers existent. Il y a quelques cas où la modification d'un fichier de configuration est la bonne manière de modifier la configuration, comme décrit dans [“Modification des services configurés par un fichier” à la page 84](#). Fichier de configuration, avant de modifier tout lire les commentaires dans le fichier et correspondante (le cas échéant) pour vous assurer que la modification de page de manuel le fichier est le meilleur moyen pour modifier la configuration pour le service concerné.

Privilèges de gestion de service

Requiert la modification de l'état et la configuration des privilèges accrus. Utilisez l'une des méthodes suivantes pour obtenir le privilège dont vous avez besoin. Reportez-vous à [“Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2”](#) pour plus d'informations sur les autorisations, profils et rôles, y compris les procédures permettant de déterminer le profil ou rôle dont vous avez besoin et la manière d'attribuer les privilèges.

Autorisations

Reportez-vous à la page de manuel [smf_security\(5\)](#) pour des informations plus détaillées sur les autorisations nécessaires pour les opérations SMF. Vous pouvez par ailleurs inspecter un service particulier pour des propriétés telles que `action_authorization`, `modify_authorization`, `read_authorization` et `value_authorization`.

Profils de droits

Utilisez la commande `profiles` pour répertorier les profils de droits d'accès qui vous sont affectés. Le profil Gestion des services vous octroie les autorités `solaris.smf.manage` et `solaris.smf.modify` et vous permet d'utiliser les commandes `svcadm` et `svccfg`. Le profil de droits Service Operator (opérateur de service) vous octroie les autorités `solaris.smf.manage` et `solaris.smf.modify.framework`.

Rôles

Utilisez la commande `roles` pour répertorier les rôles qui vous sont affectés. Si vous disposez du rôle `root`, vous pouvez utiliser la commande `su` pour prendre le rôle `root`.

Commande sudo

En fonction de la stratégie de sécurité de votre site, vous pouvez être en mesure d'utiliser la commande `sudo` avec votre mot de passe utilisateur pour exécuter une commande privilégiée.

♦ ♦ ♦ CHAPITRE 2

Obtention d'informations sur les services

Ce chapitre décrit la procédure pour obtenir des informations sur des services, tels que les opérations suivantes :

- Etat du service, les dépendances et d'autres valeurs de propriété
- Processus lancés par un service de contrat
- Emplacement du fichier journal pour consulter des informations sur le dépannage
- Transition d'événements et de services FMA des paramètres de notification d'événement

Création de listes de services sur le système

La commande `svcs` correspond à la commande primaire pour répertorier l'état et le statut des instances de service.

Affichage de l'état de service

Reportez-vous à la section [“Etats des services” à la page 21](#) pour obtenir une description des états présentés dans ces exemples.

EXEMPLE 2-1 Etablissement de la liste de tous les services Activé

Sans options ou arguments, la fonction `svcs` répertorie toutes les instances de service activées dans ce système, ainsi que les instances temporairement désactivées.

Les instances de service à l'état `disabled` dans cette liste seront activées à la prochaine initialisation du système. Les instances à l'état `legacy_run` ne sont pas gérées par SMF. Reportez-vous à [“Fichiers de configuration et services SMF” à la page 28](#) pour plus d'informations à propos de ces services hérités. Reportez-vous à [“Plus d'informations sur les états de service” à la page 39](#) si des services se trouvent dans l'état `maintenance`, `degraded` ou `offline`.

Indique le temps nécessaire à la colonne STIME saisi la liste l'état de l'instance. Si l'instance porte cet état d'il y a plus de 24 heures STIME colonne inclut la date à laquelle la.

```
$ svcs
STATE      STIME      FMRI
legacy_run Sep_09      lrc:/etc/rc2_d/S47pppd
legacy_run Sep_09      lrc:/etc/rc2_d/S81dodatadm_udaplt
legacy_run Sep_09      lrc:/etc/rc2_d/S89PRESERVE
disabled   Sep_09      svc:/system/vbiosd:default
online     Sep_09      svc:/system/early-manifest-import:default
online     Sep_09      svc:/system/svc/restarter:default
...
```

EXEMPLE 2-2 Etablissement de la liste de tous les services installés

Pour obtenir la liste de toutes les instances installées sur le système, y compris celles à l'état disabled qui ne seront pas activées automatiquement à la prochaine initialisation, utilisez la commande `svcs -a`.

```
$ svcs -a
```

Un astérisque (*) est ajouté à l'état pour les instances de service qui sont en transition de l'état répertorié à un autre état. Par exemple, `offline*` signifie probablement que l'instance est encore en train d'exécuter sa méthode de démarrage.

Un point d'interrogation (?) s'affiche si l'état est absent ou non reconnu.

EXEMPLE 2-3 Service d'une liste de toutes les instances d'une

Si un nom de service est spécifié, la commande `svcs` répertorie toutes les instances du service. Reportez-vous à la section [“Affichage des informations sur le service sélectionné” à la page 34](#) pour des informations sur l'option `-o`.

```
$ svcs -Ho inst identity
node
domain
```

Présentation d'informations supplémentaires à propos des services

La commande `svcs -l` affiche une longue liste pour chaque instance de service spécifiée, avec des informations détaillées sur l'état de l'instance, les chemins vers ses fichiers de journal et de configuration, ses types de dépendance, les valeurs des attributs de redémarrage des dépendances et l'état de dépendance. L'exemple suivant indique que toutes les dépendances de ce de l'instance de service requis sont en ligne. La dépendance désactivée est facultative. Pour plus d'informations sur les types de dépendance et les valeurs des attributs de redémarrage,

reportez-vous à [“Affichage des dépendances de service” à la page 34](#). Dans la sortie `svcs -l`, des états autres que ceux décrits dans [“Etats des services” à la page 21](#) sont possibles pour les dépendances. Reportez-vous à la page de manuel [svcs\(1\)](#) pour voir des descriptions. L'exemple suivant indique également que l'instance de service spécifiée en ligne est temporairement et n'a été activée, s'il s'agit d'un service, type de contrat de service. Reportez-vous à la section [“Modèles de service” à la page 20](#) pour des définitions des types de service. Si la valeur de service est suivie d'un astérisque, par exemple `offline*`, l'instance est en transition, et le champ d'état suivant affiche une valeur d'état au lieu de `none`. La valeur `state_time` est l'heure où l'instance est passée à l'état répertorié.

```
$ svcs -l net-snmp
fmri          svc:/application/management/net-snmp:default
name          net-snmp SNMP daemon
enabled       true (temporary)
state         online
next_state    none
state_time    September 17, 2013 05:57:26 PM PDT
logfile       /var/svc/log/application-management-net-snmp:default.log
restarter     svc:/system/svc/restarter:default
contract_id   160
manifest      /etc/svc/profile/generic.xml
manifest      /lib/svc/manifest/application/management/net-snmp.xml
dependency    require_all/none svc:/system/filesystem/local (online)
dependency    optional_all/none svc:/milestone/name-services (online)
dependency    optional_all/none svc:/system/system-log (online)
dependency    optional_all/none svc:/network/rpc/rstat (disabled)
dependency    require_all/restart svc:/system/cryptosvc (online)
dependency    require_all/restart svc:/milestone/network (online)
dependency    require_all/refresh file://localhost/etc/net-snmp/snmp/snmpd.conf (online)
dependency    require_all/none svc:/milestone/multi-user (online)
```

EXEMPLE 2-4 Processus démarrés par présentant un contrat de service

Utilisez la commande `svcs -p` pour afficher les ID de processus et les noms de commande des processus démarrés par une instance de service de contrat. Le service `net-snmp` gère l'agent SNMP `/usr/sbin/snmpd` qui récolte les informations sur un système par le biais d'un ensemble de MIB (Management Information Base, base d'informations de gestion).

```
$ svcs -p net-snmp
STATE      STIME      FMRI
online     17:57:26   svc:/application/management/net-snmp:default
           17:57:26     5022 snmpd
```

EXEMPLE 2-5 Redémarrage d'automatiquement présentant après un contrat de service Arrêt du processus

Le nombre d'instances du service de contrat sont automatiquement redémarrés lorsque le contrat vide. Les processus SMF également tente de redémarrer l'instance associée à un contrat de service dans le cadre de la récupération automatique à partir des événements de panne

matérielle ou logicielle. L'exemple suivant montre qu'après l'arrêt du processus `/usr/sbin/snmpd`, il est automatiquement redémarré avec un nouvel ID de processus. L'instance `net-snmp:default` est toujours en ligne et possède une nouvelle heure de démarrage.

```
$ kill 5022
$ svcs -p net-snmp
STATE      STIME      FMRI
online     17:57:59   svc:/application/management/net-snmp:default
           17:57:59   5037 snmpd
```

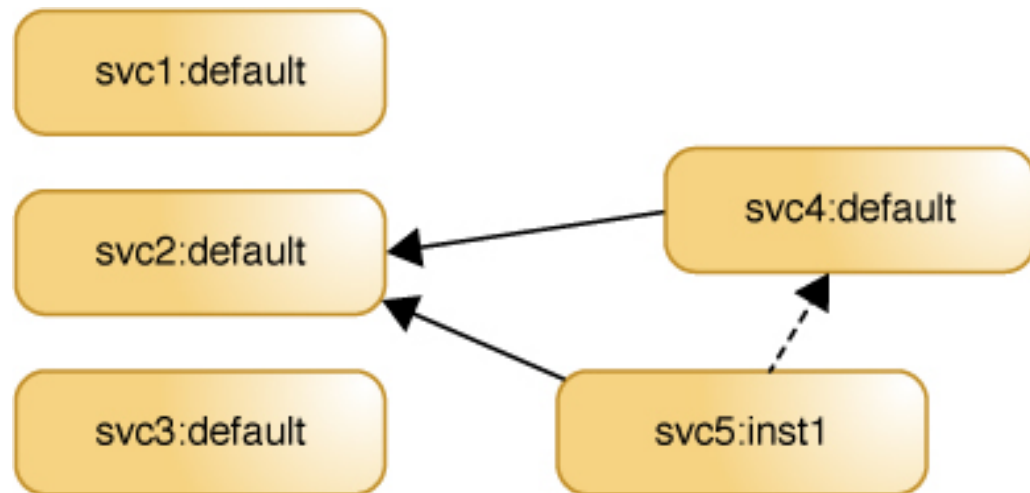
Affichage des informations sur le service sélectionné

La sortie de la commande `svcs` peut être très utile pour le traitement pipeline vers d'autres commandes ou l'utilisation dans des scripts. L'option `-o` de la commande `svcs` vous permet de spécifier les colonnes d'information que vous souhaitez voir et leur ordre. Vous pouvez diriger le nom de service et le nom de l'instance dans des colonnes différentes, l'état actuel et de l'état suivant de l'ID et que le contrat de service, par exemple. Les options `-s` et `-S` permettent de spécifier l'ordre de tri de la sortie pour une ou plusieurs colonnes.

Affichage des dépendances de service

Instance de service des relations de dépendance régissent transitions d'état. Reportez-vous à [“Dépendances de service” à la page 21](#) pour une description de haut niveau des dépendances. Reportez-vous au [Chapitre 5, Utilisation de SMF pour contrôler votre application](#) pour des descriptions détaillées et pour savoir comment spécifier différents types de dépendances.

Dans la figure suivante, les instances de service `svc1:default`, `svc2:default` et `svc3:default` ne nécessitent aucun autre service, fichier ou ressource pour démarrer. Ces instances peuvent démarrer en parallèle, exécutent leurs méthodes de démarrage, et passer à l'état online (en ligne) (pull) sans attente sur les autres ressources. L'instance `svc4:default` ne peut pas exécuter sa méthode de démarrage tant que `svc2:default` n'est pas en ligne. L'instance `svc5:inst1` a besoin des ressources `svc2:default` et `svc4:default`. La dépendance de `svc5:inst1` sur `svc4:default` est facultative ; elle est satisfaite si `svc4:default` se trouve dans l'un des états suivants : activé et en ligne, désactivé, ou absent. L'instance `svc5:inst1` doit attendre que `svc2:default` soit en ligne et si `svc4:default` est présent et activé, `svc5:inst1` doit aussi attendre que `svc4:default` soit en ligne. Si `svc4:default` est présent et désactivé, ou non présent, `svc5:inst1` n'a pas besoin d'attendre `svc4:default`.

FIGURE 2-1 Les relations de dépendance de service

Regroupements de dépendance

Chaque dépendance est affectée à un des regroupements suivants. Dans le regroupement ce regroupement définit comment les dépendances sont satisfaites.

<code>require_all</code>	Cette dépendance est remplie lorsque les deux conditions suivantes sont remplies : ■ Toutes les dépendances de service de ce groupe sont en cours d'exécution, soit <code>online</code>, soit <code>degraded</code>. ■ Toutes les dépendances de fichiers dans ce regroupement sont présents.
<code>require_any</code>	Cette dépendance est remplie lorsque l'une des conditions suivantes est remplie : ■ Au moins l'une des dépendances de service de ce groupe est en cours d'exécution, soit <code>online</code>, soit <code>degraded</code>. ■ Au moins l'un des dépendances de fichier de ce groupe n'est présent.
<code>optional_all</code>	Cette dépendance est remplie lorsque toutes les dépendances de service de ce groupe satisfont l'une des conditions suivantes :

- Le service est en cours d'exécution, soit online, soit degraded.
- Le service exige le stockage d'une action d'administration à exécuter. Le service n'est pas présent, n'est pas complet, est disabled, en maintenance ou offline et en attente des dépendances qui ont besoin d'une action administrative pour démarrer.

Dépendances de fichiers dans ce groupe peut être présente ou absente.

Cette dépendance n'est pas remplie si l'instance de service est en cours et ne nécessite pas l'intervention de l'administrateur de données puisse démarrer. Dans ce cas, le service dépendant attendes pour cet dépendance consistant à démarrer ou d'attentes pour la détermination que la dépendance ne peut pas démarrer sans d'une action d'administration.

exclude_all

Cette dépendance est remplie lorsque les deux conditions suivantes sont remplies :

- Toutes les dépendances de service de ce groupe sont disabled, en maintenance ou absentes.
- Toutes les dépendances de fichiers dans ce regroupement ne sont pas présents.

Présentation des instances dont un service dépend

La commande `svcs -d` répertorie les instances de service dont dépend un service donné.

Cet exemple montre les instances de service dont dépend le service `system-repository` :

```
$ svcs -d system-repository
STATE      STIME    FMRI
online     Sep_09   svc:/milestone/network:default
online     Sep_09   svc:/system/filesystem/local:default
online     Sep_09   svc:/system/filesystem/autofs:default
```

La commande `svcs -l` répertorie aussi les services dont dépend un service donné. Outre le nom et l'état de la dépendance, la sortie de l'option `-l` affiche le type, ou groupe, de la dépendance et la valeur de sa propriété `restart_on`. Dans cet exemple, deux des dépendances ne sont pas obligatoires et un est facultative. Reportez-vous à la section [“Regroupements de dépendance” à la page 35](#) pour des descriptions de la manière dont les dépendances dans ces groupes affectent le service dépendant. Reportez-vous à la section [“Regroupements de dépendance” à la page 35](#) pour des descriptions de la manière dont les différentes valeurs de la propriété `restart_on` de la dépendance affectent le service dépendant.

```
$ svcs -l system-repository
```

```

fmri      svc:/application/pkg/system-repository:default
name      IPS System Repository
enabled   false
state     disabled
next_state none
state_time Mon Sep 09 18:42:28 2013
restarter svc:/system/svc/restarter:default
manifest  /lib/svc/manifest/application/pkg/pkg-system-repository.xml
dependency require_all/error svc:/milestone/network:default (online)
dependency require_all/none svc:/system/filesystem/local:default (online)
dependency optional_all/error svc:/system/filesystem/autofs:default (online)

```

Vous pouvez également utiliser la commande `svccprop` pour répertorier ces dépendances. Ce formulaire affiche le groupe et les valeurs `restart_on` de la dépendance sur des lignes séparées ; il n'affiche pas l'état de la dépendance.

```

$ svccprop -g dependency system-repository:default
network/entities fmri svc:/milestone/network:default
network/grouping astring require_all
network/restart_on astring error
network/type astring service
filesystem-local/entities fmri svc:/system/filesystem/local:default
filesystem-local/grouping astring require_all
filesystem-local/restart_on astring none
filesystem-local/type astring service
autofs/entities fmri svc:/system/filesystem/autofs:default
autofs/grouping astring optional_all
autofs/restart_on astring error
autofs/type astring service

```

Présentation des instances dépendant d'un service

La commande `svcs -D` répertorie les instances de service qui dépendent d'un service donné.

Cet exemple montre les instances de service qui dépendent du service `system-repository` :

```

$ svcs -D system-repository
STATE      STIME      FMRI
online     16:39:30  svc:/application/pkg/zones-proxyd:default

```

La commande suivante confirme que `zones-proxyd` dépend de `system-repository`.

```

$ svcs -do svc,desc zones-proxyd
SVC          DESC
application/pkg/system-repository IPS System Repository
system/filesystem/minimal      minimal file system mounts
milestone/network              Network milestone

```

La commande suivante affiche plus d'informations sur la manière dont `zones-proxyd` dépend de `system-repository`. La dernière ligne de cette sortie indique que le service `zones-proxyd` requiert que le service `system-repository` soit en cours d'exécution et montre que `system-repository` est en cours d'exécution. Cette sortie indique également que le service `zones-proxyd` sera redémarré si le service `system-repository` est actualisé.

```
$ svcs -l zones-proxyd
fmri      svc:/application/pkg/zones-proxyd:default
name     Zones Proxy Daemon
enabled   true
state     online
next_state none
state_time January 6, 2014 04:39:30 PM PST
restarter svc:/system/svc/restarter:default
manifest  /lib/svc/manifest/application/pkg/zoneproxyd.xml
dependency require_any/none svc:/system/filesystem/minimal (online)
dependency require_any/error svc:/milestone/network (online)
dependency require_all/restart svc:/application/pkg/system-repository (online)
```

Redémarrage automatique d'un service

Un service en cours d'exécution peut être configuré pour redémarrer si l'une de ses dépendances est arrêté ou que vous l'avez actualisée. Si les dépendances d'un service en cours d'exécution (état `online` ou `degraded`) ne sont pas satisfaites, le service passe à l'état `offline`. Si une dépendance de redémarrage de services régénération après une peut arrêter ou être satisfaite, les dépendances et la personne à charge à nouveau service passe en remontant jusqu'à une zone est en cours d'exécution.

Les facteurs suivants déterminent si un service redémarre après l'arrêt ou l'actualisation d'une dépendance `require_all`, `require_any` ou `optional_all` :

- La dépendance a-t-elle été arrêtée ou actualisée ? Si elle a été arrêtée, a-t-elle été arrêtée en raison d'une erreur telle qu'une erreur de matériel ou un core dump, ou pour toute autre raison telle qu'une action d'administration ?
- La valeur de l'attribut `restart_on` de la dépendance. Les valeurs possibles sont `none`, `error`, `restart` et `refresh`.

Comme indiqué dans le tableau suivant, si la valeur de l'attribut `restart_on` de la dépendance est `none`, le service dépendant n'est pas redémarré lorsque la dépendance est arrêtée ou actualisée. Si la valeur de l'attribut `restart_on` de la dépendance est `refresh`, le service dépendant est redémarré à chaque fois que la dépendance est arrêtée ou actualisée. Si la valeur de `restart_on` est `error`, le service dépendant redémarre uniquement si la dépendance s'est arrêtée pour cause d'erreur. Si la valeur de `restart_on` est `restart`, le service dépendant redémarre uniquement si la dépendance a été actualisée.

TABLEAU 2-1 Après une automatiquement de dépendance Redémarrage d'une Arrêter service

Dépendance <code>require_all</code> , <code>require_any</code> OU <code>optional_all</code>	Valeur de l'attribut de dépendance <code>restart_on</code>			
L'événement d'arrêt ou de régénération	<code>none</code>	<code>error</code>	<code>restart</code>	<code>refresh</code>
Arrêt en raison d'une erreur	Pas de redémarrage	Redémarrage	Pas de redémarrage	Redémarrage
Autre arrêt	Pas de redémarrage	Pas de redémarrage	Pas de redémarrage	Redémarrage
Actualisation	Pas de redémarrage	Pas de redémarrage	Redémarrage	Redémarrage

“Présentation des instances dont un service dépend” à la page 36 montre que le service `system-repository` a deux dépendances `require_all` et une `optional_all`. La commande suivante indique que le service `system-repository` redémarre si le service `milestone/network` ou `system/filesystem/autofs` s'arrête pour cause d'erreur, mais pas si ces services s'arrêtent pour une autre raison ou sont actualisés. Le service `system-repository` ne redémarre pas si le service `system/filesystem/local` est actualisé ou arrêté pour quelque raison que ce soit.

```
$ svccfg -s system-repository:default listprop -o propname,propval '*restart_on'
network/restart_on          astring      error
filesystem-local/restart_on astring      none
autofs/restart_on           astring      error
```

Plus d'informations sur les états de service

En l'absence d'arguments, la commande `svcs -x` fournit des informations explicatives sur les instances de service suivantes :

- Instances qui sont activés, mais pas en cours d'exécution.
- D'autres instances services activés qui empêchent une ne sera pas exécutée.

Si tous les services activés sont en cours d'exécution, la commande `svcs -x` ne génère pas de sortie.

Dans l'exemple suivant, le service `pkg/depot` se trouve dans l'état maintenance parce que sa méthode de démarrage a échoué.

```
$ svcs -x
svc:/application/pkg/depot:default (IPS Depot)
  State: maintenance since September 11, 2013 01:30:42 PM PDT
Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
  See: http://support.oracle.com/msg/SMF-8000-KS
  See: pkg.depot-config(1M)
  See: /var/svc/log/application-pkg-depot:default.log
Impact: This service is not running.
```

La sortie suggère un article de la base de connaissances d'autorétablissement prédictif, à partir de My Oracle Support, une page de manuel de référence. un fichier journal pour déterminer la raison pour laquelle le à la méthode de démarrage a échoué. Reportez-vous à [“Affichage des fichiers journaux d'un service” à la page 41](#) pour des informations sur les différentes manières d'afficher les fichiers journaux. Reportez-vous à [“Réparation d'une instance endommagée, hors ligne ou en cours de maintenance” à la page 107](#) pour des informations sur la manière de réparer un service à l'état maintenance.

Dans l'exemple suivant, le service print/server a des services dépendants qui ne sont pas en cours d'exécution. Les services dépendants ne peuvent pas s'exécuter car le service print/server a été désactivé.

```
$ svcs -x
svc:/application/print/server:default (LP print server)
State: disabled since Fri Mar 08 14:42:32 2013
Reason: Disabled by an administrator.
See: http://sun.com/msg/SMF-8000-05
See: lpsched(1M)
Impact: 2 dependent services are not running. (Use -v for list.)
$ svcs -xv
svc:/application/print/server:default (LP print server)
State: disabled since Fri Mar 08 14:42:32 2013
Reason: Disabled by an administrator.
See: http://sun.com/msg/SMF-8000-05
See: man -M /usr/share/man -s 1M lpsched
Impact: 2 dependent services are not running:
    svc:/application/print/rfc1179:default
    svc:/application/print/ipp-listener:default
$ svcs -D print/server
STATE      STIME      FMRI
online     Mar_08     svc:/milestone/multi-user:default
offline    Mar_08     svc:/application/print/ipp-listener:default
offline    Mar_08     svc:/application/print/rfc1179:default
```

Si un argument donné à la commande `svcs -x` ne répond pas aux critères établis au début de cette section, la sortie n'indique aucune raison pour l'état de l'instance, mais affiche des ressources permettant d'obtenir plus d'informations.

```
$ svcs -x smb
svc:/network/smb:default (SMB properties)
State: online since Thu Sep 12 19:16:56 2013
See: smb(4)
See: /var/svc/log/network-smb:default.log
Impact: None.
```

Affichage des fichiers journaux d'un service

SMF conserve des informations sur les actions importantes des agents de redémarrage, les sorties standard des méthodes et les sorties standard des erreurs dans `/var/svc/log/service:instance.log` pour chaque instance de service. Les traits d'union du nom de *service* dans le nom du fichier journal sont remplacés par des barres obliques. La commande `svcs` avec l'option `-L`, `-l` ou `-x` affiche le chemin complet du fichier journal pour l'instance de service spécifiée. La commande `svcs -xL` présente les dernières lignes du fichier journal et vous indique que vous devez utiliser la commande `svcs -Lv` pour afficher tout le fichier journal. La commande `svcs -Lv` affiche le fichier complet, qui peut être très long. Si vous préférez afficher le fichier journal dans un éditeur ou afficher seulement les *n* dernières entrées, par exemple, travaillez sur le résultat de la commande `svcs -L`.

L'exemple suivant montre comment utiliser le fichier journal pour savoir pourquoi le service affiché dans la liste `svcs` est temporairement désactivé.

```
$ svcs
STATE          STIME      FMRI
legacy_run     Sep_09     lrc:/etc/rc2_d/S47pppd
legacy_run     Sep_09     lrc:/etc/rc2_d/S81dodatadm_udaplt
legacy_run     Sep_09     lrc:/etc/rc2_d/S89PRESERVE
disabled       Sep_09     svc:/system/vbiosd:default
online         Sep_09     svc:/system/early-manifest-import:default
online         Sep_09     svc:/system/svc/restarter:default
...
$ svcs -x vbiosd
svc:/system/vbiosd:default (BIOS call emulation)
State: disabled since Mon Sep  9 18:42:37 2013
Reason: Temporarily disabled by service method: "vbiosd is not supported on UEFI systems."
See: http://support.oracle.com/msg/SMF-8000-1S
See: vbiosd(1M)
See: /var/svc/log/system-vbiosd:default.log
Impact: This service is not running.
$ svcs -xL vbiosd
svc:/system/vbiosd:default (BIOS call emulation)
State: disabled since Mon Sep  9 18:42:37 2013
Reason: Temporarily disabled by service method: "vbiosd is not supported on UEFI systems."
See: http://support.oracle.com/msg/SMF-8000-1S
See: vbiosd(1M)
See: /var/svc/log/system-vbiosd:default.log
Impact: This service is not running.
Log:
[ Sep  9 18:42:27 Enabled. ]
[ Sep  9 18:42:37 Executing start method ("/lib/svc/method/svc-vbiosd start"). ]
[ Sep  9 18:42:37 Method "start" exited with status 101. ]
[ Sep  9 18:42:37 "start" method requested temporary disable: "vbiosd is not supported on UEFI systems"
]

Use: 'svcs -Lv svc:/system/vbiosd:default' to view the complete log.
```

```
$ svcs -L vbiosd
/var/svc/log/system-vbiosd:default.log
$ view `svcs -L vbiosd`
```

D'autres fichiers journaux que vous aurez besoin d'effectuer inclure au journal pour connaître le démon d'agent de redémarrage maître et le journal système. Pour afficher le nom de fichier journal et visualiser le fichier journal du démon de redémarrage `svc.startd`, utilisez le nom de service `restart` avec la commande `svcs`. Pour afficher le fichier journal du démon de journalisation système `syslogd`, utilisez le nom de service `system-log`.

Reportez-vous à [“Spécification de la quantité de messages au démarrage”](#) à la page 114 pour des instructions sur la manière de changer la quantité de messages affichés à l'initialisation du système. Reportez-vous à [“Configuration la notification des événements de transition d'état et FMA”](#) à la page 62 pour des instructions sur la manière de configurer les services de sorte qu'ils vous notifient lorsqu'ils passent d'un état de service à un autre ou lorsqu'un événement FMA se produit.

Inspection de la configuration du service

Configuration de service est exprimé dans les propriétés qui sont définies sur les services et instances de service et stockées dans les couches dans le référentiel de configuration de service. Les propriétés qui sont définies sur un service sont héritées par toutes les instances de ce service. Les propriétés qui sont définies sur une instance sont uniquement utilisées par cette instance. Le nombre d'instances du service hérités ont la possibilité de personnaliser les valeurs des propriétés et pouvez définir d'autres propriétés qui ne sont pas définies pour le service parent.

Cette section montre comment récupérer les valeurs de propriété et comment identifier l'utilisation d'une valeur globale pour le service, ne sont propres à une instance, a été livrée avec le logiciel, ou qu'il s'agit d'une personnalisation administrative.

Affichage des descriptions de propriétés et de groupes de propriétés

La commande `svccfg describe` affiche une description des groupes de propriétés et des propriétés d'un service, y compris les valeurs actuelles des propriétés. Sans opérandes, `describe` affiche des descriptions de toutes les propriétés et groupes de propriétés du service ou de l'instance de service sélectionné. Utilisez l'option `-v` pour afficher plus d'informations, y compris une description de la valeur actuelle et une liste des valeurs possibles. Utilisez l'option `-t` pour afficher les informations sur le modèle.

```
$ svccfg -s pkg/server describe network/restart_on
```

```

network/restart_on astring    none
    Determines whether to restart the service due to a dependency refresh, restart, or
    failure.
$ svccfg -s pkg/server describe -v network/restart_on
network/restart_on astring    none
    type: astring
    required: true
    Determines whether to restart the service due to a dependency refresh, restart, or
    failure.
    visibility: readwrite
    minimum number of values: 1
    maximum number of values: 1
    value: none
    value description: Never restart due to dependency refresh, restart, or failure.
    value constraints:
    value name: none
    value name: error
    value name: restart
    value name: refresh

```

Affichage des valeurs de propriété d'instance et de services

Les exemples de cette section décrivent la manière d'afficher des propriétés de service et d'instance et les groupes de propriétés dans différentes vues, des couches, mais pas les instantanés.

EXEMPLE 2-6 Actuellement l'établissement de la liste de l'instance en cours d'utilisation et les propriétés héritées

Par défaut, la commande `svccprop` affiche les valeurs affectées aux propriétés dans le cliché en cours d'exécution, qui sont les valeurs actuellement utilisées. Par défaut, la commande `svccprop` affiche les propriétés dans l'*affichage composé* du cliché en cours d'exécution, ce qui signifie que les propriétés spécifiques de l'instance et les propriétés héritées sont affichées. Si la valeur d'une propriété héritée a été personnalisé dans l'instance, la valeur définie pour l'instance apparaît. La sortie affiche une ligne pour chaque groupe de propriétés, avec la propriété et le nom de propriété séparés par une barre oblique, le type de données de la valeur de la propriété et la valeur de la propriété. Si aucune propriété ou nom de groupe est indiqué, en cours d'exécution toutes les instantané des valeurs de propriété sont affichés.

FMRI ou un motif si la indiquiez une instance de l'opérande n'aboutit pas, les propriétés définies que pour le service sont affichés. Les propriétés définies que pour une instance ne sont pas affichés. La commande suivante indique les propriétés, par exemple, les dépendances de service, le type du service, profil et et les chemins d'accès aux fichiers manifestes.

```
$ svccprop svc:/system/identity
```

Lorsque vous spécifiez une instance, les propriétés, reportez-vous à la vue composée de propriétés personnalisées pour hérités de à l'instance et le service parent. La commande suivante répertorie toutes les propriétés en cours d'exécution pour l'instance spécifiée instantané, y compris les propriétés héritée du parent de service et les propriétés propres à cette instance. Dans le cas de propriétés héritées dont la valeur sont personnalisés pour cette instance, la valeur personnalisé s'affiche. Cet exemple montre les propriétés telles que les dépendances supplémentaires, le chemin du fichier exécutable qui permet de démarrer cette instance, le chemin d'accès au fichier journal pour cette instance, ainsi que les informations sur l'état de l'instance.

```
$ svcprop svc:/system/identity:domain
```

EXEMPLE 2-7 Groupes de propriétés liste spécifié par l'actuellement Propriétés ou en cours d'utilisation

Utilisez l'option `-p` pour afficher des propriétés spécifiques ou toutes les propriétés d'un groupe de propriétés spécifique.

```
$ svcprop -p pkg/port pkg/server
svc:/application/pkg/server:oss:properties/pkg/port count 82
svc:/application/pkg/server:s11:properties/pkg/port count 81
svc:/application/pkg/server:default:properties/pkg/port count 80
$ svcprop -p pkg pkg/server:s11
pkg/inst_root astring /export/ipsrepos/Solaris11
pkg/port count 81
...
pkg/ssl_cert_file astring ""
pkg/ssl_key_file astring ""
...
```

EXEMPLE 2-8 Les valeurs d'instance d'répertoriant dans le Service et Modification de la vue

Avec des options, la commande `svcprop` peut montrer l'affichage d'*édition* au lieu du cliché en cours d'exécution. La modification vue affiche les dernières modifications apportées. Les modifications ou d'édition figurant dans une vue peut n'était peut - être pas été enregistrée dans l'instantané en régénérant en cours d'exécution ou le redémarrage de l'instance. Les commandes suivantes illustrent la différence entre la vue en cours d'exécution et du mode de modification au instantané. Les instances `oss` et `s11` viennent d'être créées et des valeurs de propriétés ont été modifiées, mais les instances ne sont pas encore actualisées. La première commande montre l'instantané vue composée de l'en cours d'exécution. Etant donné que les instances n'ont pas été actualisées depuis leur personnalisation, les valeurs affichées sont celles du service `pkg/server`. L'option `-c` montre l'affichage composé des valeurs d'édition. L'option `-C` affiche les valeurs d'édition sans composition. Ceci n'étant pas une vue composée, aucune valeur n'est trouvé pour l'instance qui n'a pas été personnalisé.

```
$ svcprop -p pkg/port pkg/server
svc:/application/pkg/server:oss:/properties/pkg/port count 80
svc:/application/pkg/server:s11:/properties/pkg/port count 80
```

```

svc:/application/pkg/server:default/:properties/pkg/port count 80
$ svccprop -c -p pkg/port pkg/server
svc:/application/pkg/server:oss/:properties/pkg/port count 82
svc:/application/pkg/server:s11/:properties/pkg/port count 81
svc:/application/pkg/server/:properties/pkg/port count 80
$ svccprop -C -p pkg/port pkg/server
svc:/application/pkg/server:oss/:properties/pkg/port count 82
svc:/application/pkg/server:s11/:properties/pkg/port count 81
svccprop: Couldn't find property 'pkg/port' for instance 'svc:/application/pkg/server:default'.

```

La commande `svccfg` affiche les valeurs de propriétés d'édition par défaut et non les valeurs du clicé en cours d'exécution. Vous pouvez forcer `svccfg` à afficher les valeurs du clicé en cours d'exécution à l'aide de la sous-commande `selectsnap`, comme indiqué dans [“Affichage des valeurs d'un instantané particulier” à la page 48](#).

La commande `svccfg` n'affiche les valeurs du service parent que si vous spécifiez un service parent ; de même, elle n'affiche les valeurs d'une instance que si vous spécifiez une instance. Si vous ne recevez pas de sortie de la commande `svccfg listprop`, la propriété spécifiée n'est peut-être pas définie sur le service parent ou l'instance que vous avez spécifié. Si la propriété a été supprimée, utilisez `listcust -M` pour afficher la valeur masquée, comme indiqué dans [“Affichage des personnalisations de configuration” à la page 49](#).

La commande suivante répertorie tous les modification des valeurs de propriété pour le service indiqué car il n'y a pas de groupe de propriétés ou le nom de propriété est spécifiée. En plus de la sortie illustrée par la commande `svccprop svc:/system/identity`, cette sortie inclut les noms et types de groupes de propriétés et les données sur les modèles.

```
$ svccfg -s svc:/system/identity listprop
```

La commande suivante répertorie tous les modification des valeurs de propriété pour l'instance de service spécifiée. Etant donné que cette commande n'affiche pas la vue composée, cette sortie n'affiche pas les chemins d'accès au profil et les fichiers manifestes, par exemple.

```
$ svccfg -s svc:/system/identity:domain listprop
```

EXEMPLE 2-9 Groupes de propriétés liste spécifiée dans le Propriétés ou Modification de la vue

La commande suivante répertorie tous les modification des valeurs de propriété de groupe dans la propriété indiquée pour l'instance de service spécifiée. L'option `-o` permet de sélectionner les colonnes à afficher. Reportez-vous à la page de manuel [svccfg\(1M\)](#) pour la liste des noms de colonne valides.

```

$ svccfg -s pkg/server:s11 listprop pkg
pkg          application
pkg/inst_root astring      /export/ipsrepos/Solaris11
pkg/port      count        81
$ svccfg -s pkg/server:s11 listprop -o propname,value pkg
inst_root    /export/ipsrepos/Solaris11
port         81

```

Affichage des propriétés dans un type de groupe de propriétés

En plus d'afficher les valeurs d'une propriété à le nom de propriété ou nom du groupe de propriétés, vous pouvez également afficher les valeurs de propriété par la propriété type de groupe.

EXEMPLE 2-10 Groupes de propriétés et leurs types présentant

La sous-commande `listpg` de la commande `svccfg` indique le nom et le type de chaque groupe de propriétés.

```
$ svccfg -s pkg/server listpg
pkg          application
pkg_bui      application
pkg_secure   application
fs           dependency
autofs       dependency
ntp          dependency
network      dependency
general      framework
manifestfiles framework
start        method
stop         method
tm_common_name template
$ svccfg -s pkg/server:s11 listpg
pkg          application
general      framework
restarter    framework          NONPERSISTENT
```

Les groupes de propriétés non permanents contiennent généralement un état de programme actif. Les valeurs des propriétés comprises dans des groupes de propriétés non permanents sont effacées au cours de l'initialisation du système.

Indiquez un nom du groupe de propriétés uniquement pour afficher cette propriété le type de groupe.

```
$ svccfg -s pkg/mirror listpg config
config application
```

EXEMPLE 2-11 Un groupe de propriétés répertoriant Type Properties of a

Utilisez l'option `-g` de la commande `svccprop` pour afficher les propriétés d'un groupe de propriétés spécifique. Parmi les types de groupes de propriétés, on trouve `application`, `dependency`, `method`, `framework` et `template`.

```
$ svcprop -g method pkg/server:s11
start/exec astring %{pkg/pkg_root}/lib/svc/method/svc-pkg-server\ %m
start/timeout_seconds count 0
start/type astring method
stop/exec astring %{pkg/pkg_root}/lib/svc/method/svc-pkg-server\ %m %{restarter/contract}
stop/timeout_seconds count 30
stop/type astring method
$ svcprop -g method -p exec pkg/server:s11
start/exec astring %{pkg/pkg_root}/lib/svc/method/svc-pkg-server\ %m
stop/exec astring %{pkg/pkg_root}/lib/svc/method/svc-pkg-server\ %m %{restarter/contract}
```

Affichage de la couche de définition de la valeur

Référentiel stocke les données de propriété de configuration de service dans les couches en fonction de la source des données. Les commandes `svcprop` et `svccfg` peuvent afficher la couche source d'une valeur de propriété. L'option `-l` des commandes `svcprop` et `svccfg` nécessite un argument pour spécifier la couche sur laquelle vous souhaitez obtenir des informations. Les valeurs des arguments sont `manifest`, `system-profile`, `site-profile` et `admin`. Indique si un spécifique la sortie a été définie à la valeur de la propriété manifeste de service, un profil, ou par un administrateur. Reportez-vous à [“Référentiel Couches” à la page 26](#) pour des descriptions des couches. Le mot-clé `all` est un alias pour toutes les couches. Si la couche vous spécifiez n'est pas la source des valeurs de propriété que vous avez demandé, aucune sortie n'est affichée.

La commande suivante indique que certaines valeurs de propriété, proviennent de certains du manifeste de service ont été définies par un administrateur et certaines propriétés ont des valeurs dans plusieurs couches. La propriété `pkg/readonly` a une valeur définie dans le manifeste de service et un administrateur a défini cette même valeur. Des valeurs provenant de différentes couches peuvent être différents.

```
$ svcprop -l all -p pkg pkg/server:s11
pkg/port count admin 81
pkg/inst_root astring admin /export/ipsrepos/Solaris11
pkg/address net_address manifest
pkg/cfg_file astring manifest ""
...
pkg/readonly boolean manifest true
pkg/readonly boolean admin true
...
```

L'option `-l` de la commande `svccfg listprop` peut aussi prendre l'argument `current`. L'utilisation de `current` en tant qu'argument `-l` affiche les mêmes valeurs de propriété que celles affichées sans utiliser l'option `-l`. La seule différence dans la sortie est que le nom de la couche est également affichée. Les données non permanentes n'indiquent pas de nom de couche (la troisième colonne affiche `<none>`) parce que le référentiel de configuration de service ne conserve pas les informations sur les couches pour les données non permanentes. En général, les groupes de propriétés non permanents définissent un état de programme actif, tandis que

les propriétés des groupes de propriétés non permanents sont effacés lors de l'initialisation du système.

```
$ svccfg -s pkg/server:s11 listprop -l current
pkg                                application admin
pkg/inst_root                     astring      admin      /export/ipsrepos/Solaris11
pkg/port                          count        admin      81
general                          framework    admin
general/complete                 astring      manifest
general/enabled                  boolean      admin      true
restarter                        framework    <none>      NONPERSISTENT
restarter/logfile                astring      <none>      /var/svc/log/application-pkg-
server:default.log
restarter/contract               count        <none>      121
restarter/start_pid              count        <none>      1055
restarter/start_method_timestamp time         <none>      1379605275.329096000
restarter/start_method_waitstatus integer       <none>      0
restarter/auxiliary_state        astring      <none>      dependencies_satisfied
restarter/next_state             astring      <none>      none
restarter/state                  astring      <none>      online
restarter/state_timestamp        time         <none>      1379605275.332259000
```

Affichage des valeurs d'un instantané particulier

La commande suivante répertorie les clichés que sont disponibles pour cette instance de service. Utilisez ces noms de cliché avec `svccprop` ou `svccfg` pour afficher les valeurs de propriétés définies dans ce périphérique. Seules les instances comporte des clichés. Les services n'a pas les instantanés. Reportez-vous à la section [“Référentiel Clichés” à la page 27](#) pour plus d'informations sur les clichés du référentiel de configuration de service.

```
$ svccfg -s pkg/server:default listsnap
initial
previous
running
start
$ svccfg -s pkg/server:s11 listsnap
previous
running
start
```

Les commandes suivantes indiquent que la valeur de la propriété `pkg/inst_root` était différente dans le cliché précédent.

```
$ svcprop -s previous -p pkg/inst_root pkg/server:s11
/var/share/pkg/repositories/solaris
$ svccfg -s pkg/server:s11
svc:/application/pkg/server:s11> selectsnap previous
[previous]svc:/application/pkg/server:s11> listprop pkg/inst_root
pkg/inst_root astring      /var/share/pkg/repositories/solaris
```

```
[previous]svc:/application/pkg/server:s11> exit
```

Affichage des personnalisations de configuration

La commande `svccfg listcust` affiche les personnalisations à la couche `admin` pour le service spécifié. Utilisez l'option `-L` pour afficher aussi les personnalisations de la couche `site-profile`. La commande suivante indique que les propriétés `pkg/port` et `pkg/inst_root` du service `pkg/server` ont été personnalisées par un administrateur.

```
$ svccfg -s pkg/server:s11 listcust
pkg                application admin
pkg/port           count      admin      81
pkg/inst_root      astring   admin      /export/ipsrepos/Solaris11
general           framework admin
general/complete  astring   manifest
general/enabled   boolean   admin      true
```

La commande `svccfg` suivante indique que la définition de la propriété `config/nodename` est fournie à la couche `manifest`, mais la valeur `solaris` est définie au niveau `admin`.

```
$ svccfg -s identity:node describe config/nodename
config/nodename astring      solaris
Network name of the computer
$ svccfg -p config/nodename -l all svc:/system/identity:node
config/nodename astring manifest ""
config/nodename astring admin solaris
```

La commande `svccfg listcust` affiche également tous les entités *masquées*. Utilisez l'option `-M` pour ne répertorier que les entités masquées. Avant d'utiliser la commande `svccfg delcust`, utilisez la commande `svccfg listcust` pour vérifier ce qui sera supprimé. Reportez-vous à [“Suppression de groupes de propriétés, de propriétés et de valeurs de propriétés” à la page 75](#) et à la page de manuel [smf\(5\)](#) pour une description des entités masquées.

Affichage des paramètres de notification d'événement

La commande `svcs -n` affiche les paramètres de notification des événements FMA, les paramètres de notification des transition d'état SMF à l'échelle du système et les paramètres de notification des transition d'état des instances de service. Reportez-vous à "Paramètres de notification" à la page de manuel [smf\(5\)](#) pour des informations sur ces paramètres.

```
$ svcs -n
Notification parameters for FMA Events
```

```
Event: problem-diagnosed
  Notification Type: smtp
    Active: true
    reply-to: root@localhost
    to: root@localhost

  Notification Type: snmp
    Active: true

  Notification Type: syslog
    Active: true

Event: problem-repaired
  Notification Type: snmp
    Active: true

Event: problem-resolved
  Notification Type: snmp
    Active: true

System wide notification parameters:
svc:/system/svc/global:default:
  Event: to-maintenance
  Notification Type: smtp
    Active: true
    to: sysadmins@example.com

svc:/application/pkg/mirror:default:
  Event: to-maintenance
  Notification Type: smtp
    Active: true
    to: installteam@example.com
```

Trois événements FMA sont affichés : problem-diagnosed, problem-repaired et problem-resolved. Des paramètres de notification peuvent également être configurés pour un quatrième événement : problem-updated.

Modification d'état pour les paramètres de notification à l'échelle du système, le service qui stocke ces paramètres globaux est également répertorié. Ce paramètre applicable à l'ensemble du système est personnalisée paramètre. A l'ensemble du système, ou un rôle standard, valeurs s'appliquent à toutes les instances qui n'ont pas de service personnalisé malgré tout.

Le dernier paramètre décrite ci-dessous est un paramètre pour un service particulier personnalisé instance.

Utilisez la commande `svccfg listnotify` pour afficher uniquement les paramètres de notification de l'événement indiqué. Pour les événements de transition d'état, utilisez l'option `-g` pour afficher les paramètres généraux. La sortie indique également la source de la notification des valeurs de paramètre.

```
$ svccfg listnotify problem-resolved
Event: problem-resolved (source: svc:/system/fm/notify-params:default)
```

```
Notification Type: snmp
Active: true
$ svccfg listnotify -g to-maintenance
Event: to-maintenance (source: svc:/system/svc/global:default)
Notification Type: smtp
Active: true
to: sysadmins@example.com
$ svccfg -s pkg/mirror listnotify to-maintenance
Event: to-maintenance (source: svc:/application/pkg/mirror)
Notification Type: smtp
Active: true
to: installteam@example.com
```

Reportez-vous à la section [“Configuration la notification des événements de transition d'état et FMA” à la page 62](#) pour plus d'informations sur la configuration des notifications d'événements.

♦ ♦ ♦ 3 C H A P I T R E 3

Administration de services

Ce chapitre décrit comment démarrer, arrêter et redémarrer un service et la façon de relire la configuration du service. Il explique également comment configurer le système pour vous informer des événements ou de changements d'état de service FMA. Ces modifications sont des personnalisations du niveau `admin`, comme décrit dans [“Référentiel Couches” à la page 26](#).

La commande qui modifie l'état de service est `svcadm`. La commande `svcadm` fonctionne sur une instance de service. Si vous indiquez un nom de service sans spécifier d'instance et si ce service n'a qu'une seule instance, `svcadm` fonctionne sur cette instance. Si vous indiquez un nom de service sans spécifier d'instance et si ce service a plusieurs instances, ou si vous spécifiez un autre modèle qui correspond à plusieurs instances, `svcadm` envoie un message d'erreur.

Gestion des instances de service SMF

Une instance de service se trouve toujours dans l'un des états décrits dans [“Etats des services” à la page 21](#). Cette section traite de la manière de faire passer une instance à un autre état, mais également de comment valider les valeurs de propriété mises à jour sur l'instantané en cours d'exécution, ainsi que de la procédure de suppression des instances dans la vue normale.

Démarrage d'un service

Une instance de service se trouvant dans l'un des états suivants est déjà activée et n'a pas besoin de faire l'objet d'un démarrage : `degraded`, `maintenance`, `offline`, `online`. Si l'instance que vous souhaitez démarrer se trouve dans l'état `degraded`, `maintenance` ou `offline`, reportez-vous à [“Réparation d'une instance endommagée, hors ligne ou en cours de maintenance” à la page 107](#). Si l'instance que vous souhaitez démarrer se trouve dans l'état `disabled`, activez-la de la manière montrée dans la procédure suivante. Lorsque vous activez une instance, l'agent de redémarrage de cette instance tente de faire passer l'instance à l'état `online`.

▼ Activation d'une instance de service

1. Vérifiez l'état de l'instance et les dépendances.

Vérifiez que l'instance est actuellement désactivée et que toutes ses dépendances obligatoires sont en cours d'exécution (état online ou degraded).

```
$ svcs -l FMRI
```

2. Activez l'instance.

L'agent de redémarrage pour le service tente de faire passer l'instance indiquée à l'état online.

Une instance ne peut être définitivement ou temporairement activée. L'activation permanente persiste après la réinitialisation du système et correspond à la valeur par défaut. L'activation temporaire dure jusqu'à la réinitialisation.

■ Activez l'instance de manière permanente.

```
$ svcadm enable FMRI
```

■ Activez temporairement l'instance.

Utilisez l'option -t pour spécifier l'activation temporaire.

```
$ svcadm enable -t FMRI
```

Si vous souhaitez qu'une instance soit exécutée maintenant mais ne pas l'exécuter à la réinitialisation suivante, assurez-vous que l'instance est désactivée, puis activez temporairement l'instance. Pour vérifier si l'instance est temporairement activée, utilisez la commande `svcs -l` et vérifiez la ligne `enabled` :

```
enabled      true (temporary)
```

■ Activez l'instance de manière synchrone.

Si vous spécifiez l'option -s, `svcadm` active l'instance et attend qu'elle passe à l'état online ou degraded avant le renvoi d'un résultat. La commande `svcadm` renvoie un résultat au moment où l'instance passe à l'état en ligne ou lorsqu'elle détermine qu'une intervention de l'administrateur est nécessaire pour faire passer l'instance à l'état en ligne.

Utilisez l'option -T avec l'option -s pour spécifier un plafond en secondes pour accomplir la transition ou pour déterminer que la transition est impossible.

```
$ svcadm enable -sT 10 FMRI
```

3. Vérifiez que l'instance est en ligne.

```
$ svcs FMRI
```

Si l'instance se trouve dans l'état `degraded`, `maintenance` ou `offline`, reportez-vous à [“Réparation d'une instance endommagée, hors ligne ou en cours de maintenance”](#) à la page 107.

Exemple 3-1 Activation définitivement d'une instance de service

La commande suivante montre que l'instance de service `pkg/mirror:default` est actuellement désactivée et que toutes ses dépendances obligatoires sont en ligne.

```
$ svcs -l pkg/mirror
fmri          svc:/application/pkg/mirror:default
name          IPS Repository Mirror
enabled       false
state         disabled
next_state    none
state_time    September 17, 2013 07:16:52 AM PDT
restarter     svc:/system/svc/restarter:default
manifest      /lib/svc/manifest/application/pkg/pkg-mirror.xml
dependency    require_all/error svc:/milestone/network:default (online)
dependency    require_all/none svc:/system/filesystem/local:default (online)
dependency    optional_all/error svc:/system/filesystem/autofs:default (online)
dependency    require_all/none svc:/application/pkg/repositories-setup (online)
```

La commande suivante vous permet d'activer l'instance `pkg/mirror:default`. Dans ce cas, la commande `svcadm` renvoie un résultat car l'activation de l'instance `pkg/mirror:default` a réussi.

```
$ svcadm enable -sT 10 pkg/mirror:default
$ svcs pkg/mirror
STATE      STIME      FMRI
online     22:03:53  svc:/application/pkg/mirror:default
```

Exemple 3-2 Activation temporaire d'instance de service

La commande suivante montre que l'instance de service `net-snmp:default` est actuellement désactivée et que toutes ses dépendances obligatoires sont en ligne. La dépendance désactivée est facultative.

```
$ svcs -l net-snmp
fmri          svc:/application/management/net-snmp:default
name          net-snmp SNMP daemon
enabled       false
state         disabled
next_state    none
state_time    September 17, 2013 05:56:39 PM PDT
logfile       /var/svc/log/application-management-net-snmp:default.log
restarter     svc:/system/svc/restarter:default
contract_id
manifest      /etc/svc/profile/generic.xml
manifest      /lib/svc/manifest/application/management/net-snmp.xml
```

```
dependency    require_all/none svc:/system/filesystem/local (online)
dependency    optional_all/none svc:/milestone/name-services (online)
dependency    optional_all/none svc:/system/system-log (online)
dependency    optional_all/none svc:/network/rpc/rstat (disabled)
dependency    require_all/restart svc:/system/cryptosvc (online)
dependency    require_all/restart svc:/milestone/network (online)
dependency    require_all/refresh file://localhost/etc/net-snmp/snmp/snmpd.conf (online)
dependency    require_all/none svc:/milestone/multi-user (online)
```

Après avoir activé l'instance avec l'option `-t` comme montré à l'exemple suivant, elle est temporairement activée, est en ligne, et possède un ID de contrat car elle a démarré le processus `snmpd`, comme indiqué par la commande `svcs -p`.

```
$ svcadm enable -t net-snmp:default
$ svcs -l net-snmp
fmri          svc:/application/management/net-snmp:default
name          net-snmp SNMP daemon
enabled       true (temporary)
state         online
next_state    none
state_time    September 17, 2013 05:57:26 PM PDT
logfile       /var/svc/log/application-management-net-snmp:default.log
restarter     svc:/system/svc/restarter:default
contract_id   160
manifest      /etc/svc/profile/generic.xml
manifest      /lib/svc/manifest/application/management/net-snmp.xml
dependency    require_all/none svc:/system/filesystem/local (online)
dependency    optional_all/none svc:/milestone/name-services (online)
dependency    optional_all/none svc:/system/system-log (online)
dependency    optional_all/none svc:/network/rpc/rstat (disabled)
dependency    require_all/restart svc:/system/cryptosvc (online)
dependency    require_all/restart svc:/milestone/network (online)
dependency    require_all/refresh file://localhost/etc/net-snmp/snmp/snmpd.conf (online)
dependency    require_all/none svc:/milestone/multi-user (online)
$ svcs -p net-snmp
STATE      STIME      FMRI
online     17:57:26   svc:/application/management/net-snmp:default
           17:57:26      5022 snmpd
```

Arrêt d'un service

Utilisez la commande `svcadm disable` pour désactiver une instance de service activée ou temporairement désactivée. Une instance désactivée ne peut pas être redémarrée. Vous devez d'abord activer l'instance.

▼ Désactivation d'une instance de service

1. Vérifiez si d'autres services dépendent de cette instance.

a. Répertoriez les services qui dépendent de cette instance.

```
$ svcs -D FMRI
```

b. Vérifiez si le service dépendant requiert cette instance.

Pour chaque résultat de la commande `svcs -D`, utilisez la commande `svcs -l` pour vérifier si la dépendance est obligatoire.

Vous ne devez pas désactiver cette instance si cette instance est une dépendance obligatoire d'un autre service.

2. Désactiver l'instance.

L'agent de redémarrage pour le service tente de faire passer l'instance spécifiée à l'état `disabled`.

Une instance peut être définitivement ou temporairement désactivée. La désactivation permanente persiste après la réinitialisation du système et correspond à la valeur par défaut. La désactivation temporaire dure jusqu'à la réinitialisation.

■ Désactivez définitivement l'instance.

```
$ svcadm disable FMRI
```

■ Désactivez temporairement l'instance.

Utilisez l'option `-t` pour spécifier la désactivation temporaire.

```
$ svcadm disable -t FMRI
```

Si vous souhaitez qu'une instance soit désactivée pour le moment, mais s'exécute à la prochaine réinitialisation, assurez-vous qu'elle est en cours d'exécution (dans l'état `online` ou `degraded`), puis désactivez-la temporairement. Pour vérifier si l'instance est temporairement désactivée, utilisez la commande `svcs -l` et vérifiez la ligne `enabled` :

```
enabled      false (temporary)
```

■ Désactivez l'instance de manière synchrone.

Si vous spécifiez l'option `-s`, `svcadm` désactive l'instance et attend qu'elle passe à l'état `disabled` avant le renvoi d'un résultat. La commande `svcadm` renvoie un résultat au moment où l'instance passe à l'état `disabled` ou lorsqu'elle détermine qu'une intervention de l'administrateur est nécessaire pour faire passer l'instance à l'état `disabled`.

Utilisez l'option `-T` avec l'option `-s` pour spécifier un plafond en secondes pour accomplir la transition ou pour déterminer que la transition est impossible.

```
$ svcadm disable -sT 10 FMRI
```

3. Vérifiez que l'instance est désactivée.

```
$ svcs FMRI
```

Exemple 3-3 Désactivation d'une instance de service

Cet exemple montre que l'instance de service `pkg/update:default` est en ligne à l'initialisation et qu'aucun autre service ne dépend d'elle. La commande `svcadm disable` réussit, l'instance est actuellement à l'état `disabled` et le redémarrage échoue.

```
$ svcs pkg/update
STATE      STIME      FMRI
online     7:18:17   svc:/application/pkg/update:default
$ svcs -D pkg/update:default
STATE      STIME      FMRI
$ svcadm disable pkg/update
$ svcs pkg/update
STATE      STIME      FMRI
disabled   22:51:12   svc:/application/pkg/update:default
$ svcadm restart pkg/update:default
$ svcs pkg/update
STATE      STIME      FMRI
disabled   22:51:12   svc:/application/pkg/update:default
```

Redémarrage d'un service

L'opération de redémarrage redémarre uniquement les instances en cours d'exécution (dans l'état `online` ou `degraded`). Vous devrez peut-être relancer une instance en cours d'exécution car vous avez effectué une modification de configuration qui ne peut pas entraîner une baisse lors de l'instance est en cours d'exécution, par exemple.

Le redémarrage d'une instance de service ne met pas à jour la configuration. La commande `svcadm restart` exécute la méthode d'arrêt de l'instance, puis la méthode de démarrage de l'instance. La commande `svcadm restart` ne valide pas les modifications de propriétés dans l'instantané en cours d'exécution et n'exécute pas la méthode d'actualisation de l'instance. Reportez-vous à [“Relecture de la configuration d'un service” à la page 59](#) pour des informations sur la validation des modifications de configuration dans l'instantané en cours d'exécution.

Le redémarrage du service `manifest-import` est un cas particulier. Le redémarrage du service `manifest-import` importe tous les manifestes ou profils modifiés dans les emplacements standard, valide les modifications dans le référentiel de configuration du service, prend un nouveau cliché en cours d'exécution, puis exécute la méthode d'actualisation des instances modifiées si une telle méthode existe.

▼ Procédure de redémarrage d'une instance de service

1. Vérifiez l'état de l'instance.

L'instance doit se trouver dans l'état `online` ou `degraded`.

```
$ svcs FMRI
```

2. Redémarrez l'instance.

L'agent de redémarrage pour le service tente de faire passer l'instance indiquée à l'état `online`. La plupart des agents de redémarrage mettent en oeuvre l'opération de redémarrage en tant qu'opération d'arrêt suivi d'une opération de démarrage.

■ Redémarrez l'instance.

```
$ svcadm restart FMRI
```

■ Redémarrez l'instance de manière synchrone.

Si vous spécifiez l'option `-s`, `svcadm` redémarre l'instance et attend qu'elle passe à l'état `online`, `degraded` ou `maintenance` avant le renvoi d'un résultat. La commande `svcadm` renvoie un résultat au moment où l'instance passe à l'un de ces états ou lorsqu'elle détermine qu'une intervention de l'administrateur est nécessaire pour faire passer l'instance à l'un de ces états.

Utilisez l'option `-T` avec l'option `-s` pour spécifier un plafond en secondes pour accomplir la transition ou pour déterminer que la transition est impossible.

```
$ svcadm restart -sT 10 FMRI
```

3. Vérifiez que l'instance est démarrée.

Si le redémarrage réussit, l'instance se trouve dans l'état `online`, `degraded` ou `maintenance`. Si l'instance se trouve dans l'état `degraded` ou `maintenance`, reportez-vous à [“Réparation d'une instance endommagée, hors ligne ou en cours de maintenance”](#) à la page 107.

```
$ svcs FMRI
```

Relecture de la configuration d'un service

Lorsque vous modifiez la configuration de service, les modifications n'apparaissent pas immédiatement dans l'instantané en cours d'exécution. Ces modifications sont stockées dans le référentiel de configuration de service, en tant que valeurs de propriétés d'édition ou actuelles. L'opération d'actualisation met à jour l'instantané en cours d'exécution de l'instance de service spécifiée avec les valeurs de la configuration de modification.

Les commandes `svcadm refresh` et `svccfg refresh` réalisent les étapes suivantes :

1. Créez un instantané en cours d'exécution pour valider les propriétés de modification dans l'instantané en cours d'exécution.
2. Exécutez la méthode d'actualisation de l'instance si une telle méthode existe et si l'instance se trouve dans l'état `online` ou `degraded`. La méthode d'actualisation doit notifier l'application des changements apportés. La méthode d'actualisation peut relire les valeurs de propriété depuis l'instantané en cours d'exécution. Même si aucune méthode d'actualisation n'existe, la configuration de l'instantané en cours d'exécution est mise à jour.

La commande `svcadm refresh` fonctionne sur une instance de service. La commande `svccfg refresh` fonctionne sur une instance de service ou sur un service parent. Si un service est spécifié, la commande `svccfg refresh` actualise toutes ses instances. Même si les instantanés ne sont pris que pour les instances de service et non pour les services parent, les propriétés de services parent sont héritées par les instances de service. Les propriétés de service parent modifiées apparaissent dans un instantané d'instance de service si l'instance ne remplace pas ces modifications.

Il se peut que certaines modifications, telles que les modifications des dépendances, prennent effet immédiatement. D'autres modifications ne sont pas prises en compte tant que le service n'est pas redémarré comme décrit dans [“Redémarrage d'un service” à la page 58](#). Les modifications qui ne peuvent pas avoir lieu si l'application est en cours d'exécution requièrent une actualisation suivie d'un redémarrage. Les modifications qui ne peuvent être effectuées si l'application est en cours d'exécution incluent la fermeture ou l'ouverture d'un socket ou la réinitialisation de la variable d'environnement.

Si vous spécifiez l'option `-s` avec la commande `svcadm refresh`, `svcadm` actualise l'instance et attend qu'elle passe à l'état `online`, `degraded` ou maintenance avant le renvoi d'un résultat. La commande `svcadm` renvoie un résultat au moment où l'instance passe à l'un de ces états ou lorsqu'elle détermine qu'une intervention de l'administrateur est nécessaire pour faire passer l'instance à l'un de ces états. Utilisez l'option `-T` avec l'option `-s` pour spécifier un plafond en secondes pour accomplir la transition ou pour déterminer que la transition est impossible.

Suppression d'un service

La commande `svccfg delete` ne supprime pas une instance de service du système. Au lieu de cela, la commande `svccfg delete` masque l'instance. Après l'exécution de la commande `svccfg delete`, le manifeste de service est conservé dans `/lib/svc/manifest`. SMF synchronise le référentiel de configuration de service au contenu du système de fichiers. Etant donné que le manifeste est toujours présent dans le système de fichiers dans un emplacement standard, ces informations de service demeurent stockées dans le référentiel et sont uniquement masquées dans la vue normale. Toutes les personnalisations administratives sont supprimées d'une instance masquée. Reportez-vous à la page de manuel [smf\(5\)](#) pour une description des entités masquées.

Les fichiers prenant en charge une instance de service sont mis à jour lorsque vous utilisez les commandes pkg, même si l'instance de service en question est masquée. Lorsque des fichiers prenant en charge une instance de service sont mis à jour par les commandes pkg, le magasin de données SMF est mis à jour malgré le fait que le service soit encore masqué. Si l'instance de service est démasquée, l'instance en question est déjà mise à jour à partir des fichiers fournis par pkg et aucune action n'est requise.. Pour démasquer une instance de service, reportez-vous à [“Annulation de la suppression d'une instance de service” à la page 61.](#)

▼ Suppression d'une instance de service

1. Vérifiez les dépendants de l'instance à supprimer.

Utilisez la commande `svcs -D` pour montrer les instances dépendant de cette instance. Une fois que vous avez supprimé les instances de cette instance, à une personne à charge ne pourrez peut-être pas exécuter. Utilisez la commande `svcs -l` pour vérifier si l'instance est une dépendance obligatoire de l'instance dépendante.

2. Masquez l'instance.

Utilisez la commande `svccfg delete` pour masquer l'instance dans l'affichage normal. Utilisez la commande `svcs` pour afficher l'état de l'instance. Si l'instance est en cours d'exécution (dans l'état `online` ou `degraded`), utilisez la commande `svccfg delete -f` pour la masquer dans l'affichage normal.

```
$ svcs -H my-svc
disabled          7:25:37 svc:/site/my-svc:default
$ svccfg delete svc:/site/my-svc:default
```

3. Vérifiez si l'instance est masquée.

Utilisez la commande `svccfg listcust -M` pour confirmer que l'instance est masquée. Les commandes telles que `svcs` doivent afficher un message d'erreur disant qu'aucune instance correspondante n'a été trouvée.

```
$ svccfg listcust -M
svc:/site/my-svc:default manifest MASKED
  general                admin    MASKED
  general/complete astring  admin    MASKED
  general/enabled boolean  admin    MASKED true
$ svcs -H my-svc
svcs: Pattern 'my-svc' doesn't match any instances
```

▼ Annulation de la suppression d'une instance de service

1. Confirmez que l'instance est masquée.

Utilisez la commande `svccfg listcust -M` comme illustré dans la procédure précédente.

2. Affichez l'instance.

```
$ svccfg -s svc:/site/my-svc:default delcust
Deleting customizations for instance: default
```

La réimportation du manifeste ne supprime pas le masque.

3. Vérifiez si l'instance n'est plus masquée.

Utilisez la commande `svccfg listcust -M` pour confirmer que l'instance n'est pas masquée. La commande `svcs` doit afficher l'état de l'instance.

Configuration la notification des événements de transition d'état et FMA

Vous pouvez configurer le système pour qu'il vous avertisse lorsqu'une ou lorsqu'un service FMA modifie l'état événement se produit. Vous pouvez choisir une notification par SMTP (Simple Mail Transfer Protocol) ou SNMP (Simple Network Management Protocol).

Par défaut, des dérouterments SNMP sont envoyés sur des transitions de maintenance. Si vous utilisez SNMP pour la notification de transition, vous pouvez configurer des dérouterments supplémentaires pour d'autres transitions d'état.

Les exemples suivants montrent comment définir les paramètres de notification pour les événements SMF et FMA, et indiquent comment supprimer des paramètres de notification.

EXEMPLE 3-4 Configuration d'une notification générale pour un événement d'état de service

La commande suivante crée une notification qui envoie un e-mail lorsque les services passent à l'état maintenance.

```
$ svccfg setnotify -g to-maintenance mailto:sysadmins@example.com
```

-g L'option `-g` définit ce paramètre de notification pour toutes les instances de service sans valeurs personnalisées. Toutes les instances de service mises à jour sont actualisées. L'option `-g` ne peut être utilisée pour définir les notifications pour transition d'état et non pour les événements FMA.

to-maintenance L'argument `to-maintenance` est un événement de transition d'état tel que décrit dans "Paramètres de notification" à la page de manuel [smf\(5\)](#). Spécifiez uniquement le nom d'état pour inclure les transitions `to-state` et `from-state`. Cet événement peut également être dans une liste de transition séparée par des virgules.

mailto: L'argument `mailto` spécifie la notification que vous souhaitez recevoir pour l'événement spécifié. Cet argument pourrait aussi spécifier `snmp`. Une valeur de notification `snmp` doit être soit `snmp:active`, soit `snmp:inactive`. Une valeur de notification `mailto` peut être `mailto:active` ou `mailto:inactive`, en plus de la forme montrée dans cet exemple. La définition d'un paramètre de notification remplace toute valeur existante de l'événement. Les paramètres `active` et `inactive` ne remplacent pas les valeurs existantes ; elles déterminent si la notification existante est appliquée ou non pour l'événement spécifié.

EXEMPLE 3-5 Configuration d'une notification pour une instance de service spécifiée

La commande suivante crée une notification qui envoie un e-mail lorsque les transitions du service `pkg/mirror` passent à l'état maintenance.

```
$ svccfg -s pkg/mirror setnotify to-maintenance mailto:installteam@example.com
```

La commande suivante crée une notification qui envoie un e-mail lorsque le service `http:apache22` passe hors de l'état online.

```
$ svccfg -s http:apache22 setnotify from-online mailto:webservices@example.com
```

EXEMPLE 3-6 Configuration d'une notification pour un événement FMA

L'argument `problem-diagnosed` est un événement FMA. Cet argument peut être dans une liste séparée par des virgules des événements FMA. Reportez-vous à la liste des événements FMA dans "Paramètres de notification" à la page de manuel [smf\(5\)](#).

```
$ svccfg setnotify problem-diagnosed mailto:IT@example.com
```

EXEMPLE 3-7 Suppression de paramètres de notification

Les commandes suivantes suppriment les paramètres de notification définis dans les exemples précédents.

```
$ svccfg delnotify -g to-maintenance
$ svccfg -s pkg/mirror delnotify to-maintenance
$ svccfg setnotify problem-diagnosed mailto:root@localhost
```


Configuration des services

Stocke les données de configuration SMF dans le référentiel de configuration de service. Services implique la configuration de modifier les données SMF dans le référentiel de configuration, puis en cours d'exécution de la validation (commit) instantané modifications dans la. Ce chapitre décrit la procédure pour modifier les données dans le référentiel de configuration. Pour l'affichage des données dans le référentiel de configuration, reportez-vous à [“Inspection de la configuration du service” à la page 42](#). Pour valider les modifications de configuration dans le cliché en cours d'exécution, voir [“Relecture de la configuration d'un service” à la page 59](#).

Chaque service et instance de service stocke les données de configuration dans les propriétés, eux-mêmes organisés selon les groupes de propriétés. Modifier les données dans le référentiel de configuration inclut la modification de support. pour plus d'informations des valeurs de propriété de service, groupe de propriétés et les propriétés, ce qui crée de nouvelles instances d'un service, et l'application d'un profil. La suppression des personnalisations modification de la configuration et permet de revenir également référentiel inclut des clichés.

Ce chapitre indique également comment modifier un service inetd.

Les modifications apportées à la configuration SMF peuvent être consignées à l'aide de la structure d'audit d'Oracle Solaris. Pour plus d'informations, reportez-vous à la section "Configuration du service d'audit (liste des tâches) dans *Gestion de l'audit dans Oracle Solaris 11.2*

Utilisation de la commande de configuration de service

La commande `svccfg` manipule les données dans le référentiel de configuration de service. Toutes les modifications que vous apportez avec la commande `svccfg` sont enregistrées dans la couche admin. Pour plus d'informations sur les couches, reportez-vous à la section [“Référentiel Couches” à la page 26](#). Les modifications apportées avec la commande `svccfg` sont enregistrées dans le référentiel de configuration en tant que valeurs de propriété actuelles ou de modification ; elles ne s'affichent pas immédiatement dans le cliché en cours d'exécution. Lorsque vous modifiez des données de configuration, utilisez la commande `svcadm refresh` ou `svccfg refresh` pour valider les nouvelles valeurs dans le cliché en cours d'exécution.

La gestion séparée des données nouvellement modifiées et de l'instantané en cours d'exécution vous permet d'effectuer plusieurs modifications, puis d'enregistrer ensemble toutes les modifications dans l'instantané en cours d'exécution. Tant que vous êtes dans le processus qui consiste à rendre les valeurs de propriété plusieurs certaines modifications effectuées incohérente risquent de ne pas être compatibles en cours d'exécution ou instantané, mais l'est pas modifié. Une fois les modifications apportées, effectuer une actualisation.

Vous pouvez utiliser la commande `svccfg` des manières suivantes :

- Utilisez la commande `svccfg editprop` pour appeler un éditeur sur le groupe de propriétés et les propriétés de l'entité sélectionnée.
- Saisissez une commande `svccfg` complète dans la ligne de commande, en spécifiant les sous-commandes telles que `setprop`.
- Saisissez uniquement `svccfg` ou `svccfg -s FMRI` dans la ligne de commande pour lancer une session interactive.
- Spécifiez l'option `-f` pour lire les commandes `svccfg` depuis un fichier.

Appel d'un éditeur de propriétés

L'appel de la commande `svccfg` selon l'exemple suivant ouvre un éditeur pour les propriétés de l'entité sélectionnée. Cette forme de la commande `svccfg` peut être très rapide et pratique pour modifier plusieurs valeurs de propriété. Pour la sous-commande `editprop`, il est impératif de spécifier une entité avec l'option `-s`.

```
$ svccfg -s pkg/server:s11 editprop
```

Un fichier de commandes `setprop` pour les valeurs actuelles de chaque propriété de l'entité indiquée s'ouvre dans l'éditeur spécifié par la variable d'environnement `VISUAL`. Si `VISUAL` n'est pas définie, c'est l'éditeur spécifié par `EDITOR` qui s'ouvre. Si ni `VISUAL` ni `EDITOR` ne sont définies, le fichier de propriété s'ouvre dans `vi`.

Chaque ligne du fichier est précédée d'un caractère de commentaire. Pour modifier la valeur d'une propriété dans la configuration de modification `svccfg`, supprimez le caractère de commentaire, modifiez la valeur et enregistrez le fichier. Pour modifier la valeur d'une propriété du cliché en cours d'exécution, supprimez le caractère de commentaire à la dernière ligne du fichier, qui contient la sous-commande `refresh`.

La liste suivante présente un exemple partiel d'un fichier créé par la sous-commande `editprop` :

```
##
## Change property values by removing the leading '#' from the
## appropriate lines and editing the values. svccfg subcommands
## such as delprop can also be added to the script.
```

```
##
## Property group "pkg"
## The following properties are defined in the selected instance
## (svc:/application/pkg/server:s11)

# setprop pkg/port = count: 81
# setprop pkg/inst_root = astring: /export/ipsrepos/Solaris11

## The following properties inherit from the parent service
## (svc:/application/pkg/server)

# ...

## Property group "pkg_bui"

# ...

## Property group "pkg_secure"

# ...

## Uncomment to apply these changes to this instance.
# refresh
```

Conformément aux instructions dans le fichier, vous pouvez ajouter d'autres sous-commandes que `setprop`. Par exemple, vous pouvez ajouter une commande `delprop`. Certains groupes de propriétés, tels que `framework` et `dependency`, ne sont pas affichés par défaut. Spécifiez `editprop -a` pour afficher toutes les propriétés.

Commandes de cette temporaire le fichier non commentée sont exécutés lors de l'enregistrement et de quitter la session de modification.

Appel de `svccfg` de manière interactive ou avec un fichier

L'appel interactif de la commande `svccfg` comme dans l'exemple suivant peut s'avérer pratique pour réaliser plusieurs opérations de configuration.

```
$ svccfg
svc:> select pkg/server
svc:/application/pkg/server> list
:properties
default
svc:/application/pkg/server> add s11
svc:/application/pkg/server> select s11
svc:/application/pkg/server:s11> setprop pkg/inst_root=/export/ipsrepos/Solaris11
svc:/application/pkg/server:s11> setprop pkg/port=81
svc:/application/pkg/server:s11> unselect
```

```

svc:/application/pkg/server> add oss
svc:/application/pkg/server> select oss
svc:/application/pkg/server:oss> setprop pkg/inst_root=/export/ipsrepos/SolarisStudio
svc:/application/pkg/server:oss> setprop pkg/port=82
svc:/application/pkg/server:oss> unselect
svc:/application/pkg/server> list
:properties
default
s11
oss
svc:/application/pkg/server> refresh
svc:/application/pkg/server> select pkg/mirror:default
svc:/application/pkg/mirror:default> listprop config/crontab_period
config/crontab_period astring      "30 2 25 * *"
svc:/application/pkg/mirror:default> setprop config/crontab_period="00 3 25 * *"
svc:/application/pkg/mirror:default> refresh
svc:/application/pkg/mirror:default> exit
$

```

Les commandes émises aux invites interactives de l'exemple précédent peuvent aussi être fournies dans un fichier pour exécution par le biais d'une commande telle que la suivante.

```
$ svccfg -f cfgpkgrepos
```

Définition des valeurs de propriété

Les commandes suivantes définir les valeurs de propriété, procédez comme suit :

```
svccfg setprop
```

Modifie la valeur d'une propriété.

```
svccfg addpropvalue
```

Ajoute une valeur à une propriété à valeurs multiples.

```
svccfg setenv
```

Change la valeur d'une variable d'environnement pour un environnement d'exécution de processus de service.

N'oubliez pas d'utiliser la commande `svccfg refresh` ou `svcadm refresh` pour valider les modifications de la configuration dans l'instantané en cours d'exécution.

EXEMPLE 4-1 La définition d'une valeur simple

Pour une utilisation basique de `setprop`, spécifiez un *pg/name* pour le service ou l'instance sélectionné, où *pg* est le nom du groupe de propriétés et *name* celui de la propriété, puis

spécifiez la nouvelle valeur après un signe égal. Si la propriété existe déjà ou est la méthode de réponse par modèle, il est inutile d'indiquer le type de propriété.

```
$ svccfg -s pkg/server:s11 setprop pkg/port=81
```

EXEMPLE 4-2 Une Valeur qui contient un paramètre deux points de caractères

Si la valeur de la propriété contient le signe deux-points (:), spécifiez le type de propriété comme indiqué dans l'exemple suivant, où le type est `astring` :

```
$ svccfg -s system-repository:default setprop config/http_proxy = astring:
https://proxyURI
```

Utilisez la sous-commande `listprop` pour trouver le type de la propriété que vous souhaitez définir.

```
$ svccfg -s system-repository:default listprop config/http_proxy
config/http_proxy astring
```

EXEMPLE 4-3 Embedded une Valeur qui contient la définition de Spaces

Utilisez des guillemets doubles pour définir une valeur qui contient d'espaces. En fonction de votre shell, vous pouvez être amené à mettre la chaîne entre guillemets mis entre apostrophes.

```
$ svccfg -s pkg/mirror setprop config/crontab_period = "00 3 25 * *"
$ svccfg -s pkg/mirror setprop config/crontab_period = "'00 3 25 * *'"
```

Utiliser des guillemets pour définir une valeur qui contient des barres obliques inverses avec des guillemets ou utilisez une barre oblique inverse, et comme caractère d'échappement pour tout des guillemets ou des barres obliques inverses.

EXEMPLE 4-4 Une valeur que l'option Est la définition d'un jeu de valeurs (LOV)

Utilisez des parenthèses pour indiquer un ensemble de valeurs en tant que valeur unique. En fonction de votre shell, vous pouvez être amené à placer la valeur définie dans les guillemets simples.

```
$ svccfg -s dns/client setprop config/nameserver = (10.0.0.1 192.168.0.1)
$ svccfg -s dns/client setprop config/nameserver = '(10.0.0.1 192.168.0.1)'
$ svccfg -s dns/client listprop config/nameserver
config/nameserver net_address 10.0.0.1 192.168.0.1
```

Utilisez la sous-commande `describe` pour connaître le nombre de valeurs autorisées dans l'ensemble de valeurs.

```
$ svccfg -s dns/client describe -v config/nameserver
config/nameserver net_address 10.0.0.1 192.168.0.1
type: net_address
```

```
required: false
The IP address of a DNS nameserver to be used by the resolver.
visibility: readwrite
minimum number of values: 1
maximum number of values: 3
value: 10.0.0.1
value: 192.168.0.1
```

EXEMPLE 4-5 Ajout d'une valeur

Utilisez la sous-commande `addpropvalue` pour ajouter la valeur donnée à la propriété spécifiée pour le service ou l'instance de service sélectionné. La nouvelle valeur est ajouté à la fin de la liste de valeurs de propriété existante pour la propriété.

```
$ svcprop -p keymap/layout keymap:default
US-English
$ svccfg -s keymap:default addpropvalue keymap:layout UK-English
$ svccfg -s keymap:default listprop keymap:layout
keymap/layout astring      "US-English" "UK-English"
```

Dans l'exemple précédent de `setprop`, toutes les valeurs de l'ensemble de valeurs doivent être spécifiées d'un coup. Si une seule valeur est spécifiée, cette valeur devient le nouvel ensemble d'une seule valeur. Dans cet exemple d'`addpropvalue`, les nouvelles valeurs sont différentes. Pour accéder à ces nouvelles valeurs, vous devez utiliser la fonction `libscf_scf_iter_property_values()` to iterate over the values. `listprop` répertorie les deux valeurs, `describe` ne répertorie que la première et signale que le nombre maximal autorisé de valeurs pour cette propriété est un.

```
$ svccfg -s keymap:default describe -v keymap/layout
keymap/layout astring      US-English
  type: astring
  required: true
  The keyboard layout
  visibility: readwrite
  minimum number of values: 1
  maximum number of values: 1
  value: US-English
```

▼ Modification d'une valeur de propriété ttymon

Cette procédure montre comment modifier les paramètres transmis à `ttymon`.

1. Identifier le service à modifier.

La page de manuel [ttymon\(1M\)](#) affirme que le service à modifier est `svc:/system/console-login`. La page de manuel `ttymon(1M)` contient aussi des descriptions des propriétés du groupe `ttymon`.

La commande ci-dessous affiche plusieurs instances du service `console-login` dans cette image et indique que l'instance `default` est la seule actuellement en ligne :

```
$ svcs console-login
STATE      STIME      FMRI
disabled   10:49:43   svc:/system/console-login:terma
disabled   10:49:43   svc:/system/console-login:termb
online     10:50:54   svc:/system/console-login:default
```

2. Identifier la propriété à modifier.

La commande suivante permet d'afficher le nom, le type de données, la valeur et une brève description de chaque propriété du groupe de propriétés `ttymon` dans l'instance `default` :

```
$ svccfg -s console-login:default describe ttymon
ttymon          application
ttymon/device   astring      /dev/console
    The terminal device to be used for the console login prompt.
ttymon/terminal_type astring
    Sets the initial value of the TERM environment variable
```

La sortie ci-dessus n'affiche pas de valeur pour la propriété `terminal_type`. La commande suivante confirme que la valeur de la propriété `ttymon/terminal_type` de l'instance `console-login:default` est actuellement zéro :

```
$ svcprop -p ttymon/terminal_type console-login:default
""
```

3. Modifier la valeur de la propriété.

Saisissez la commande suivante afin de faire passer la valeur de la propriété `ttymon/terminal_type` de l'instance `console-login:default` à `xterm` :

```
$ svccfg -s system/console-login:default setprop ttymon/terminal_type=xterm
```

4. Dans la validation (commit) en cours d'exécution de la nouvelle valeur instantané.

La sortie suivante montre que la valeur de la propriété `terminal_type` est modifiée dans la configuration d'édition mais pas dans le cliché en cours d'exécution :

```
$ svccfg -s console-login:default listprop ttymon/terminal_type
ttymon/terminal_type astring      xterm
$ svcprop -p ttymon/terminal_type console-login:default
""
```

Après actualisation de la valeur de la propriété de l'instance de service, l'en cours d'exécution instantané sont modifiés dans le, procédez comme suit :

```
$ svcadm refresh console-login:default
$ svcprop -p ttymon/terminal_type console-login:default
xterm
```

▼ Procédure de modification d'une variable d'environnement pour un service de processus d'environnement

Cette procédure montre comment définir une valeur pour une variable d'environnement dans l'environnement dans lequel processus lancés par le service sera exécuté.

L'exemple dans cette procédure montre comment modifier des variables d'environnement `cron` afin d'obtenir de l'aide pour le débogage.

1. Assurez-vous que le service est en cours d'exécution.

La sortie suivante indique que le service `cron` est en ligne et qu'un processus `cron` est en cours d'exécution.

```
$ svcs -p cron
STATE          STIME      FMRI
online         10:24:05   svc:/system/cron:default
               10:24:05      1089 cron
```

2. Définissez des variables d'environnement.

La sous-commande `setenv` définit une variable d'environnement pour l'environnement où s'exécute un processus démarré par un service ou une instance de service.

Exécutez la commande suivante pour vérifier les valeurs en cours du vous souhaitez définir les variables d'environnement :

```
$ pargs -e `pgrep -f /usr/sbin/cron`
```

Les variables d'environnement qui sont définies dans cet exemple ne disposent pas de valeurs en cours.

Les commandes suivantes définissent les variables d'environnement `UMEM_DEBUG` et `LD_PRELOAD` pour le processus `/usr/sbin/cron` démarré par l'instance de service `svc:/system/cron:default` :

```
$ svccfg -s system/cron:default setenv UMEM_DEBUG default
$ svccfg -s system/cron:default setenv LD_PRELOAD libumem.so
```

3. Actualisez et redémarrez le service

La modification d'une valeur de variable d'environnement doit être redémarré, ainsi qu'un doit être actualisé.

```
$ svcadm refresh system/cron:default
$ svcadm restart system/cron:default
```

4. Vérifiez que la modification a été effectuée.

La sortie suivante indique que le service a été redémarré dispose d'un nouveau, l'ID processus, ainsi que les deux processus variables d'environnement sont définies pour cet environnement de traitement.

```
$ svcs -p cron
STATE          STIME      FMRI
online         9:24:39   svc:/system/cron:default
               9:24:39      5601  cron

$ svcprop -g method -p environment system/cron:default
start/environment astring LD_PRELOAD=libumem.so UMEM_DEBUG=default
$ pargs -e `pgrep -f /usr/sbin/cron`
5601: /usr/sbin/cron
envp[0]: LOGNAME=root
envp[1]: LD_PRELOAD=libumem.so
envp[2]: PATH=/usr/sbin:/usr/bin
envp[3]: SMF_FMRI=svc:/system/cron:default
envp[4]: SMF_METHOD=start
envp[5]: SMF_RESTARTER=svc:/system/svc/restarter:default
envp[6]: SMF_ZONENAME=global
envp[7]: UMEM_DEBUG=default
```

Voir aussi La sous-commande `unsetenv` annule la définition d'une variable d'environnement pour un processus démarré par un service ou une instance de service.

Ajout de groupes, des propriétés et valeurs de propriété

Les commandes suivantes ajoutent les propriétés et les groupes de propriétés, procédez comme suit :

```
svccfg setprop
svccfg addpropvalue
```

Ajoute la propriété dont la valeur est en cours de définition si la propriété n'existe pas déjà.

```
svccfg addpg
```

Ajoute un nouveau groupe de propriétés à un service ou d'une instance de service.

N'oubliez pas d'utiliser la commande `svccfg refresh` ou `svcadm refresh` pour valider les modifications de la configuration dans l'instantané en cours d'exécution.

EXEMPLE 4-6 Utilisation de `addpg` pour créer un nouveau groupe de propriétés

Utilisez la sous-commande `addpg` pour ajouter un groupe de propriétés au service ou à l'instance de service sélectionné.

```
svccfg -s FMRI addpg name type [flags]
```

<i>type</i>	Par convention, la valeur de <i>type</i> est généralement <i>application</i> . Reportez-vous au Chapitre 5, Utilisation de SMF pour contrôler votre application pour plus d'informations sur les types de groupes de propriétés.
<i>flags</i>	Spécifiez <i>P</i> pour la valeur de <i>flags</i> pour enregistrer le groupe de propriétés et les propriétés ajoutées sous forme non permanente. Si <i>P</i> est spécifié, ce groupe de propriétés et les propriétés qu'il contient seront automatiquement supprimés à la réinitialisation ; la valeur <i>P</i> est un alias pour <code>SCF_PG_FLAG_NONPERSISTENT</code> . Voir la page de manuel <code>scf_service_add_pg</code> (3SCF).

```
$ svccfg -s svc:/site/my-svc addpg config application
$ svccfg -s my-svc listprop config
config application
$ svccfg -s my-svc:default listprop config
$
```

Dans cet exemple, l'administrateur a ajouté le groupe de propriétés `config` au service parent, `my-svc`, mais pas à l'instance, `my-svc:default`. La commande `listprop` montre que le groupe de propriétés `config` n'existe pas dans l'instance de service.

EXEMPLE 4-7 Utilisation de `setprop` pour créer une nouvelle propriété

Utilisez la sous-commande `setprop` pour définir une valeur de propriété, comme décrit dans la section [“Définition des valeurs de propriété” à la page 68](#). Si le groupe de propriétés n'existe pas encore dans le service ou l'instance sélectionné, le groupe de propriétés est créé si le *type* et les indicateurs se trouvent dans les définitions des matrices. Si la propriété n'existe pas encore dans le service ou l'instance sélectionné, vous devez indiquer le *type* de propriété.

```
$ svccfg -s my-svc:default setprop config/vendor = astring: vendora
$ svccfg -s my-svc:default listprop config/vendor
config/vendor astring      vendora
```

EXEMPLE 4-8 Utilisation d'`addpropvalue` pour créer une nouvelle propriété

Utilisez la sous-commande `addpropvalue` pour ajouter une valeur de propriété, comme décrit dans la section [“Définition des valeurs de propriété” à la page 68](#). Si le groupe de propriétés n'existe pas encore dans le service ou l'instance sélectionné, le groupe de propriétés est créé si le *type* et les indicateurs se trouvent dans les définitions des matrices. Si la propriété n'existe pas encore dans le service ou l'instance sélectionné, vous devez indiquer le *type* de propriété.

```
$ svccfg -s my-svc:default addpropvalue config/vendor astring: vendorb
$ -s my-svc:default addpropvalue config/customer astring: acustomer
$ svccfg -s my-svc:default listprop config
config          application
config/vendor   astring      "vendora" "vendorb"
config/customer astring      acustomer
```

Suppression de groupes de propriétés, de propriétés et de valeurs de propriétés

Les valeurs de propriété les commandes suivantes, supprimer et les groupes de propriétés, les propriétés :

`svccfg setprop`

Supprimer toutes les valeurs d'une propriété.

`svccfg delpropvalue`

Supprimer toutes les valeurs de la propriété spécifiée qui correspondent au critère indiqué.

`svccfg delprop`

Supprimez une propriété.

`svccfg delpg`

Supprimer un groupe de propriétés.

`svccfg delcust`

Supprimer personnalisations administratives.

N'oubliez pas d'utiliser la commande `svccfg refresh` ou `svcadm refresh` pour valider les modifications de la configuration dans l'instantané en cours d'exécution.

Suppression de la configuration de l'administration

Les modifications apportées à la configuration à l'aide des commandes `svccfg` ou `libscf` modifient uniquement la couche `admin` du référentiel de configuration de service. Pour plus d'informations sur les couches, reportez-vous à la section [“Référentiel Couches” à la page 26](#). Lorsque vous supprimez la configuration qui n'est définie que dans la couche `admin` et n'existe dans aucune autre couche, cette configuration disparaît. Les commandes permettant d'afficher la configuration ne montreront plus la configuration supprimée, même si vous utilisez l'option `-l` pour afficher toutes les couches du référentiel de configuration de service. Reportez-vous à la section [“Suppression de la configuration non administrative” à la page 77](#) pour plus d'informations sur la suppression de la configuration qui existe dans d'autres couches.

EXEMPLE 4-9 Propriétés de la suppression de toutes les valeurs d'un

Utilisez la sous-commande `setprop` comme décrit dans la section [“Définition des valeurs de propriété” à la page 68](#). Pour supprimer toutes les valeurs d'une propriété, vous ne devez

pas indiquer n'importe quel type ou une valeur. Les valeurs sont supprimés, mais la propriété existe toujours.

```
$ svccfg -s my-svc:default setprop config/vendor =
$ svccfg -s my-svc:default listprop config/vendor
config/vendor    astring
```

EXEMPLE 4-10 De la suppression d'une propriété Toutes les valeurs concordantes

Utilisez la sous-commande `delpropvalue` pour supprimer toutes les valeurs de la propriété nommée qui correspondent au modèle donné.

```
$ svccfg -s my-svc:default setprop config/tool = astring: '(hammer tongs wrench)'
$ svccfg -s my-svc:default listprop config
config          application
config/customer astring    acustomer
config/vendor   astring    "vendora" "vendorb"
config/tool     astring    "hammer tongs wrench"
$ svccfg -s my-svc:default delpropvalue config/vendor '*b'
$ svccfg -s my-svc:default delpropvalue config/tool '*tong*'
$ svccfg -s my-svc:default listprop config
config          application
config/customer astring    acustomer
config/vendor   astring    vendora
config/tool     astring    "hammer tongs wrench"
$ # config/tool is a single value that is a value set
$ svccfg -s my-svc:default delpropvalue config/tool '*tong*'
$ svccfg -s my-svc:default listprop config
config          application
config/customer astring    acustomer
config/vendor   astring    vendora
config/tool     astring
```

EXEMPLE 4-11 Suppression d'une propriété

Utilisez la sous-commande `delprop` pour supprimer la propriété nommée du service ou de l'instance de service sélectionné.

```
$ svccfg -s my-svc:default delprop config/tool
$ svccfg -s my-svc:default listprop config
config          application
config/customer astring    acustomer
config/vendor   astring    vendora
```

EXEMPLE 4-12 La suppression d'un groupe de propriétés

Les sous-commandes `delpg` et `delprop` peuvent supprimer un groupe de propriétés. La sous-commande `delpg` supprime la propriété nommée du service ou de l'instance de service

sélectionné. La sous-commande `delprop` supprime le groupe de propriétés nommé en l'absence de propriété nommée.

```
$ svccfg -s my-svc:default delpg config
$ svccfg -s my-svc:default listprop config
$
```

EXEMPLE 4-13 Suppression de personnalisations

La sous-commande `delcust` supprime les personnalisations réalisées par un administrateur sur le service ou l'instance de service sélectionné. Avant d'utiliser la sous-commande `delcust`, utilisez la sous-commande `listcust` avec le même modèle ou la même option pour voir ce qui sera supprimé. Si un modèle n'est indiquée, à la propriété doit correspondre à une propriété ou d'un groupe.

```
$ svccfg -s my-svc:default listcust
config                                application admin
config/customer                      astring      admin      acustomer
config/vendor                       astring      admin      "vendora" "vendorb"
config/tool                         astring      admin      "hammer tongs wrench"
$ svccfg -s my-svc:default listcust '*tool'
config/tool                         astring      admin      "hammer tongs wrench"
$ svccfg -s my-svc:default delcust '*tool'
Deleting customizations for property: config/tool
$ svccfg -s my-svc:default listcust '*tool'
$ svccfg -s my-svc:default listcust
config                                application admin
config/customer                      astring      admin      acustomer
config/vendor                       astring      admin      "vendora" "vendorb"
```

Suppression de la configuration non administrative

La configuration des couches `site-profile`, `system-profile` et `manifest` du référentiel de configuration de service est définie dans des manifestes de service et des fichiers de profil. Pour plus d'informations sur les couches, reportez-vous à la section [“Référentiel Couches” à la page 26](#). SMF synchronise le référentiel de configuration de service au contenu du système de fichiers. Les configurations qui est défini dans un manifeste ou d'un fichier de profil dans l'emplacement standard est toujours présent dans le système de fichiers après personnalisation administrative, y compris après la date en cours de suppression et qu'elle est toujours stockées dans le référentiel de configuration de service. Configuration qui est défini dans un manifeste ou d'un nouveau profil héberge des lot d'offres prennent en charge. Lorsque vous supprimez une configuration avec `bundle support`, les informations ne sont pas supprimées du système de fichiers mais *masquées* de sorte qu'elles ne sont plus visibles dans l'affichage normal. Reportez-vous à la page de manuel [smf\(5\)](#) pour une description des entités masquées.

La prise en charge la suppression de la configuration est dès lors que celui-ci a une personnalisation administrative lot d'offres. Dans ce cas, la sous-commande `delcust` *démasque* la configuration, sans supprimer quoi que ce soit. Utilisez la sous-commande `listcust -M` pour afficher la configuration masquée. Utilisez la sous-commande `delcust -M` pour démasquer la configuration ou pour annuler la suppression ou le masquage de la configuration.

EXEMPLE 4-14 Prise en charge la suppression d'un lot d'offres configuration qui a

Dans “[Suppression de la configuration de l'administration](#)” à la page 75, le groupe de propriétés `config` du service `my-svc` existait uniquement dans la couche `admin`. Le groupe de propriétés `config` n'existait dans aucun manifeste ou profil. Lorsque ces propriétés ont été supprimés, ils ont été passée du système. L'exemple ci-dessous illustre le résultat différent lorsque vous supprimez la configuration qui a lot d'offres prennent en charge.

La propriété est définie dans le manifeste du service, procédez comme suit :

```
$ svccfg -s pkg/server listprop -l all pkg/inst_root
pkg/inst_root astring      admin          /export/ipsrepos/Solaris11
pkg/inst_root astring      manifest       /var/pkgrepo
$ svccfg -s pkg/server delprop pkg/inst_root
```

Après la suppression, la propriété n'est pas affiché à l'aide de la commande `listprop` sans option. Etant donné que la propriété `has bundle support`, elle existe encore dans le référentiel de configuration de service et peut être affichée à l'aide des options `-l` ou `-M` de la sous-commande `listprop`.

```
$ svccfg -s pkg/server listprop pkg/inst_root
$ svccfg -s pkg/server listprop -l all pkg/inst_root
pkg/inst_root astring      admin          MASKED /export/ipsrepos/Solaris11
pkg/inst_root astring      manifest       MASKED /var/pkgrepo
$ svccfg -s pkg/server listcust -M
pkg/inst_root astring      admin          MASKED /export/ipsrepos/Solaris11
```

EXEMPLE 4-15 Annulation de masquage Configuration

La propriété lorsque vous les personnalisations le masquage ne sont plus là, les deux, procédez comme suit :

- La propriété n'est plus masquée ou masqués.
- Son la propriété de valeur personnalisé ne dispose plus.

```
$ svccfg -s pkg/server delcust -M
Deleting customizations for property: pkg/inst_root
$ svccfg -s pkg/server listprop -l all pkg/inst_root
pkg/inst_root astring      manifest       /var/pkgrepo
$ svccfg -s pkg/server listprop pkg/inst_root
pkg/inst_root astring      /var/pkgrepo
```

Ajout d'instances de service

Autoriser plusieurs instances des configurations d'un service d'un service de s'exécuter simultanément. Le nombre d'instances du service de configuration et de personnaliser de service communs héritent.

Utilisez la sous-commande `add` pour créer une nouvelle entité portant le nom indiqué en tant qu'enfant du service sélectionné.

```
$ svcs -Ho inst pkg/server
default
$ svccfg -s pkg/server add s11
$ svcs -Ho inst pkg/server
default
s11
```

Rétablissement des instantanés

Chacune des opérations suivantes : crée un nouvel instantané en cours d'exécution

- `svcadm restart manifest-import`
- `svcadm refresh`
- `svccfg refresh`

La sous-commande `revert` rétablit les personnalisations administratives (couche `admin`) de l'instance spécifiée par l'option `-s`, ainsi que les services associés, aux valeurs enregistrées dans le cliché nommé ou le cliché actuellement sélectionné. Utilisez la sous-commande `listsnap` pour afficher la liste des clichés possibles pour cette instance de service. Utilisez la sous-commande `selectsnap` pour sélectionner un cliché en mode interactif.

```
$ svcprop -p pkg/inst_root pkg/server:default
pkg/inst_root astring /export/ipsrepos/Solaris11
$ svccfg -s pkg/server:default listsnap
initial
previous
running
start
$ svcprop -s previous -p pkg/inst_root pkg/server:default
pkg/inst_root astring /var/pkgrepo
```

Etant donné que la sous-commande `revert` rétablit toutes les personnalisations administratives, répertoriez-les toutes et examinez leurs valeurs avant le rétablissement.

```
$ svcprop -s previous -l admin pkg/server:default
pkg/inst_root astring /var/pkgrepo
```

```
$ svccfg -s pkg/server:default revert previous
$ svcadm refresh pkg/server:default
$ svcprop -p pkg/inst_root pkg/server:default
pkg/inst_root astring /var/pkgrepo
```

Importation et application de manifestes et de profils

Lorsque vous redémarrez le service `manifest-import`, les manifestes des emplacements standard sont importés et les profils des emplacements standard sont appliqués dans la mesure où ils sont nouveaux ou modifiés. Reportez-vous à la section [“Les lots d'offres de service” à la page 25](#) pour connaître les emplacements standard des manifestes et des profils. Dans le cas d'une importation un manifeste ou dans le profil des résultats de l'application d'un service démarré ou arrêté, la méthode appropriée est exécutée si elle existe.

Si vous spécifiez un fichier dans un emplacement standard à la commande `svccfg import`, cela redémarre le service `manifest-import`.

Il est recommandé de mettre vos fichiers de manifeste et de profil dans les emplacements standard et de redémarrer le service `manifest-import` plutôt que d'utiliser les commandes `svccfg import` ou `svccfg apply`.

```
$ svcadm restart manifest-import
```

Lorsque vous redémarrez le service `manifest-import`, la configuration des profils et manifestes dans les emplacements standard est appliquée à la couche `manifest`, `system-profile` ou `site-profile` des instances concernées ; celles-ci sont actualisées et validées, et un nouveau cliché est créé.

Lorsque vous importez ou appliquez des profils et des manifestes dans des emplacements non standard, la configuration est appliquée à la couche `admin` des instances concernées. Les emplacements de l'aide de non-standard ou il est fortement déconseillé d'utiliser la configuration initiale de livraison pour les souscriptions par défaut. Pour procéder à de nombreuses modifications de configuration, l'importation ou l'application à partir d'un emplacement non standard peut s'avérer plus facile que l'exécution d'un grand nombre de commandes, mais vous perdez les mécanismes de gestion automatique du service `manifest-import`. Pour gérer la prestation des services, le service `manifest-import` doit connaître les emplacements et les états attendus.

Configuration de plusieurs systèmes

Pour mettre en oeuvre la même configuration sur plusieurs systèmes, créez un profil qui indique les services SMF à activer et les valeurs des propriétés, puis placez ce profil dans le

répertoire `site` de chaque système. Comme indiqué dans [“Référentiel Couches” à la page 26](#), les personnalisations locales ont la priorité sur les valeurs fournies à l'installation du système et les personnalisations de la couche `site-profile` ont la priorité sur toutes les autres, sauf celles créées par un administrateur ou une application.

Ajouter et définir les propriétés des profils des services et des instances existant et spécifiez de nouvelles instances de service. Que les profils peuvent indiquer un manifeste quasiment comme vous pouvez spécifier.

Pour créer un profil de site, utilisez l'une des méthodes suivantes :

- Utilisez la commande `svcbundle` avec `bundle-type=profile` pour créer un nouveau fichier de profil.
- Utilisez la commande `svccfg extract` pour obtenir les informations de profil d'un service existant.

Personnaliser les valeurs de propriété dans le fichier des profils en question et inclure des commentaires sur le motif de chaque personnalisation. Copiez le fichier `/etc/svc/profile/site` et redémarrez le service `manifest-import`.

▼ Création d'un profil à l'aide de `svcbundle`

Le service `svc:/system/rmtmpfiles` est responsable du nettoyage du répertoire `/tmp` à l'initialisation. Par défaut, le service `rmtmpfiles` ne nettoie pas `/var/tmp`. Pour nettoyer `/var/tmp` durant l'initialisation, modifiez le comportement du service `svc:/system/rmtmpfiles` en définissant la propriété `options/clean_vartmp` sur `true`. Le moyen le plus simple pour obtenir ce comportement sur plusieurs systèmes consiste à créer un profil et le placer dans `/etc/svc/profile/site` sur chaque système.

1. Créez le profil.

La commande suivante permet de créer un profil dans `/tmp/rmtmpfiles.xml`.

```
$ svcbundle -o /tmp/rmtmpfiles.xml -s service-name=system/rmtmpfiles \
-s bundle-type=profile -s service-property=options:clean_vartmp:boolean:true
```

2. Apportez les modifications nécessaires au profil.

3. Copiez le profil dans le répertoire approprié.

```
$ cp /tmp/rmtmpfiles.xml /etc/svc/profile/site/rmtmpfiles.xml
```

4. Redémarrez le service d'importation de manifeste pour appliquer le profil au système.

```
$ svcadm restart manifest-import
```

Exemple 4-16 Installation automatique d'un profil à l'aide de `svcbundle`

Si vous n'avez pas besoin d'apporter des modifications au nouveau profil, vous pouvez utiliser l'option `-i` pour l'installer dès sa création. La commande `svcbundle` écrit le profil dans `/etc/svc/profile/site/rmtmpfiles.xml` et redémarre le service `manifest-import`. Tout fichier existant ayant le même nom dans le répertoire `/etc/svc/profile/site` sera remplacé.

```
# svcbundle -i -s service-name=system/rmtmpfiles \  
-s bundle-type=profile -s service-property=options:clean_vartmp:boolean:true
```

▼ Création d'un profil à l'aide de `svccfg`

1. Créer un profil.

La commande `svccfg extract` affiche un profil de service pour le service ou l'instance de service spécifié pour les couches `admin` et `site-profile`. Pour extraire des valeurs d'autres couches, utilisez l'option `-l`.

```
$ svccfg extract -l current network/dns/client > dnsclientprofile.xml
```

2. Apportez les modifications nécessaires au profil.

Modifiez le nom du profil à un nom explicite. Par défaut, le nom est défini sur `extract`, comme illustré dans l'exemple suivant :

```
<service_bundle type='profile' name='extract'>
```

3. Copiez le profil dans le répertoire approprié.

```
$ cp dnsclientprofile.xml /etc/svc/profile/site/dnsclientprofile.xml
```

4. Redémarrez le service d'importation de manifeste pour appliquer le profil au système.

```
$ svcadm restart manifest-import
```

Modification des services contrôlés par `inetd`

Un service contrôlé par `inetd` est un service SMF converti à partir d'une configuration dans le fichier `inetd.conf`. La commande `inetd` est l'agent de redémarrage délégué pour ces services.

La procédure suivante explique comment modifier les valeurs de propriété des services contrôlés par `inetd`.

Pour vérifier si le service que vous souhaitez modifier est contrôlé par `inetd`, appelez la commande `inetadm` sans options ni arguments pour répertorier tous les services contrôlés par `inetd`. L'exemple suivant présente uniquement une liste partielle.

```
$ inetadm
ENABLED STATE FMRI
enabled online svc:/application/cups/in-lpd:default
...
disabled disabled svc:/application/x11/xfs:default
```

L'option `-l` de la commande `inetadm` répertorie toutes les propriétés du service contrôlé par `inetd`. Dans l'exemple suivant, le message d'erreur indique que le service spécifié n'est pas un service contrôlé par `inetd`. "No restarter property" signifie que l'agent de redémarrage principal, `svc.startd`, gère l'instance de service.

```
$ inetadm -l ssh
Error: Specified service instance "svc:/network/ssh:default" has no
restarter property. inetd is not the delegated restarter of this instance.
```

De même, dans l'exemple suivant, le message "Couldn't find property 'general/restarter'" indique que l'agent de redémarrage par défaut, `svc.startd`, gère l'instance de service.

```
$ svccprop -p general/restarter ssh
svccprop: Couldn't find property 'general/restarter' for instance
'svc:/network/ssh:default'.
```

Si un service est contrôlé par `inetd`, `inetd` est son agent de redémarrage, comme illustré à l'exemple suivant.

```
$ svccprop -p general/restarter xfs
svc:/network/inetd:default
```

▼ Modification d'une valeur de propriété pour un service contrôlé par `inetd`

1. Dressez la liste des propriétés pour le service.

Utilisez l'option `-l` de la commande `inetadm` pour répertorier toutes les propriétés du service spécifié. Inspectez les valeurs actuelles des propriétés.

```
# inetadm -l FMRI
```

2. Modifier une valeur de la propriété.

Utilisez l'option `-m` de la commande `inetadm` pour modifier la valeur d'une propriété spécifiée. Des informations spécifiques sur les propriétés d'un service sont disponibles dans la page de manuel associée au service.

```
# inetadm -m FMRI property-name=value
```

Pour supprimer une valeur de la propriété, que vous spécifiez une valeur vide.

```
$ inetadm -m svc property=""
```

3. Vérifiez que la valeur de la propriété est modifiée.

Dressez de nouveau la liste des propriétés pour vous assurer que les modifications ont eu lieu.

```
# inetadm -l FMRI
```

4. Assurez-vous que la modification a pris effet.

Confirmez que a prévu que la modification de la propriété effet.

Exemple 4-17 Modification de la commande à exécuter au démarrage d'un service contrôlé par inetd

Cet exemple montre comment ajouter ou supprimer une option ou un argument à la ligne de commande d'un service contrôlé par inetd. La commande qui s'exécute lorsque le service démarre est la valeur de la propriété exec.

Utilisez l'option -l de la commande inetadm pour répertorier toutes les propriétés du service spécifié de manière à pouvoir inspecter la valeur actuelle de la propriété exec. Cet exemple montre le service svc:/application/x11/xfs, c'est-à-dire le serveur de police du système X Window. Voir la page de manuel xfs(1) pour plus d'informations.

```
$ inetadm -l xfs | grep exec
exec="/usr/bin/xfs -inetd"
```

Utilisez l'option -m de la commande inetadm pour modifier la valeur de la propriété exec du service spécifié.

```
$ inetadm -m xfs exec="/usr/bin/xfs -inetd -config /opt/site/fs/config"
```

Vérifiez que la valeur de la propriété est modifiée.

```
$ inetadm -l xfs | grep exec
exec="/usr/bin/xfs -inetd -config /opt/site/fs/config"
```

Modification des services configurés par un fichier

Un petit nombre de services SMF qui ne sont pas gérés par inetd reçoivent une partie de leur configuration à partir d'un fichier plutôt que des propriétés de services. Pour modifier ce fichier de configuration de procéder à la configuration, modifiez et utilisez les commandes pour redémarrer le service SMF. Ces fichiers de configuration peuvent être modifiés pendant que le service est en cours d'exécution, mais le contenu des fichiers n'est lu au démarrage du service.

Fichier de configuration avant de le modifier directement une, vérifiez les conditions suivantes :

- Assurez - vous que le fichier de configuration ne contient pas un message vous informant de ne pas directement le modifier.
- Assurez-vous que le service n'a pas de groupe de propriétés du type configfile.

```
$ svcprop -g configfile network/ntp
```

Si le service a un groupe de propriétés du type configfile, modifiez les propriétés dans ces groupes et non dans le fichier de configuration. Reportez-vous à [“Création d'un fichier de configuration à l'aide d'un pochoir”](#) à la page 98.

Par exemple, pour ajouter un serveur NTP pour vos clients NTP, ajoutez une entrée pour le serveur au fichier `/etc/inet/ntp.conf`, puis redémarrez le service NTP comme illustré dans la commande suivante :

```
$ svcadm restart svc:/network/ntp:default
```

Pour activer IKEv2, modifiez le fichier `/etc/inet/ike/ikev2.config` pour configurer le démon IKEv2, puis activez le service IKEv2 comme illustré dans la commande suivante. Pour modifier le fichier `ikev2.config`, utilisez la commande `pfedit` en tant qu'utilisateur ayant le profil Network IPsec Management (gestion IPsec du réseau). Modifier le fichier en le fichier correct de cette façon conserve la propriété. Voir la page de manuel [pfedit\(1M\)](#) pour des informations sur l'utilisation de `pfedit`.

```
$ svcadm enable svc:/network/ipsec/ike:ikev2
```


Utilisation de SMF pour contrôler votre application

Ce chapitre explique comment fournir des services pour votre application, y compris la manière d'utiliser un pochoir pour créer un fichier de configuration si votre application ne peut pas utiliser les interfaces de bibliothèques `libscf` pour lire les propriétés.

Pour d'autres exemples de la création et la diffusion de services pour effectuer des tâches telles que la configuration d'applications, reportez-vous au [Chapitre 7, “Automating System Change as Part of Package Installation”](#) du manuel “Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2” et au [Chapitre 8, “Advanced Topics For Package Updating”](#) du manuel “Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2”.

Création d'un service SMF

Un manifeste de service contient l'ensemble des propriétés associées à un service particulier, y compris les instances, les dépendances, les scripts à exécuter lorsque le service démarre et s'arrête et les valeurs par défaut des propriétés d'application. Les informations sur le modèle manifestes tels que fournissent également une description du service. Reportez-vous à la page de manuel [service_bundle\(4\)](#) et à la DTD d'ensemble de service `/usr/share/lib/xml/dtd/service_bundle.dtd.1` pour une description complète du contenu et du format d'un manifeste SMF. Reportez-vous aussi à [“Services de noms, instances, groupes de propriétés et propriétés” à la page 96](#) pour les règles de nommage et l'assignation de types de groupes de propriétés.

L'utilitaire de génération d'ensembles de services `svcbundle` est un bon outil pour créer un service simple ou démarrer un service complexe. Pour plus d'informations, reportez-vous à la page de manuel [svcbundle\(1M\)](#). Ensemble de services. vous pouvez utiliser et d'autres les manifestes de service DTD une page plus complexe de contrat pour terminer.

L'emplacement standard des manifestes personnalisés est `/lib/svc/manifest/site`. Les manifestes stockés à cet emplacement sont importés dans le référentiel de configuration de service par le service `svc:/system/early-manifest-import:default` pendant le processus d'initialisation avant le lancement des services. L'exécution précoce du processus d'importation

garantit que le référentiel contient les informations issues des derniers manifestes avant que les services ne soient démarrés. Les manifestes stockés à cet emplacement sont également importés lorsque le service `svc:/system/manifest-import` redémarre.

Plusieurs manifestes peuvent être utilisés pour décrire un service unique. Cela peut être utile, par exemple, pour définir une nouvelle instance d'un service sans modifier le manifeste existant du service. Cependant, si la même propriété dans la même couche du même service ou de la même instance est définie par plusieurs manifestes, SMF ne peut pas déterminer la valeur à utiliser. Lorsque ce type de conflit est détecté, l'instance est placée en état de maintenance. Pour plus d'informations sur les couches, reportez-vous à [“Référentiel Couches” à la page 26](#).

Ajoutez des métadonnées de nom et de description aux manifestes pour que les utilisateurs puissent obtenir des informations sur le service par les commandes `svcs` et `svccfg describe`. Vous pouvez également ajouter des descriptions de valeurs de propriété. Reportez-vous aux éléments `value`, `values` et `template` dans la DTD.

Utilisez la commande `svccfg validate` pour valider le fichier de manifeste de service ou le FMRI d'instance de service. Avec vos fichiers de manifeste, de méthode et de profil dans leurs emplacements standards, redémarrez le service `manifest-import` pour installer et configurer vos instances de service. Utilisez la commande `svcs` pour vérifier le statut de vos instances de service.

Cette section décrit la façon dont les services SMF pour créer vos propres. Le préfixe `site` est réservé aux personnalisations propres au site. Un service nommé `svc:/site/service-name` n'entrera pas en conflit avec les services fournis dans une version d'Oracle Solaris.

▼ Procédure de création d'un service SMF à l'aide de l'outil de génération de bundles de services

Cette procédure montre comment créer un service personnalisé existant qui utilise un script en tant que la méthode `start`.

1. Déterminer le modèle.

Par défaut, `svcbundle` crée un service transient. Déterminer si le script de méthode de démarrage de ce service démarre un démon au long cours et si c'est un service contract. Pour plus d'informations sur les modèles de service, reportez-vous à la section [“Modèles de service” à la page 20](#) et la page de manuel [`svc.startd\(1M\)`](#).

2. Copiez le script dans l'emplacement standard.

Dans cet exemple, le script de méthode de démarrage pour ce service est appelé `ex_svc`. Copiez ce script vers `/lib/svc/method/ex_svc`.

3. Créez un manifeste.

Dans cet exemple, le nom du service est `site/ex_svc`. Il s'agit d'un service transitoire qui n'a pas besoin de méthode d'arrêt.

```
$ svcbundle -o /tmp/ex_svc.xml -s service-name=site/ex_svc \  
-s start-method=/lib/svc/method/ex_svc
```

4. Apportez les modifications nécessaires au manifeste.

Vérifiez que le contenu du manifeste `/tmp/ex_svc.xml` correspond à vos besoins. Ajouter des commentaires, le cas échéant.

5. Vérifiez si le manifeste est valide.

Utilisez la commande `svccfg validate` pour vous assurer que le manifeste de service est valide.

6. Copiez le manifeste dans le répertoire standard.

```
$ cp /tmp/ex_svc.xml /lib/svc/manifest/site/ex_svc.xml
```

7. Importez le manifeste et copiez le service.

```
$ svcadm restart manifest-import
```

Exemple 5-1 Installation automatique d'un manifeste généré

Si vous n'avez pas besoin de saisir des modifications à ce nouveau manifeste de service, vous pouvez utiliser l'option `-i` pour installer le manifeste dès sa création. La commande `svcbundle` écrit le manifeste vers `/lib/svc/manifest/site` et redémarre le service `manifest-import`. Tout fichier du même nom dans le répertoire `/lib/svc/manifest/site` sera écrasé.

```
# svcbundle -i -s service-name=site/ex_svc \  
-s start-method=/lib/svc/method/ex_svc
```

Création d'un service de démarrage ou d'arrêt d'une instance de base de données Oracle

Cet exemple présente les services suivants qui vous aider à gérer l'Oracle Database, procédez comme suit :

- Un service de base de données qui démarre ou arrête une instance Oracle Database
- Service qui démarre le listener, qui est un processus de qui gère les demandes de connexion client le trafic entrant à l'instance de base de données

Cet exemple utilise un stockage sauvegardé sur fichier. Lieu d'effectuer une mise à jour de stockage à l'aide d'un fichier de sauvegarde est d'utiliser l'ASM Automatic Storage Management

(gestion des services) d'. ASM est un gestionnaire de volumes et un système de fichiers pour les fichiers Oracle Database.

Les variables d'environnement ci-dessous doit être défini pour chaque installation de l'Oracle Database, procédez comme suit :

ORACLE_HOME	L'emplacement où est installée la base de données. Dans l'exemple présenté dans cette section, il se peut que l'emplacement d'installation de la base de données soit /opt/oracle/product/home.
ORACLE_SID	Les systèmes ID pour identifier de manière unique une base de données particulière sur un système.

Dans cet exemple, ces variables d'environnement sont définies dans les fichiers manifestes de service, puis utilisés dans la méthode des scripts.

Base de données de démarrage et d'arrêt de service

Cette section montre le manifeste de service de contrôle d'instance d'Oracle Database, /lib/svc/manifest/site/oracle.xml. Vous trouverez ci-dessous quelques fonctions manifeste à noter pour ce service, procédez comme suit :

- Une instance de service est définie, appelée `svc:/site/application/database/oracle:default`. Cette instance est activée par défaut.
Cet exemple montre deux façons de définir l'instance par défaut. Dans cet exemple, l'instance par défaut est définie dans l'élément `create_default_instance` en haut du manifeste. L'élément `instance` en bas du manifeste présente l'autre manière de faire.
- Ce service exige que tous les systèmes de fichiers à monter local et toutes les interfaces réseau à initialiser.
Si vous utilisez une base de données, le fichier de sauvegarde devrait dépendre de la base de données de service de fichiers local. Si vous utilisez ASM, le service de base de données doit dépendre d'ASM qui permet de gérer le service. Devrait dépendre de la base de données de service par le client distant de la mise en réseau à autoriser les connexions.
- L'élément `method_environment` dans l'élément `method_context` définit les variables d'environnement `ORACLE_HOME` et `ORACLE_SID`, qui identifient l'instance de base de données à démarrer ou arrêter. Ces valeurs sont alors disponibles pour le script de méthode à utiliser.
Si vous créez plusieurs instances de ce service (reportez-vous à l'élément `instance` en bas du manifeste), chaque instance pourrait avoir besoin d'un élément `method_context` propre pour définir les valeurs d'`ORACLE_HOME` et `ORACLE_SID` pour cette base de données en particulier.
- Les attributs de l'élément `method_context` peuvent définir une réserve de ressources en plus du répertoire de projet et du répertoire actif présentés dans cet exemple. Vous pouvez aussi

définir un élément `method_profile` ou `method_credential` dans l'élément `method_context` element. L'élément `method_credential` peut spécifier les valeurs `supp_groups` et `limit_privileges` en plus des valeurs `user`, `group` et `privileges` présentées dans cet exemple. Pour plus d'informations à ce sujet, reportez-vous à la DTD.

- Le script de démarrage/arrêt de méthode est `/lib/svc/method/oracle`. Le nombre de secondes écoulées avant l'augmentation de méthode qu'il y a temporisation de la valeur par défaut.
- L'utilisateur doit bénéficier de la permission `solaris.smf.manage.oracle` pour activer ou désactiver cette instance de service. Dans cet exemple, l'utilisateur `oracle` a la permission `solaris.smf.manage.oracle`.

```
<?xml version="1.0"?>

<!--
  Define a service to control the startup and shutdown of a database instance.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">

<service_bundle type="manifest" name="oracle">
<service name="site/application/database/oracle" type="service" version="1">
  <create_default_instance enabled="true" />

  <!--
    Wait for all local file systems to be mounted.
    Wait for all network interfaces to be initialized.
  -->

  <dependency type="service"
    name="fs-local"
    grouping="require_all"
    restart_on="none">
    <service_fmri value="svc:/system/filesystem/local" />
  </dependency>

  <dependency type="service"
    name="network"
    grouping="require_all"
    restart_on="none">
    <service_fmri value="svc:/milestone/network:default" />
  </dependency>

  <!-- Define the methods. -->

  <method_context project=":default" working_directory=":default">
    <method_credential user="oracle" group="dba" privileges=":default" />
    <method_environment>
      <envvar name="ORACLE_HOME" value="/opt/oracle/product/home" />
      <envvar name="ORACLE_SID" value="oracle" />
    </method_environment>
  </method_context>
```

```

<exec_method type="method"
  name="start"
  exec="/lib/svc/method/oracle start"
  timeout_seconds="120"/>

<exec_method type="method"
  name="stop"
  exec="/lib/svc/method/oracle stop"
  timeout_seconds="120" />

<!--
  What authorization is needed to allow the framework
  general/enabled property to be changed when performing the
  action (enable, disable, etc) on the service.
-->

<property_group name="general" type="framework">
  <propval type="astring"
    name="action_authorization"
    value="solaris.smf.manage.oracle" />
</property_group>

<!-- Define an instance of the database. -->

<!--<instance name="default" enabled="true" />-->

<stability value="Evolving" />
</service>
</service_bundle>

```

Ajoutez des métadonnées de nom et de description au manifeste pour que les utilisateurs puissent obtenir des informations sur le service par les commandes `svcs` et `svccfg describe`. Reportez-vous à l'élément template de la DTD.

Utilisez la commande `svccfg validate` pour vous assurer que le manifeste de service est valide.

Ceci est le script de méthode de démarrage/arret, `/lib/svc/method/oracle`, pour le service de contrôle d'instance d'Oracle Database. Cette méthode appelle les commandes de base de données `dbstart` et `dbshut`.

```

#!/bin/ksh -p

. /lib/svc/share/smf_include.sh

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
export PATH=$PATH:$ORACLE_HOME/bin

function startup
{
    dbstart $ORACLE_HOME
}

```

```
function shutdown
{
    dbshut $ORACLE_HOME
}

case $1 in
    start) startup ;;
    stop) shutdown ;;

    *) echo "Usage: $0 { start | stop }" >&2
       exit $SMF_EXIT_ERR_FATAL
       ;;
esac

exit $SMF_EXIT_OK
```

Service de module d'écoute de base de données

Listener est un processus permettant la le trafic entrant de client de gérer les demandes de connexion vers l'instance de base de données. Le listener dépend le service de l'instance de service de base de données qu'il gère dont les connexions client.

Cette section présente le manifeste de service du listener d'instance d'Oracle Database, `/lib/svc/manifest/site/listener.xml`. Vous trouverez ci-dessous quelques fonctions manifeste à noter pour ce service, procédez comme suit :

- Une instance de service est définie, appelée `svc:/site/application/database/listener:default`. Cette instance est activée par défaut.
- Pour ce service, il faut que le service de contrôle d'instance d'Oracle Database, `svc:/site/application/database/oracle`, soit démarré. Si la base de données redémarrage de l'instance de listener pour quelque raison que ce soit, le sera également redémarrée.
- L'élément `method_environment` dans l'élément `method_context` définit les variables d'environnement `ORACLE_HOME` et `ORACLE_SID`, qui identifient l'instance de base de données à démarrer ou arrêter. Ces valeurs sont alors disponibles pour le script de méthode à utiliser.

Si vous créez plusieurs instances de ce service (reportez-vous à l'élément `instance` en bas du manifeste), chaque instance pourrait avoir besoin d'un élément `method_context` propre pour définir les valeurs d'`ORACLE_HOME` et `ORACLE_SID` pour cette base de données en particulier.

- Les attributs de l'élément `method_context` peuvent définir une réserve de ressources en plus du répertoire de projet et du répertoire actif présentés dans cet exemple. Vous pouvez aussi définir un élément `method_profile` ou `method_credential` dans l'élément `method_context` élément. L'élément `method_credential` peut spécifier les valeurs `supp_groups` et `limit_privileges` en plus des valeurs `user`, `group` et `privileges` présentées dans cet exemple. Pour plus d'informations à ce sujet, reportez-vous à la DTD.

- Le script de méthode de démarrage/arrêt est /lib/svc/method/listener. Le nombre de secondes écoulées avant l'augmentation de méthode qu'il y a temporisation de la valeur par défaut.
- L'utilisateur doit bénéficier de la permission solaris.smf.manage.oracle pour activer ou désactiver cette instance de service.
- Le service est transient. Voir [“Modèles de service” à la page 20](#).

```
<?xml version="1.0"?>

<!--
  Define a service to control the startup and shutdown of a database listener.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">

<service_bundle type="manifest" name="listener">
  <service name="site/application/database/listener" type="service" version="1">
    <create_default_instance enabled="true" />
    <!--<single_instance /-->

    <!-- Wait for the database to be started. -->

    <dependency type="service"
      name="oracle"
      grouping="require_all"
      restart_on="refresh">
      <service_fmri value="svc:/site/application/database/oracle" />
    </dependency>

    <!-- Define the methods. -->

    <method_context project=":default" working_directory=":default">
      <method_credential user="oracle" group="dba" privileges=":default" />
      <method_environment>
        <envvar name="ORACLE_HOME" value="/opt/oracle/product/home" />
        <envvar name="ORACLE_SID" value="oracle" />
      </method_environment>
    </method_context>

    <exec_method type="method"
      name="start"
      exec="/lib/svc/method/listener start"
      timeout_seconds="150"/>

    <exec_method type="method"
      name="stop"
      exec="/lib/svc/method/listener stop"
      timeout_seconds="30" />

    <!--
      What authorization is needed to allow the framework
      general/enabled property to be changed when performing the
      action (enable, disable, etc) on the service.
```

```
-->

<property_group name="general" type="framework">
  <propval type="astring"
    name="action_authorization"
    value="solaris.smf.manage.oracle" />
</property_group>

<!-- Make the instance transient (since it backgrounds itself). -->

<property_group name="startd" type="framework">
  <propval name="duration" type="astring" value="transient" />
</property_group>

<!-- Define an instance of the listener. -->

<!--<instance name="default" enabled="true" />-->

  <stability value="Evolving" />
</service>
</service_bundle>
```

Ajoutez des métadonnées de nom et de description au manifeste pour que les utilisateurs puissent obtenir des informations sur le service par les commandes `svcs` et `svccfg describe`. Reportez-vous à l'élément `template` de la DTD.

Voici le script de méthode de démarrage/arrêt, `/lib/svc/method/listener`, pour le service `listener` d'instance d'Oracle Database. Cette méthode démarre ou arrête un processus `listener`, `lsnrctl`. Au démarrage de `lsnrctl`, il contrôle l'état du service de base de données.

```
#!/bin/ksh -p

. /lib/svc/share/smf_include.sh

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
export PATH=$PATH:$ORACLE_HOME/bin

function startup
{
    lsnrctl start

    # Wait for the listener to report ready.

    i=0
    while ! lsnrctl status | grep -i ready ; do
        ((i = i+1))
        if (( $i == 120 )) ; then
            # It's been *at least* 2 minutes, time to give up.
            echo "The listener failed to report ready." >&2
            exit $SMF_EXIT_ERR_FATAL
        fi

        sleep 1
    done
}
```

```

done

# Ping the database once to prove it is now available.

if ! tnsping $ORACLE_SID ; then
    exit $SMF_EXIT_ERR_FATAL
fi
}

function shutdown
{
    lsnrctl stop
}

case $1 in
    start) startup ;;
    stop) shutdown ;;

    *) echo "Usage: $0 { start | stop }" >&2
       exit $SMF_EXIT_ERR_FATAL
       ;;
esac

exit $SMF_EXIT_OK

```

Services de noms, instances, groupes de propriétés et propriétés

Les noms d'instance de service doit être contenu et l'expression suivante :

```
[A-Za-z0-9][_A-Za-z0-9.-]*,)?[A-Za-z0-9][_A-Za-z0-9.-]*
```

Les noms de service ou d'instance sont sensibles à la casse et doivent commencer par un caractère alphanumérique ; ils peuvent contenir des caractères alphanumériques et les caractères suivants : tiret bas (_), tiret (-), point (.). Pour des raisons de compatibilité en amont, une seule virgule (,) est autorisée entre le premier et le dernier caractère.

Groupe de propriétés et les noms de propriété doit tenir dans l'expression suivante :

```
[A-Za-z0-9-._~:/?#\[\]@!$&'(\)*+,;= %]+
```

Les noms des propriétés ou des groupes de propriétés sont sensibles à la casse ; ils peuvent contenir des caractères alphanumériques et les caractères suivants : tiret (-), point (.), tiret bas (_), tilde (~), deux points (:), barre oblique (/), point d'interrogation (?), dièse (#), parenthèses carrées ([et]), arobase (@), point d'exclamation (!), signe dollar (\$), esperluette (&), guillemet simple ('), parenthèses ((et)), astérisque (*), signe plus (+), virgule (,), point-virgule (;), signe égal (=), espace et signe pourcent (%).

Dans un FMRI, les noms de propriétés et de groupes de propriétés sont encodés conformément à [Uniform Resource Identifier \(URI\) Generic Syntax RFC 3986](#), à l'exception de la virgule, qui n'est pas encodée.

Un type de groupe de propriétés est une catégorie de groupe de propriétés. Parmi les types de groupe de propriétés, on trouve `application`, `dependency`, `method`, `framework`, `implementation` et `template`. Vous pouvez ajouter d'autres types, pourvu qu'ils soient conformes à la convention de nommage étendue dans `smf(5)`. Ne spécifiez pas `framework`, `implementation` ou `template` comme type de votre groupe de propriétés. Les groupes de propriétés des types `framework`, `implementation` et `template` ont des utilisations réservées dans SMF. Les groupes de propriétés du type `application` ne devraient présenter un intérêt que pour le service auquel ce groupe est rattaché.

▼ Procédure de conversion d'un script de contrôle d'exécution en service SMF

Cette procédure montre comment remplacer un script de contrôle d'exécution par un manifeste de service SMF de contrôle d'exécution de manière à ce que le service puisse être géré par SMF. Pour convertir un script de contrôle d'exécution, utilisez le nom `rc-script` avec l'option `-s` de la commande `svcbundle`. Pour plus d'informations, reportez-vous à la page de manuel [svcbundle\(1M\)](#) ou saisissez `svcbundle help rc-script`.

1. Déterminer le modèle.

Par défaut, `svcbundle` crée un service `transient`. Déterminez si ce script de contrôle d'exécution démarre un démon au long cours et s'il s'agit d'un service `contract`. Pour plus d'informations sur les modèles de service, reportez-vous à la section “[Modèles de service](#)” à la page 20 et la page de manuel [svc.startd\(1M\)](#).

2. Créez un manifeste.

Dans cet exemple, le service s'appelle `ex_con` ; il s'agit d'un service de contrat s'exécutant au niveau 2.

```
$ svcbundle -o /tmp/ex_con.xml -s service-name=ex_con  
-s rc-script=/etc/init.d/ex_con:2 -s model=contract
```

3. Apportez les modifications nécessaires au manifeste.

Vérifiez que le contenu du manifeste `/tmp/ex_con.xml` correspond à vos besoins. Ajouter des commentaires, le cas échéant.

4. Copiez le manifeste dans le répertoire standard.

```
$ cp /tmp/ex_con.xml /lib/svc/manifest/site/ex_con.xml
```

5. Arrêtez le service existant.

```
$ /etc/init.d/ex_con stop
```

6. Désactivez le script de contrôle d'exécution.

Supprimez tous les liens vers le script de contrôle d'exécution depuis les répertoires *rcn.d* appropriés.

7. Importez le manifeste et copiez le service.

```
$ svcadm restart manifest-import
```

8. Liste du nouveau service.

Vérifiez que le nouveau service existe et qu'il est dans l'état attendu.

```
$ svcs ex-con
```

Création d'un fichier de configuration à l'aide d'un pochoir

Si l'application ne peut pas utiliser les interfaces de bibliothèques *libscf* pour lire les propriétés, vous pouvez utiliser un pochoir pour créer un fichier de configuration. Un *service pochoir* crée des fichiers de configuration à l'aide d'un fichier pochoir et de valeurs de propriétés définies dans le service pochoir. Un *fichier pochoir* contient une définition structurelle d'un fichier de configuration nécessaire à un service, même si ce service est désormais géré par SMF. Stencil services vous permettent de SMF de gestion de configuration tirer parti des mais n'entraîne aucune modification de l'application existante.

Le pochoir est utilisé pour générer un fichier de configuration immédiatement avant l'exécution de la méthode d'instance de service *start*. Si vous mettez à jour les valeurs de propriété *stenciled*, redémarrez le service pour incorporer les changements dans le fichier de configuration avant l'application démarre et lit le fichier de configuration.

▼ Création d'un service de pochoir

Structurelle stencil fichier contient une définition d'un fichier de configuration qui continue à être nécessaires à un service-ce qui n'est maintenant gérée par SMF de service. L'utilitaire *svcio* génère le fichier de configuration à partir des définitions du fichier pochoir et des propriétés dans le service SMF. Reportez-vous à la page de manuel *svcio*(1) pour plus d'informations à propos de l'utilitaire *svcio* et à la page de manuel *smf_stencil*(4) pour plus d'informations à propos du format des fichiers pochoirs.

1. Créer un stencil fichier.

Le fichier pochoir indique à l'utilitaire `svcio` le format à utiliser pour la création du fichier de configuration. L'utilitaire `svcio` convertit les propriétés de SMF en fichiers de configuration spécifiques à l'application en partant d'un modèle appelé "pochoir".

2. Ajouter un groupe de propriétés au service.

Le groupe de propriétés du service pochoir indique à l'utilitaire `svcio` le chemin et le propriétaire à utiliser pour la création du fichier de configuration. Stencil SMF-permet de régénérer la configuration de tous les services prenant en charge les avant la date de début ou de méthodes de régénération en cours d'exécution. Les groupes de propriétés du type `configfile` indiquent à SMF comment générer les fichiers de configuration. Chaque groupe de propriétés `configfile` décrit un fichier de configuration pour le service et indique à `svcio` comment générer ces fichiers à partir d'autres propriétés enregistrées dans le référentiel SMF.

Pour configurer ce dernier soit stencil propriété prenant en charge les étiquettes, ajoutez un fichier de configuration pour chaque gérée par un groupe qui contient les chemins de fichier de façon à utiliser à la fois le stencil et le résultat sous la forme d'un fichier de configuration modèle. Le groupe de propriétés présente les propriétés suivantes :

<code>path</code>	Le chemin où enregistrer le fichier de configuration, par exemple <code>/etc/svc.conf</code> .
<code>stencil</code>	Le chemin du fichier pochoir à utiliser, par rapport à <code>/lib/svc/stencils</code> . Par exemple, si la valeur de la propriété <code>stencil</code> et <code>svc.stencil</code> , le fichier <code>/lib/svc/stencils/svc.stencil</code> est utilisé.
<code>mode</code>	Le mode à utiliser pour le fichier de configuration (<code>path</code>), par exemple <code>644</code> .
<code>owner</code>	Le propriétaire à définir pour le fichier de configuration (<code>path</code>). Si cette propriété n'est pas définie, le propriétaire du fichier est celui qui appelle <code>svcio</code> .
<code>group</code>	Le groupe à définir pour le fichier de configuration (<code>path</code>). Si cette propriété n'est pas définie, le groupe est le groupe par défaut de <code>path</code> .

Exemple de service de pochoir dans Oracle Solaris

Kerberos Services pour utiliser Spectacle stencils et les fichiers de configuration doivent être appliquées.

Spectacle stencil de service

Pour la gestion est une boîte à outils Spectacle de nombreux systèmes. Spectacle sur, l'application Oracle Solaris est géré par SMF.

La vue des Spectacle de haut niveau Services

Lorsque vous installez le package `system/management/puppet`, vous obtenez deux instances de service SMF : `puppet:master` et `puppet:agent`. Ces instances sont désactivés par défaut.

Une fois ces instances activées, la commande suivante montre que `puppet:master` et `puppet:agent` sont les deux des services contrats :

```
$ svcs -p puppet
STATE      STIME      FMRI
online     17:19:32   svc:/application/puppet:agent
           17:19:32   2565 puppet
online     17:19:32   svc:/application/puppet:master
           17:19:32   2567 puppet
```

La commande suivante indique un peu plus d'informations sur les services de sous-traitance processus lancés par le :

```
$ ps -o pid,args -p 2565,2567
PID COMMAND
2565 /usr/ruby/1.9/bin/ruby /usr/sbin/puppet agent --logdest /var/log/puppet/puppet-
2567 /usr/ruby/1.9/bin/ruby /usr/sbin/puppet master --logdest /var/log/puppet/puppet
```

Comme la sortie `ps` l'indique, `puppet` écrit vers des fichiers journaux dans `/var/log/puppet:`

```
$ ls /var/log/puppet
puppet-agent.log  puppet-master.log
```

Spectacle Fichier de configuration initiale

Puppet s'attend à utiliser un fichier de configuration appelé `/etc/puppet/puppet.conf`. L'application `/usr/sbin/puppet` lit les informations de configuration depuis `/etc/puppet/puppet.conf`, et non depuis les propriétés configurées dans les instances de service `application/puppet`. Afin de fournir le fichier de configuration requis, chaque instance de `puppet` fournit un fichier pochoir et un groupe de propriétés `configfile`. Le groupe de propriétés `configfile` ordonne à l'utilitaire `svcio` de s'exécuter et de créer le fichier de configuration spécifié. Le fichier est utilisé pour l'écriture stencil de données à partir du service au fichier de configuration des valeurs de propriété dans le format correct.

La commande suivante montre toutes les propriétés du service `puppet` faisant partie d'un groupe de propriétés du type `configfile`. Cette sortie indique que les deux instances du service `puppet` ont les mêmes propriétés de `configfile` avec les mêmes valeurs. Chaque instance du service

puppet fournit le chemin du fichier de configuration, le mode du fichier de configuration et le chemin du fichier pochoir.

```
$ svcprop -g configfile puppet
svc:/application/puppet:master/:properties/puppet_stencil/mode astring 0444
svc:/application/puppet:master/:properties/puppet_stencil/path astring /etc/puppet/puppet.conf
svc:/application/puppet:master/:properties/puppet_stencil/stencil astring puppet.stencil
svc:/application/puppet:agent/:properties/puppet_stencil/mode astring 0444
svc:/application/puppet:agent/:properties/puppet_stencil/path astring /etc/puppet/puppet.conf
svc:/application/puppet:agent/:properties/puppet_stencil/stencil astring puppet.stencil
```

Les commandes suivantes confirmer que ces propriétés d'instance sont héritées du service parent.

```
$ svccfg -s puppet listprop -l all puppet_stencil
puppet_stencil      configfile manifest
puppet_stencil/mode astring      manifest      0444
puppet_stencil/path astring      manifest      /etc/puppet/puppet.conf
puppet_stencil/stencil astring    manifest      puppet.stencil
$ svccfg -s puppet:agent listprop -l all puppet_stencil
$ svccfg -s puppet:master listprop -l all puppet_stencil
```

En fonction de votre infrastructure, vous pourriez avoir besoin, par exemple, des instances puppet:agent1 et puppet:agent2. Dans ce cas, vous personnalisez les valeurs de propriétés et ajoutez des propriétés pour chaque instance, comme indiqué dans [“La modification du Spectacle Fichier de configuration” à la page 102](#).

Voici le contenu initial du fichier de configuration, /etc/puppet/puppet.conf:

```
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE. EDITS WILL BE LOST.
#
# See puppet.conf(5) and http://docs.puppetlabs.com/guides/configuring.html
# for details.
```

Fichier Spectacle stencil

Le contenu du fichier indique quelle stencil les propriétés et d'autres informations sont écrites dans le fichier de configuration. Le chemin de puppet.stencil qui est la valeur de la propriété puppet_stencil/stencil est par rapport /lib/svc/stencils. Voici le contenu du fichier pochoir, /lib/svc/stencils/puppet.stencil:

```
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE. EDITS WILL BE LOST.
#
# See puppet.conf(5) and http://docs.puppetlabs.com/guides/configuring.html
# for details.
; walk each instance and extract all properties from the config PG
$%(svc:/%s:(.*)/properties)/ {
  $%{$%1/general/enabled:?
  [%%2]
  $%/%%1/config/(.*)/ {
```

```
%3 = ${%1/config/%3} }  
}  
}
```

Dans le fichier pochoir, `svc:/%s:(.*)/:properties` (ou %1) s'agrandit pour donner `svc:/application/puppet:agent/:properties` et `svc:/application/puppet:master/:properties`, où `.*` (ou %2) trouve toutes les instances. Le nom de l'instance est ensuite utilisé pour étiqueter ce bloc dans le fichier de configuration. L'occurrence suivante de `.*` (ou %3) trouve toutes les propriétés du groupe de propriétés `config` pour l'instance de service %1. Le pochoir ordonne à `svcio` d'écrire le nom et la valeur de propriété issus de l'instance de service vers le fichier de configuration.

La modification du Spectacle Fichier de configuration

Comme vous pouvez le constater dans [“Spectacle Fichier de configuration initiale” à la page 100](#), seules les lignes de commentaires littérales sont écrites vers le fichier de configuration au début. L'écriture de valeurs de propriétés vers le fichier de configuration est bloquée par le test de la valeur de la propriété `general/enabled` dans le fichier pochoir. La commande suivante indique que par défaut, la valeur de la propriété `general/enabled` est `false` :

```
$ svcprop -p general/enabled puppet  
svc:/application/puppet:master/:properties/general/enabled boolean false  
svc:/application/puppet:agent/:properties/general/enabled boolean false
```

L'utilisation de la commande `svcadm enable` pour activer une instance ne modifie pas la valeur de la propriété `general/enabled`. Lorsque vous changez la valeur de la propriété `general/enabled` à `true` et redémarrez l'instance, toutes les propriétés du groupe de propriétés `config` de cette instance sont écrites vers le fichier de configuration.

```
$ svccfg -s puppet:agent setprop general/enabled=true  
$ svcprop -p general/enabled puppet:agent  
false  
$ svcadm refresh puppet:agent  
$ svcprop -p general/enabled puppet:agent  
true  
$ svcadm restart puppet:agent
```

La commande suivante montre qu'initialement, la seule propriété du groupe de propriétés `config` est le chemin du fichier journal de chaque instance :

```
$ svcprop -p config puppet  
svc:/application/puppet:master/:properties/config/logdest astring /var/log/puppet/puppet-  
master.log  
svc:/application/puppet:agent/:properties/config/logdest astring /var/log/puppet/puppet-  
agent.log
```

La propriété `config` de l'instance activée est ajoutée au fichier de configuration dans un bloc identifié par le nom d'instance :

```
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE.  EDITS WILL BE LOST.
#
# See puppet.conf(5) and http://docs.puppetlabs.com/guides/configuring.html
# for details.
```

```
[agent]
```

```
logdest = /var/log/puppet/puppet-agent.log
```

La documentation de configuration de Puppet affirme que le fichier de configuration peut comporter des blocs [main], [agent] et [master]. La configuration dans le bloc [main] s'applique à l'agent ainsi qu'au master. Pour l'agent Puppet, la configuration du bloc [agent] prime sur la même configuration dans le bloc [main]. Pour le master Puppet, la configuration du bloc [master] prime la même configuration dans le bloc [main]. Si vous souhaitez fournir un bloc de configuration [main] qui soit commun à l'agent et au master, créez une instance puppet:main et des propriétés config appropriées pour cette instance.

Les commandes suivantes indiquent comment Spectacle pour ajouter à votre fichier de configuration de configuration.

```
$ svccfg -s puppet:agent
svc:/application/puppet:agent> setprop config/report=true
svc:/application/puppet:agent> setprop config/pluginsync=true
svc:/application/puppet:agent> refresh
svc:/application/puppet:agent> exit
$ svcadm restart puppet:agent
$ cat /etc/puppet/puppet.conf
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE.  EDITS WILL BE LOST.
#
# See puppet.conf(5) and http://docs.puppetlabs.com/guides/configuring.html
# for details.
```

```
[agent]
```

```
logdest = /var/log/puppet/puppet-agent.log
pluginsync = true
report = true
```

Des commandes similaires peuvent être utilisées pour enlever des propriétés et modifier les valeurs de propriété. Reportez-vous au [Chapitre 4, Configuration des services](#). Pour ajouter une instance main, utilisez la commande svccfg add comme indiqué dans [“Ajout d'instances de service” à la page 79](#).

Service Kerberos stencil

Oracle Solaris un autre exemple d'utilisation de la stencil est service qui utilise un Kerberos. La commande suivante indique que le groupe de propriétés configfile est krb5_conf, le fichier

pochoir /lib/svc/stencils/krb5.conf.stencil et le fichier de configuration /etc/krb5/krb5.conf.

```
$ svcprop -g configfile svc:/system/kerberos/install:default
krb5_conf/disabled boolean true
krb5_conf/group astring sys
krb5_conf/mode integer 644
krb5_conf/owner astring root
krb5_conf/path astring /etc/krb5/krb5.conf
krb5_conf/stencil astring krb5.conf.stencil
```



Pratiques recommandées dans le cadre de SMF et du dépistage des pannes

Cette annexe fournit certaines des meilleures pratiques recommandées et montre comment résoudre certains problèmes liés au service SMF.

Pratiques recommandées dans le cadre de SMF

La plupart des services décrivent la stratégie de configuration. Si la configuration que vous recherchez n'est pas mis en oeuvre, modifiez la description de la règle en modifiant le service. Modifiez les valeurs des propriétés de service ou créer de nouvelles instances de service avec des valeurs de propriété différentes. Ne désactivez pas les instances de service et ne modifiez pas les fichiers de configuration qui sont prévus pour être gérés par un service SMF.

Ne modifiez pas les profils système et manifestes fournis par Oracle ou des fournisseurs de logiciels tiers. Ces manifestes et les profils sont susceptibles d'être remplacés lorsque vous effectuez une mise à niveau de votre système. Les modifications apportées à ces fichiers seront perdues. Créez plutôt un profil de site pour personnaliser le service ou utilisez la commande `svccfg` ou la commande `inetadm` pour manipuler directement les propriétés. Les répertoires `/lib/svc/manifest/site` et `/var/svc/manifest/site` sont également réservés à un usage propre au site. Oracle Solaris ne fournit pas de manifestes dans ces répertoires.

Pour appliquer la même configuration personnalisée sur plusieurs systèmes, utilisez la commande `svcbundle` ou la commande `svccfg extract` pour créer un fichier de profil. Personnalisez les valeurs de propriété dans ce fichier, les valeurs et incluez des commentaires sur le motif de chaque personnalisation. Copiez le fichier dans `/etc/svc/profile/sites` sur chaque système, et redémarrez le service `manifest-import` sur chacun des systèmes. Reportez-vous à la section [“Configuration de plusieurs systèmes” à la page 80](#).

Lorsque vous créez un profil de site, assurez-vous que la configuration définie n'est pas en conflit avec la configuration définie dans un autre profil de site pour le même service ou la même instance de service. Lorsque SMF trouve une configuration en conflit sur la même couche du référentiel de configuration de service, l'instance de service affectée est placée en état de maintenance.

N'utilisez pas les emplacements non standard pour le manifeste et les fichiers de profils. Reportez-vous à la section [“Les lots d'offres de service” à la page 25](#) pour connaître les emplacements standard des manifestes et des profils.

Lors de la création d'un service pour votre usage personnel, utilisez `site` au début du nom de service : `svc:/site/service_name:instance_name`.

Ne modifiez pas la configuration du service de redémarrage principal, `svc:/system/svc/restart:default`, sauf pour configurer les niveaux de journalisation, comme décrit dans [“Spécification de la quantité de messages au démarrage” à la page 114](#).

Avant d'utiliser la commande `svccfg delcust`, utilisez la commande `svccfg listcust` avec les mêmes options. La sous-commande `delcust` peut supprimer toutes les personnalisations administratives d'un service. Utilisez la sous-commande `listcust` pour vérifier les personnalisations qui seront supprimées par la sous-commande `delcust`.

Dans les scripts, utilisez le FMRI complet de l'instance de service :
`svc:/service_name:instance_name`.

Dépistage des pannes de services

Cette section traite des sujets suivants :

- Validation des modifications de configuration dans l'instantané en cours d'exécution
- Dépannage des services comportant des problèmes
- Transition manuelle d'une instance à l'état `degraded` ou `maintenance`
- Réparation d'un référentiel de configuration de service corrompu
- Configuration de la quantité de messages à afficher ou à stocker au démarrage du système
- Transition vers ou initialisation à partir d'un jalon spécifique
- Utilisation de SMF pour rechercher des problèmes d'initialisation
- Conversion des services `inetd` en services SMF

Compréhension des modifications de la configuration

Dans le référentiel de configuration de service, SMF stocke les modifications apportées aux propriétés séparément aux propriétés de l'instantané en cours d'exécution. Lorsque vous modifiez la configuration de service, les modifications n'apparaissent pas immédiatement dans l'instantané en cours d'exécution.

L'opération d'actualisation met à jour l'instantané en cours d'exécution de l'instance de service spécifiée avec les valeurs de la configuration de modification.

Par défaut, la commande `svccprop` affiche les propriétés dans l'instantané en cours d'exécution ; la commande `svccfg` affiche les propriétés dans la configuration d'édition. Si vous avez modifié les valeurs de propriété mais n'avez pas actualisé la configuration, les commandes `svccprop` et `svccfg` n'affichent pas les mêmes valeurs de propriétés. Après avoir actualisé la configuration, les commandes `svccprop` et `svccfg` affichent les mêmes valeurs de propriété.

La réinitialisation ne modifie pas l'instantané en cours d'exécution. La commande `svcadm restart` n'actualise pas la configuration. Utilisez la commande `svcadm refresh` ou `svccfg refresh` pour enregistrer les modifications de configuration dans l'instantané en cours d'exécution.

Réparation d'une instance endommagée, hors ligne ou en cours de maintenance

Utilisez la commande `svcs -x` sans arguments pour afficher des explications sur toutes les instances de service correspondant à une des descriptions suivantes :

- Le service est activé mais n'est pas en cours d'exécution.
- Le service empêche un autre service activé de s'exécuter.

La liste ci-dessous présente brièvement comment traiter les problèmes de service :

1. Diagnostiquez le problème, en affichant d'abord le fichier journal du service.
2. Résolez le problème. Si la correction du problème implique la modification de la configuration du service, actualisez le service.
3. Faites passer les services concernés à l'état d'exécution.

▼ Procédure de réparation d'une instance en maintenance

Une instance de service est en maintenance est activée mais ne parvient pas à s'exécuter.

1. Déterminez la raison pour laquelle l'instance est en maintenance.

L'instance peut passer par l'état maintenance parce qu'une action administrative n'est pas encore terminée. Si l'instance est en transition, son état devrait s'afficher sous la forme `maintenance*`.

Dans l'exemple suivant, les lignes "State" et "Reason" montrent que le service `pkg/depot` se trouve dans l'état maintenance parce que sa méthode de démarrage a échoué.

```
$ svcs -x
svc:/application/pkg/depot:default (IPS Depot)
  State: maintenance since September 11, 2013 01:30:42 PM PDT
Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
```

```
See: http://support.oracle.com/msg/SMF-8000-KS
See: pkg.depot-config(1M)
See: /var/svc/log/application-pkg-depot:default.log
Impact: This service is not running.
```

Connectez-vous au site de support Oracle pour visualiser l'article de connaissances sur l'autorétablissement prédictif dont il est fait référence. Dans ce cas, l'article vous indique que vous devez afficher le fichier journal afin de déterminer la raison pour laquelle la méthode de démarrage a échoué. La sortie `svcs` indique le nom du fichier journal. Reportez-vous à [“Affichage des fichiers journaux d'un service” à la page 41](#) pour plus d'informations sur l'affichage du fichier journal. Dans cet exemple, le fichier de journalisation montre la méthode de démarrage d'appels et le message d'erreur fatale.

```
[ Sep 11 13:30:42 Executing start method ("/lib/svc/method/svc-pkg-depot start"). ]
pkg.depot-config: Unable to get publisher information:
The path '/export/ipsrepos/Solaris11' does not contain a valid package repository.
```

2. Résolvez le problème.

Une ou plusieurs des étapes suivantes peut alors s'avérer nécessaire.

■ Mettez à jour la configuration du service.

Si la résolution du problème a nécessité de modifier la configuration du service , utilisez la commande `svccfg refresh` ou `svcadm refresh` pour tous les services dont la configuration a changé. Assurez-vous que la configuration a été mise à jour dans l'instantané en cours d'exécution en utilisant la commande `svcprop` pour vérifier les valeurs de propriétés ou au moyen d'autres tests spécifiques au service.

■ Assurez-vous que les dépendances sont exécutées.

Parfois la ligne "Impact" de la sortie `svcs -x` vous indique que les services dépendant du service dans l'état maintenance ne fonctionnent pas. Utilisez la commande `svcs -l` pour vérifier l'état actuel des services dépendants. Assurez - vous que toutes les dépendances requises sont en cours d'exécution. Utilisez la commande `svcs -x` pour vérifier si tous les services activés sont en cours d'exécution.

■ ProcèsAssurez-vous que les processus de contrat sont arrêtés.

Si le service se trouvant dans l'état maintenance est un service de contrat, vérifiez si les processus démarrés par ce service ne se sont pas arrêtés. Lorsqu'une instance de service de contrat est à l'état de maintenance, l'ID de contrat doit être vide, comme l'illustre l'exemple suivant, et tous les processus associés à ce contrat doit être arrêtés. Utilisez `svcs -l` ou `svcs -o ctid` pour vous assurer qu'aucun contrat n'existe pour une instance de service en maintenance. Utilisez `svcs -p` pour vérifier si des processus associés à cette instance de service sont encore en cours d'exécution. Tous les processus affichés par `svcs -p` pour une instance de service en maintenance doivent être arrêtés.

```
$ svcs -l system-repository
fmri          svc:/application/pkg/system-repository:default
```

```

name          IPS System Repository
enabled       true
state         maintenance
next_state    none
state_time    September 17, 2013 07:18:19 AM PDT
logfile       /var/svc/log/application-pkg-system-repository:default.log
restarter     svc:/system/svc/restarter:default
contract_id
manifest      /lib/svc/manifest/application/pkg/pkg-system-repository.xml
dependency    require_all/error svc:/milestone/network:default (online)
dependency    require_all/none svc:/system/filesystem/local:default (online)
dependency    optional_all/error svc:/system/filesystem/autofs:default (online)

```

3. Notifiez l'agent de redémarrage du fait que l'instance est réparée.

Lorsque le problème signalé est réparé, utilisez la commande `svcadm clear` pour faire repasser le service à l'état `online`. Pour les services à l'état `maintenance`, la sous-commande `clear` indique à l'agent de redémarrage du service que celui-ci est réparé.

```
$ svcadm clear pkg/depot:default
```

Si vous spécifiez l'option `-s`, la commande `svcadm` attend que l'instance passe à l'état `online` ou jusqu'à ce qu'elle détermine que l'instance ne pourra pas passer à l'état `online` sans intervention de l'administrateur, avant de renvoyer un résultat. Utilisez l'option `-T` avec l'option `-s` pour spécifier un plafond en secondes pour accomplir la transition ou pour déterminer que la transition est impossible.

4. Vérifiez que l'instance est réparée.

Utilisez la commande `svcs` pour vérifier si le service qui était en `maintenance` est désormais en `ligne`. Utilisez la commande `svcs -x` pour vérifier si tous les services activés sont en cours d'exécution.

▼ Procédure de réparation d'une instance qui est hors ligne

Une instance de service hors ligne est activée, mais n'est pas en cours d'exécution ou disponible pour s'exécuter.

1. Déterminez la raison pour laquelle l'instance est hors ligne.

L'instance peut passer par l'état `offline` parce que ses dépendances ne sont pas encore satisfaites. Si l'instance est en transition, son état devrait s'afficher comme `offline*`.

2. Résolvez le problème.

■ Activez les dépendances de service.

Si les dépendances obligatoires sont désactivées, activez-les à l'aide de la commande suivante :

```
$ svcadm enable -r FMRI
```

■ **Corrigez le fichier de dépendance.**

Un fichier de dépendance peut être manquant ou illisible. Vous pouvez utiliser `pkg fix` ou `pkg revert` pour réparer ce type de problème. Reportez-vous à la page du manuel [pkg\(1\)](#).

3. Redémarrez l'instance si nécessaire.

Si l'instance était hors ligne, car une dépendance obligatoire n'a pas été satisfaite, le dépannage ou l'activation de la dépendance peut causer le redémarrage et la mise en ligne de l'instance hors ligne sans autre action administrative.

Si vous avez procédé à une autre correction du service, redémarrez l'instance.

```
$ svcadm restart FMRI
```

4. Vérifiez que l'instance est réparée.

Utilisez la commande `svcs` pour vérifier si l'instance qui était hors ligne est maintenant en ligne. Utilisez la commande `svcs -x` pour vérifier si tous les services activés sont en cours d'exécution.

▼ **Procédure de réparation d'une instance qui est endommagée**

Une instance de service qui est dégradée est activée et en cours d'exécution ou peut être exécutée, mais fonctionne à capacité limitée.

1. Déterminez la raison pour laquelle l'instance est dégradée.

2. Résolvez le problème.

3. Demandez à l'agent de redémarrage de mettre l'instance en ligne.

Lorsque le problème signalé est réparé, utilisez la commande `svcadm clear` pour faire repasser l'instance à l'état `online`. Pour les instances dans l'état `degraded`, la sous-commande `clear` demande à l'agent de redémarrage de l'instance de la repasser à l'état `online`.

```
$ svcadm clear pkg/depot:default
```

4. Vérifiez que l'instance est réparée.

Utilisez la commande `svcs` pour vérifier si l'instance qui était endommagée est maintenant en ligne. Utilisez la commande `svcs -x` pour vérifier si tous les services activés sont en cours d'exécution.

Marquage d'une instance comme dégradée ou en maintenance

Vous pouvez marquer une instance de service comme se trouvant dans l'état `degraded` ou `maintenance`. Cela peut vous être utile si l'application est bloquée dans une boucle ou verrouillée, par exemple. L'information sur le changement d'état se propage aux dépendances de l'instance marquée, ce qui aide au dégouage des instances associées.

Spécifiez l'option `-I` pour demander une modification d'état immédiate.

Lorsque vous marquez une instance comme `maintenance`, vous pouvez spécifier l'option `-t` pour demander une modification d'état temporaire. Les demandes temporaires ne durent que jusqu'au redémarrage.

Si vous spécifiez l'option `-s` avec la commande `svcadm mark`, `svcadm` marque l'instance et attend qu'elle passe à l'état `degraded` ou `maintenance` avant l'envoi d'un résultat. Utilisez l'option `-T` avec l'option `-s` pour spécifier un plafond en secondes pour accomplir la transition ou pour déterminer que la transition est impossible.

Diagnostic et réparation des problèmes de référentiel

Au démarrage du système, le démon du référentiel, `svc.configd`, effectue une vérification de l'intégrité du référentiel de configuration stocké dans `/etc/svc/repository.db`. En cas d'échec de la vérification d'intégrité de `svc.configd`, le démon `svc.configd` écrit à la console un message similaire à l'exemple suivant :

```
svc.configd: smf(5) database integrity check of:
```

```
    /etc/svc/repository.db
```

```
failed. The database might be damaged or a media error might have
prevented it from being verified. Additional information useful to
your service provider is in:
```

```
    /system/volatile/db_errors
```

```
The system will not be able to boot until you have restored a working
database. svc.startd(1M) will provide a sulogin(1M) prompt for recovery
purposes. The command:
```

```
    /lib/svc/bin/restore_repository
```

```
can be run to restore a backup version of your repository. See
```

<http://support.oracle.com/msg/SMF-8000-MY> for more information.

Le démon `svc.configd` se ferme ensuite. Cette sortie est détectée par le démon `svc.startd` ; ensuite, `svc.startd` lance `sulogin`.

A l'invite `sulogin`, saisissez Ctrl-D pour quitter `sulogin`. Le démon `svc.startd` reconnaît l'arrêt de `sulogin` et redémarre le démon `svc.configd` qui revérifie le référentiel. Le problème peut ne pas se reproduire après le redémarrage supplémentaire. N'appellez pas directement le démon `svc.configd`. Le démon `svc.startd` lance le démon `svc.configd`.

Si `svc.configd` signale un nouvel échec de la vérification d'intégrité et que l'invite `sulogin` apparaît, assurez-vous que les systèmes de fichiers requis ne sont pas remplis. Connectez-vous soit à distance, soit à l'invite `sulogin`, avec le mot de passe `root`. Vérifiez que l'espace est disponible sur les systèmes de fichiers `root` et `system/volatile`. Si l'un de ces systèmes de fichiers est plein, nettoyez et démarrez de nouveau le système. Si aucun de ces systèmes de fichiers n'est plein, suivez la procédure [“Restauration d'un référentiel à partir d'une sauvegarde” à la page 112](#).

Le référentiel de configuration de service peut être corrompu par l'un des motifs suivants :

- Défaillance du disque
- Bogue matériel
- Bogue logiciel
- Ecrasement accidentel du fichier

La procédure suivante montre comment remplacer un référentiel endommagé par une copie de sauvegarde du référentiel.

▼ Restauration d'un référentiel à partir d'une sauvegarde

1. Connectez-vous.

Connectez-vous soit à distance, soit à l'invite `sulogin`, avec le mot de passe `root`.

2. Exécutez la commande de restauration de référentiel :

```
# /lib/svc/bin/restore_repository
```

L'exécution de cette commande vous guide à travers les étapes nécessaires pour restaurer une sauvegarde non endommagée. SMF effectue automatiquement les sauvegardes du référentiel comme décrit dans [“Sauvegardes du référentiel” à la page 27](#).

SMF gère des données de configuration permanentes et non permanentes. Reportez-vous à la section [“Référentiel de configuration de service” à la page 24](#) pour une description de ces deux référentiels. La commande `restore_repository` restaure uniquement le référentiel permanent. En outre, la commande `restore_repository` réinitialise le système, ce qui détruit les données de configuration non permanentes. Les données non permanentes sont des données d'exécution qui ne sont pas nécessaires après une réinitialisation du système.

Lorsqu'elle est démarrée, la commande `/lib/svc/bin/restore_repository` affiche un message similaire au suivant :

See <http://support.oracle.com/msg/SMF-8000-MY> for more information on the use of this script to restore backup copies of the smf(5) repository.

If there are any problems which need human intervention, this script will give instructions and then exit back to your shell.

Une fois le système de fichiers root (/) installé avec des autorisations d'écriture, ou si le système est une zone locale, vous êtes invité à sélectionner la sauvegarde de référentiel à restaurer :

The following backups of `/etc/svc/repository.db` exists, from oldest to newest:

... *list of backups* ...

Le nom attribué à une sauvegarde dépend du type et de l'heure de son exécution. Les sauvegardes commençant par `boot` sont effectuées avant que la première modification ne soit apportée au référentiel après l'initialisation du système. Les sauvegardes commençant par `manifest_import` sont effectuées après que `svc:/system/manifest-import:default` termine son processus. L'heure de la sauvegarde est indiquée au format `YYYYMMDD_HHMMSS`.

3. Entrez la réponse appropriée.

En règle générale, la sauvegarde la plus récente est sélectionnée.

Please enter either a specific backup repository from the above list to restore it, or one of the following choices:

CHOICE	ACTION
-----	-----
boot	restore the most recent post-boot backup
manifest_import	restore the most recent manifest_import backup
-seed-	restore the initial starting repository (All customizations will be lost, including those made by the install/upgrade process.)
-quit-	cancel script and quit

Enter response [boot]:

Si vous appuyez sur la touche Entrée sans spécifier de sauvegarde à restaurer, la réponse par défaut, entourée de [], est sélectionnée. Sélectionnez `-quit-` pour quitter le script `restore_repository` et revenir à l'invite du shell.

Remarque - Sélectionnez `-seed-` pour restaurer le référentiel seed. Ce référentiel est conçu pour être utilisé lors de l'installation initiale et des mises à niveau. L'utilisation du référentiel seed à des fins de récupération doit être un dernier recours.

Une fois que la sauvegarde à restaurer a été sélectionnée, elle est validée et son intégrité est vérifiée. En cas de problèmes, la commande `restore_repository` imprime des messages d'erreur et vous invite à effectuer une autre sélection. Lorsqu'une sauvegarde valide est sélectionnée, les informations suivantes sont imprimées et vous êtes invité à confirmer l'opération.

After confirmation, the following steps will be taken:

```
svc.startd(1M) and svc.configd(1M) will be quiesced, if running.
/etc/svc/repository.db
-- renamed --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS
/system/volatile/db_errors
-- copied --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS_errors
repository_to_restore
-- copied --> /etc/svc/repository.db
and the system will be rebooted with reboot(1M).
```

Proceed [yes/no]?

4. Entrez **yes** pour résoudre la panne.

Le système se réinitialise après que la commande `restore_repository` a exécuté toutes les actions de la liste.

Spécification de la quantité de messages au démarrage

Par défaut, chaque service qui démarre lors de l'initialisation du système n'affiche aucun message sur la console. Utilisez l'une des méthodes suivantes pour modifier la liste des messages qui s'affichent sur la console et ceux qui sont enregistrés uniquement dans le fichier journal `svc.startd`. La valeur *logging-level* peut être l'une des valeurs du tableau ci-dessous.

- Lors de l'initialisation d'un système SPARC, spécifiez l'option `-m` pour la commande `boot` à l'invite `ok`. Reportez-vous à la section "Options des messages" à la page de manuel [kernel\(1M\)](#).

`ok boot -m logging-level`
- Lors de l'initialisation d'un système x86, modifiez le menu GRUB pour spécifier l'option `-m`. Reportez-vous à “ [Ajout d'arguments de noyau en modifiant le menu GRUB au moment de l'initialisation](#) ” du manuel “ [Initialisation et arrêt des systèmes Oracle Solaris 11.2](#) ” et à "Options des messages" à la page de manuel [kernel\(1M\)](#).
- Avant de réinitialiser un système, utilisez la commande `svccfg` pour modifier la valeur de la propriété `options/logging`. Si cette propriété n'a jamais été modifiée sur ce système, l'arrêt

est impossible et celle-ci devra être ajoutée. L'exemple suivant passe en message détaillé. La modification prend effet au prochain redémarrage du démon `svc.startd`.

```
$ svccfg -s system/svc/restarter:default listprop options/logging
$ svccfg -s system/svc/restarter:default addpg options application
$ svccfg -s system/svc/restarter:default setprop options/logging=verbose
$ svccfg -s system/svc/restarter:default listprop options/logging
options/logging astring      verbose
```

TABLEAU A-1 Niveaux de journalisation des messages de démarrage de SMF

Mot-clé de niveau de journalisation	Description
quiet	Affiche, sur la console, des message d'erreur nécessitant une intervention de l'administrateur. Enregistre ces messages dans <code>syslog</code> et <code>/var/svc/log/svc.startd.log</code> .
verbose	En plus des messages fournis au niveau quiet, affiche sur la console un message unique pour chaque service démarré, et enregistre dans <code>/var/svc/log/svc.startd.log</code> des informations sur les erreurs qui ne nécessitent pas une intervention de l'administrateur.
debug	En plus des messages fournis au niveau quiet, affiche sur la console un message unique pour chaque service démarré, et enregistre tous les messages de débogage <code>svc.startd</code> dans <code>/var/svc/log/svc.startd.log</code> .

Spécification du jalon SMF sur lequel initialiser

Lors de l'initialisation d'un système, vous pouvez spécifier le jalon SMF sur lequel effectuer l'initialisation.

Par défaut, tous les services pour lesquels la valeur de la propriété `general/enabled` est `true` sont démarrés à l'initialisation du système. Pour modifier le jalon sur lequel initialiser un système, utilisez l'une des méthodes suivantes. La valeur de *milestone* peut être le FMRI d'un service ou d'un mot-clé jalon, comme illustré au [Tableau A-2, “Jalons d'initialisation SMF et niveaux d'exécution correspondant”](#).

- Lors de l'initialisation d'un système SPARC, spécifiez l'option `-m` pour la commande `boot` à l'invite `ok`. Voir l'option `-m` à la page de manuel [kernel\(1M\)](#).

```
ok boot -m milestone=milestone
```

- Lors de l'initialisation d'un système x86, modifiez le menu GRUB pour spécifier l'option `-m`. Voir “[Ajout d'arguments de noyau en modifiant le menu GRUB au moment de l'initialisation](#)” du manuel “[Initialisation et arrêt des systèmes Oracle Solaris 11.2](#)” et l'option `-m` à la page de manuel [kernel\(1M\)](#).
- Avant de réinitialiser un système, utilisez la commande `svcadm milestone` avec l'option `-d`. Notez qu'avec ou sans l'option `-d`, cette commande limite et restaure immédiatement les services en cours d'utilisation. Avec l'option `-d`, la commande fait aussi du jalon spécifié

le jalon d'initialisation par défaut. Cette nouvelle valeur par défaut persiste après les réinitialisations.

```
$ svcadm milestone -d milestone
```

Cette commande ne modifie pas le niveau d'exécution actuel du système. Pour modifier le niveau d'exécution actuel du système, utilisez la commande `init`.

Si vous spécifiez l'option `-s`, `svcadm` modifie le jalon, puis attend la fin de la transition vers le jalon actuel avant d'envoyer un résultat. La commande `svcadm` renvoie un résultat lorsque toutes les instances passe à l'état nécessaire pour atteindre le jalon spécifié ou lorsqu'elle estime qu'une intervention de l'administrateur est nécessaire pour effectuer une transition. Utilisez l'option `-T` avec l'option `-s` pour spécifier un plafond en secondes pour accomplir le changement de jalon ou le renvoi.

Le tableau suivant décrit les jalons SMF d'initialisation, y compris les niveau d'exécution Oracle Solaris correspondant. Le *niveau d'exécution* d'un système définit les services et les ressources à la disposition des utilisateurs. Un système peut être dans un seul niveau d'exécution à la fois. Pour plus d'informations sur les niveaux d'exécution, reportez-vous à la section “[Fonctionnement des niveaux d'exécution](#)” du manuel “[Initialisation et arrêt des systèmes Oracle Solaris 11.2](#)”, à la page de manuel [inittab\(4\)](#) et au fichier `/etc/init.d/README`. Pour plus d'informations sur les jalons d'initialisation SMF, voir la sous-commande `milestone` à la page de manuel [svcadm\(1M\)](#).

TABEAU A-2 Jalons d'initialisation SMF et niveaux d'exécution correspondant

Mot-clé ou FMRI du jalon SMF	Niveau d'exécution correspondant	Description
none		Le mot-clé <code>none</code> représente un jalon où le seul service qui s'exécute est l'agent de redémarrage principal. Si <code>none</code> est spécifié, tous les services à part <code>svc:/system/svc/restarter:default</code> sont temporairement désactivés. Le jalon <code>none</code> peut être utile pour le débogage des problèmes de démarrage. Voir “ Démarrage Services comment Examiner des problèmes lors de l'initialisation du système ” à la page 117 pour des instructions plus détaillées.
all		Le mot-clé <code>all</code> représente un jalon qui dépend de tous les services. Si <code>all</code> est spécifié, les demandes d'activation et de désactivation sont ignorées pour tous les systèmes. Il s'agit du jalon par défaut utilisé par <code>svc.startd</code> .
svc:/milestone/single-user	s ou S	Ignorer les demandes d'activation et de désactivation temporaire pour <code>svc:/milestone/single-user:default</code> et tous les services dont il dépend, que ce soit directement ou indirectement. Désactiver temporairement tous les autres services.
svc:/milestone/multi-user	2	Ignorer les demandes d'activation et de désactivation temporaire pour <code>svc:/milestone/multi-user:default</code> et tous les services dont il dépend, que ce soit directement ou indirectement. Désactiver temporairement tous les autres services.

Mot-clé ou FMRI du jalon SMF	Niveau d'exécution correspondant	Description
svc:/milestone/multi-user-server	3	Ignorer les demandes d'activation et de désactivation temporaire pour svc:/milestone/multi-user-server:default et tous les services dont il dépend, que ce soit directement ou indirectement. Désactiver temporairement tous les autres services.

Pour déterminer le jalon auquel un système s'est actuellement initialisé, utilisez la commande `svcs`. L'exemple suivant montre que le système est initialisé au niveau d'exécution 3, `milestone/multi-user-server` :

```
$ svcs 'milestone/*'
STATE      STIME      FMRI
online     9:08:05   svc:/milestone/unconfig:default
online     9:08:06   svc:/milestone/config:default
online     9:08:07   svc:/milestone/devices:default
online     9:08:25   svc:/milestone/network:default
online     9:08:31   svc:/milestone/single-user:default
online     9:08:51   svc:/milestone/name-services:default
online     9:09:13   svc:/milestone/self-assembly-complete:default
online     9:09:23   svc:/milestone/multi-user:default
online     9:09:24   svc:/milestone/multi-user-server:default
```

Utilisation de SMF pour rechercher les problèmes d'initialisation du système

Cette section décrit les actions à entreprendre si votre système se bloque durant l'initialisation ou si une clé cas d'échec du démarrage du service au cours de l'initialisation.

▼ Démarrage Services comment Examiner des problèmes lors de l'initialisation du système

En cas de problème lors du démarrage des services à l'initialisation du système, il arrive parfois que le système se bloque pendant l'initialisation. Cette procédure montre comment examiner les services des problèmes qui se produisent au moment de l'initialisation.

1. Initialisez le système sans démarrer de services.

Cette commande indique au démon `svc.startd` de désactiver temporairement tous les services et de démarrer `sulogin` sur la console.

```
ok boot -m milestone=none
```

Reportez-vous à [“Spécification du jalon SMF sur lequel initialiser”](#) à la page 115 pour une liste de jalons SMF que vous pouvez utiliser avec la commande `boot -m`.

2. **Connectez-vous au système en tant qu'utilisateur root.**

3. **Activez tous les services.**

```
# svcadm milestone all
```

4. **Déterminez l'endroit où le processus d'initialisation est bloqué.**

Lorsque le processus d'initialisation se bloque, déterminez les services qui ne sont pas en cours d'exécution en exécutant `svcs`. Recherchez des messages d'erreur dans les fichiers journaux dans `/var/svc/log`.

5. **Après avoir corrigé les problèmes, vérifiez que tous les services ont démarré.**

a. **Vérifiez que tous les services requis sont en ligne.**

```
# svcs -x
```

b. **Vérifiez que les dépendances de service `console-login` sont satisfaites.**

Cette commande vérifie que le processus `login` sur la console va s'exécuter.

```
# svcs -l system/console-login:default
```

6. **Poursuivez le processus d'initialisation normal.**

▼ **Procédure d'affichage forcée d'un utilisateur système de fichiers de connexion si la fonction Local Service Fails During Boot**

Les systèmes de fichiers locaux qui ne sont pas requis pour initialiser le système sont montés par le service `svc:/system/filesystem/local:default`. Lorsqu'un de ces systèmes de fichiers ne peut pas être monté, le service `filesystem/local` passe à l'état de maintenance. Le démarrage du système se poursuit et les services qui ne dépendent pas de `filesystem/local` sont démarrés. Les services ayant une dépendance obligatoire sur le service `filesystem/local` ne sont pas démarrés.

Cette procédure explique comment modifier la configuration du système de sorte qu'une invite `su login` s'affiche immédiatement après l'échec du service au lieu de permettre la poursuite du démarrage du système.

1. **Modifiez le service `system/console-login`.**

```
$ svccfg -s svc:/system/console-login
svc:/system/console-login> addpg site,filesystem-local dependency
svc:/system/console-login> setprop site,filesystem-local/entities = fmri: svc:/system/
filesystem/local
```

```

svc:/system/console-login> setprop site,filesystem-local/grouping = astring: require_all
svc:/system/console-login> setprop site,filesystem-local/restart_on = astring: none
svc:/system/console-login> setprop site,filesystem-local/type = astring: service
svc:/system/console-login> end

```

2. Actualisez le service.

```
$ svcadm refresh console-login
```

En cas d'échec avec le service `system/filesystem/local:default`, utilisez la commande `svcs -vx` pour identifier l'échec. Une fois l'échec corrigé, utilisez la commande suivante pour effacer l'état d'erreur et autoriser la poursuite de l'initialisation du système, procédez comme suit :

```
$ svcadm clear filesystem/local
```

Conversion des services `inetd` en services SMF

Le fichier `inetd.conf` sur votre système ne doit contenir aucune entrée. Le fichier `inetd.conf` doit contenir uniquement les commentaires indiquant qu'il s'agit d'un fichier hérité qui n'est plus utilisé directement. Si le fichier `inetd.conf` contient des entrées, suivez les instructions de cette section pour convertir ces configurations en services SMF. Les services configurés dans le fichier `inetd.conf` mais non configurés en tant que services SMF ne peuvent pas être utilisés. Les services configurés dans le fichier `inetd.conf` ne sont pas directement redémarrés par la commande `inetd`. La commande `inetd` est l'agent de redémarrage délégué pour les services convertis.

Durant l'initialisation du système, les configurations du fichier `inetd.conf` sont automatiquement converties en services SMF. Après l'initialisation du système, des entrées peuvent être ajoutées au fichier `inetd.conf` en installant de nouveaux logiciels non fournis en packages Image Packaging System (IPS). IPS logiciel distribué par manifeste SMF les packages comprend éventuellement requises manifeste instance et que l'instance de service SMF avec la bonne que les valeurs de propriété.

Si le fichier `inetd.conf` sur votre système contient des entrées, utilisez la commande `inetconv` pour convertir ces configurations en services SMF. La commande `inetconv` convertit les entrées de `inetd.conf` en fichiers de manifeste de service SMF, puis importe ces manifestes dans le référentiel SMF pour instancier les instances de services. Reportez-vous à la page de manuel [inetconv\(1M\)](#) pour des informations sur les options de commande et pour voir des exemples d'utilisation de la commande.

Le nom du nouveau manifeste SMF comprend le `service_name` de l'entrée `inetd.conf`. L'entrée du fichier `inetd.conf` est enregistrée en tant que propriété de la nouvelle instance de service. Le nouveau manifeste SMF spécifie les propriétés et groupes de propriétés pour définir les actions répertoriées dans l'entrée `inetd.conf`. Après l'exécution de la commande `inetconv`,

utilisez les commandes `svcs` et `svccprop` pour vérifier si la nouvelle instance de service a été créée, avec les valeurs de propriétés correctes.

La commande `inetd` est l'agent de démarrage délégué pour les services internet SMF. N'utilisez pas directement la commande `inetd` pour gérer ces services. Utilisez la commande `inetadm` sans options ou opérandes pour afficher la liste des services contrôlés par `inetd`. Utilisez les commandes `inetadm`, `svcadm` et `svccfg` pour configurer et gérer les services convertis.

La commande `inetconv` ne modifie pas le fichier d'entrée `inetd.conf`. Vous devez supprimer manuellement toutes les entrées dans le fichier `inetd.conf` après avoir exécuté correctement `inetconv`.

Pour plus d'informations sur la configuration des services `inetd` déjà convertis en services SMF, reportez-vous à [“Modification des services contrôlés par `inetd`” à la page 82](#).

Index

A

- actions, 18
- activation d'un service, 53
- actualisation de la configuration de service, 59
- affichage composé, 24, 43
- agent de redémarrage de service, 22
- agent de redémarrage par défaut, 22
- agents de redémarrage, 18, 22
 - agent de redémarrage par défaut, 22
 - agents de redémarrage délégués, 23
 - démon d'agent de redémarrage principal
 - `svc.startd`, 22
- agents de redémarrage délégués, 23
- arrêt d'un service, 56
- ASM, 89
- ASM (Automatic Storage Management), 89
- autorisations, 30

B

- base de données SMF *Voir* référentiel de configuration de service
- bibliothèque `libscf`, 24, 98
 - fonctions de traitement par lots, 16
- bibliothèque Utilitaire de configuration de service, `libscf`, 24
- bundle support, 77

C

- cliché en cours d'exécution, 43
- cliché `initial`, 27
- cliché `previous`, 27
- cliché `running`, 27
- cliché `start`, 27
- clichés, 45, 48

- cliché en cours d'exécution, 43
- `initial`, 27
- mise à jour de la configuration en cours d'exécution, 59
- `previous`, 27
- rétablissement, 79
- `running`, 27
- `start`, 27
- colonne `STATE`, 31
- comande `svccfg`
 - sous-comande `delcust`, 77
- commande `boot`
 - jalón `none`, 117
 - option `milestone`, 115
- commande `inetadm`, 119
 - exemple, 82
- commande `inetconv`, 119
- commande `inetd`, 119
- commande `restore_repository`, 112
- commande `sulogin`, 112
- commande `svcadm`, 53
 - options synchrones, 16
 - sous-commande `clear`, 107
 - sous-commande `disable`, 56
 - sous-commande `enable`, 53
 - sous-commande `mark`, 111
 - sous-commande `milestone`, 115, 117
 - sous-commande `refresh`, 59
 - sous-commande `restart`, 58, 107
 - sous-commande `restart manifest-import`, 58, 60
- commande `svcbundle` `command`
 - création de profils, 81
- commande `svcbundle`,
 - création de manifestes, 87

- service rc-script, 97
- commande svccfg
 - affichage des propriétés, 42
 - fichier d'entrée de commande, 68
 - sous-commande add, 79
 - sous-commande addpg , 73
 - sous-commande addpropvalue, 68, 73
 - sous-commande apply, 80
 - sous-commande delcust, 75
 - sous-commande delcust -M, 77
 - sous-commande delete, 60
 - sous-commande delpg, 75
 - sous-commande delprop, 75
 - sous-commande delpropvalue, 75
 - sous-commande describe, 42
 - sous-commande editprop, 66
 - sous-commande extract, 82
 - sous-commande import, 80
 - sous-commande listcust, 45, 49
 - sous-commande listcust -M, 60, 77
 - sous-commande listpg, 46
 - sous-commande listprop, 45
 - sous-commande listsnap, 48
 - sous-commande refresh, 59
 - sous-commande revert, 79
 - sous-commande selectsnap, 45, 79
 - sous-commande setenv, 68, 72
 - sous-commande setprop, 68, 73, 75
 - sous-commande unsetenv, 72
 - utilisation interactive, 67
- commande svccfg,
 - validation de sous-commande, 88
- commande svcprop
 - affichage des propriétés, 42, 43
- commande svcs, 31
- configuration
 - actualisation, 59
 - ajout de propriétés et groupes de propriétés, 73
 - ajout de valeurs de propriété, 68
 - définition des valeurs de propriété, 68
 - démasquage, 77
 - masquage, 77
 - modification, 65

- suppression de configuration bundle-supported, 77
- suppression de groupes de propriétés, 75
- suppression de personnalisations, 75, 77
- suppression de propriétés, 75
- suppression de valeurs de propriété, 75
- configuration des notifications, 62
- configuration masquée, 77
- connexion monutilisateur, 112
- consignation d'erreurs, 41
- couche admin, 26, 49
- couche manifest, 26, 49
- couche site-profile, 26, 49
- couche system-profile, 26
- couches, 26
 - couche admin, 49
 - couche manifest, 49
 - couche site-profile, 49

D

- démarrage d'un service, 53
- démon d'agent de redémarrage principal, 22
- démon d'agent de redémarrage principal svc.startd, 22
- démon de redémarrage principal svc.startd, 112
- démon de référentiel svc.configd, 111
- dépendances, 21, 21
 - effet sur l'état d'un service dépendant, 38
 - groupes, 35
 - instances dépendant d'un service, 37
 - instances dont dépend un service, 36
 - liste, 34
- désactivation d'un service, 56
- données modèles, 24
- DTD, 87

E

- ensembles, 25
- ensembles de services
 - emplacements standard, 25
- ensembles de services,
 - DTD, 87
- entités masquées, 60

entités masquées, 49
entrées `inittab`, 28
état de service `degraded`, 21, 107, 111
état de service `disabled`, 21
état de service `legacy_run`, 31
état de service `maintenance`, 21, 107, 111
état de service `offline`, 21, 107
état de service `online`, 21
état de service `uninitialized`, 21
états d'initialisation *Voir* niveaux d'exécution
états de service
 description, 21
 modification manuelle, 107, 111
états de services
 transitions, 21
états des services
 liste, 31
fichier `/etc/inetd.conf`, 28, 84, 119
scripts `/etc/init.d`, 28
entrées `/etc/inittab`, 28
scripts `/etc/rc?.d`, 28

F

Fault Management Resource Identifier *Voir* FMRI
fichier `inetd.conf`, 28, 119
fichier `resolv.conf`, 28
fichiers de configuration, 15, 16, 28, 98
fichiers journaux, 41
fichiers pochoir, 16, 98, 98
FMRI, 20, 97

G

groupe de propriétés `config`, 23
groupe de propriétés `general`, 23
groupe de propriétés `rester`, 23
groupe de propriétés `start`, 23
groupes de propriétés, 23, 42
 non permanents, 46
 type, 46
groupes de propriétés de service, 23, 42
groupes de propriétés non permanents, 46
groupes de propriétés,

nommage, 96
type, 97

I

initialisation
 jalon actuel, 117
 jalons, 115
 niveaux de journalisation, 114
 sans démarrer des services, 117
 vers un jalon SMF, 117
instances, 18
 ajout, 79
 nommage, 20
instances de service, 18
 ajout, 79
instances,
 nommage, 96

J

jalons
 actuellement initialisés, 117
 initialisation, 115
 niveaux d'exécution correspondants, 115
 none, 117

K

Kerberos
 exemple service pochoir, 103

M

manifestes, 18
 emplacement standard, 25
manifestes,
 emplacement standard, 87
 format, 87
 répertoire `site`, 87
méthodes, 18
mise à jour de la configuration de service, 59
modèles de service, 20

N

niveaux d'exécution
 actuel, 117
 jalons SMF correspondants, 115
 niveau par défaut, 115
notification d'événement, 62
notification de transition d'état, 62

O

Oracle Database, 89
 ASM, 89
 service de base de données, 90
 service listener, 93

P

paramètres de notification
 affichage, 49
permissions, 30
personnalisations
 liste, 49
privilèges, 30
profils, 18
 création, 80
 emplacement standard, 25
 répertoire site, 25, 80
profils de droits, 30
propriétés, 23, 24, 42
 affichage composé, 24, 43
 dans le cliché en cours d'exécution, 43
 non permanentes, 46
propriétés de service, 23, 24, 42
propriétés non permanentes, 47
propriétés,
 nommage, 96
Puppet,
 exemple de service pochoir, 99

R

redémarrage d'un service, 58
redémarrage du service `manifest-import`, 58
référentiel *Voir* référentiel de configuration de service

référentiel de configuration *Voir* référentiel de configuration de service
référentiel de configuration de service, 18
 clichés, 27
 couches, 26
 démon `svc.configd`, 111
 données modèles, 24
 interfaces de bibliothèque, 24
 modification, 65
 réparation, 111
 sauvegardes, 27
référentiel de configuration de service., 24
rôles, 30

S

sauvegardes, 27
sauvegardes boot, 27
sauvegardes `manifest-import`, 27
scripts de contrôle d'exécution, 28
scripts de contrôle d'exécution,
 conversion en service SMF, 97
scripts `init.d`, 28
scripts `rc?.d`, 28
service d'attente, 20
service de contrat, 20, 20
service démon, 20
service enfant, 20
service `inetd`
 conversion en service SMF, 119
 modification des arguments de ligne de commande, 82
 modification des propriétés, 82
service pochoir, 98
 exemple Kerberos, 103
service pochoir,
 exemple Puppet, 99
service transitoire, 20
services, 18
 activation, 53
 actualisation de la configuration, 59
 arrêt, 56
 démarrage, 53
 désactivation, 56
 modification de la configuration, 65

- nommage, 20
- redémarrage, 58
- suppression, 60
- services de jalon, 18
- services lrc, 28
- services,
 - nommage, 96
- sous-commande describe
 - commande svccfg, 42
- sous-commande listcust
 - commande svccfg, 45, 49
- sous-commande listpg
 - commande svccfg, 46
- sous-commande listprop
 - commande svccfg, 45
- sous-commande listsnap
 - commande svccfg, 48
- sous-commande selectsnap
 - commande svccfg, 45
- suppression d'un service, 60, 60
- svcbundle, commande
 - installation automatique, 82, 89

U

- utilitaire svcio, 16, 98

V

- variables d'environnement
 - modification dans les méthodes, 72

