



Guide d'administration système du serveur Netra™ 240

Sun Microsystems, Inc.
www.sun.com

Référence 817-5010-11
Juillet 2004, Révision A

Envoyez vos remarques à propos de ce document à l'adresse : <http://www.sun.com/hwdocs/feedback>

Copyright 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuelle relatants à la technologie qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats-Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, OpenBoot, Netra SunVTS, Sun Enterprise Authentication Mechanism, et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licences de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISÉE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Adobe PostScript

Table des matières

Préface xi

1. Outils de dépannage 1

Aperçu des outils de diagnostic 2

Invites système 3

Advanced Lights Out Manager 4

Indicateurs d'état des serveurs 4

▼ Pour afficher l'état du voyant Localisation 5

▼ Pour allumer le voyant Locator 5

▼ Pour éteindre le voyant Locator 5

Indicateurs d'état des alarmes 6

Diagnostics d'auto-test à la mise sous tension 8

Contrôle des diagnostics POST 9

▼ Pour lancer les diagnostics POST 11

Commandes OpenBoot 11

Commandes `probe-scsi` et `probe-scsi-all` 12

Commande `probe-ide` 13

Commande `show-devs` 13

▼ Pour lancer les commandes OpenBoot 14

OpenBoot Diagnostics	14
▼ Pour lancer OpenBoot Diagnostics	15
Contrôle des tests OpenBoot Diagnostics	16
Commandes <code>test</code> et <code>test-all</code>	17
Messages d'erreur OpenBoot Diagnostics	17
Outils de diagnostic du système d'exploitation	18
Fichiers journaux de messages d'erreur et messages système	19
Commandes d'information du système logiciel Solaris	19
Commande <code>prtconf</code>	19
Commande <code>prtdiag</code>	21
Commande <code>prtfwu</code>	23
Commande <code>psrinfo</code>	24
Commande <code>showrev</code>	25
▼ Pour exécuter les commandes d'informations système de la plate-forme Solaris	26
Résultats de tests de diagnostic récents	27
▼ Pour afficher les résultats des tests récents	27
Variables de configuration OpenBoot	27
▼ Pour afficher et définir les variables de configuration OpenBoot	28
Utilisation des commandes <code>watch-net</code> et <code>watch-net-all</code> pour vérifier les connexions réseau	28
Reprise automatique du système	29
Options Auto-Boot	30
Récapitulatif sur le traitement des erreurs	31
Scénarios de réinitialisation	32
▼ Pour activer ASR	32
▼ Pour désactiver ASR	33

2. Logiciel SunVTS	35
Présentation du logiciel SunVTS	35
Tests SunVTS	36
Logiciel SunVTS et sécurité	37
▼ Pour vérifier l'installation du logiciel SunVTS	38
Installation du logiciel SunVTS	38
Consultation de la documentation SunVTS	39
3. Advanced Lights Out Manager	41
Présentation du logiciel Advanced Lights Out Manager	41
Ports ALOM	42
Définition du mot de passe admin	43
Fonctions ALOM de base	43
▼ Pour passer à l'invite ALOM	44
▼ Pour passer à l'invite de la console serveur	44
▼ Pour supprimer les droits d'écriture d'un utilisateur sur la console	44
Redémarrage automatique du serveur	45
Surveillance et contrôle de l'environnement	46
A. API de sortie des relais d'alerte	49
Index	55

Figure

- FIGURE 1-1 Circulation des invites système 3
- FIGURE 1-2 Emplacement des indicateurs du panneau avant 4

Tableaux

TABLEAU 1-1	Résumé d'outils de dépannage	2
TABLEAU 1-2	Voyants d'état du serveur (avant et arrière)	4
TABLEAU 1-3	Indicateurs d'alarme et états de l'alarme du contact sec	6
TABLEAU 1-4	Variables de configuration OpenBoot	9
TABLEAU 1-5	Mots clés associés à la variable de configuration OpenBoot <code>test-args</code>	16
TABLEAU 1-6	Commandes d'affichage des informations de la plate-forme Solaris	26
TABLEAU 2-1	Tests du logiciel SunVTS	36
TABLEAU 3-1	Composants surveillés par ALOM	42
TABLEAU 3-2	Seuils de température du système du serveur Netra 240	47

Préface

Le *Guide d'administration système du serveur Netra 240* s'adresse aux administrateurs système expérimentés. Il propose une description générale des outils de diagnostic du serveur Netra™ 240 et de différentes tâches d'administration du serveur.

Pour une utilisation optimale des informations contenues dans ce manuel, vous devez posséder une expérience pratique des notions et des termes relatifs aux réseaux informatiques, ainsi qu'une connaissance approfondie du système d'exploitation Solaris™.

Avant de lire cet ouvrage

Ce manuel ne traite pas de l'installation et du montage en rack des serveurs. Pour plus de détails à ce sujet, veuillez consulter le *Guide d'installation du serveur Netra 240* (817-4994-11).

Vous devez lire le *Important Safety Information for Sun Hardware Systems* (816-7190-10) avant de suivre les instructions fournies dans ce manuel.

Utilisation des commandes UNIX

Utilisez cette section pour avertir les lecteurs que toutes les commandes UNIX ne sont pas fournies. Par exemple :

Ce manuel ne contient pas d'informations sur les commandes et les procédures UNIX® de base, telles que l'arrêt ou l'initialisation du système et la configuration des périphériques. Pour plus d'informations, reportez-vous aux documents suivants :

- Toute autre documentation sur les logiciels livrée avec votre système
- Contenant la documentation de l'environnement d'exploitation Solaris™ située à l'adresse suivante :

<http://docs.sun.com>

Invites Shell

Shell	Invite
C shell	<i>nom-machine%</i>
C shell superutilisateur	<i>nom-machine#</i>
Bourne shell et Korn shell	\$
Bourne shell et Korn shell superutilisateur	#

Conventions typographiques

Mise en forme ⁱ	Description	Exemples
AaBbCc123	Noms de commandes, fichiers et répertoires ; messages-système	Éditez votre fichier <code>.login</code> . Utilisez la commande <code>ls -a</code> pour obtenir la liste de tous les fichiers. % You have mail.
AaBbCc123	Ce que l'utilisateur tape par opposition aux messages apparaissant à l'écran.	% su Password:
<i>AaBbCc123</i>	Titres de guide, nouveaux mots ou termes, mots à mettre en valeur. Remplacez les variables de ligne de commande par des noms ou des valeurs réels.	Lisez le chapitre 6 du <i>Guide de l'utilisateur</i> . Ces options sont appelées options de <i>classe</i> . Pour effectuer cette opération, vous devez être superutilisateur. Pour supprimer un fichier, tapez <code>rm nom_fichier</code> .

i Les paramètres de votre navigateur peuvent être différents de ces paramètres.

Documentation connexe

Application	Titre	Numéro de référence
Présentation générale de l'installation	<i>Netra 240 Server Quick Start Guide</i>	817-3904-xx
Dernières mises à jour du produit	<i>Netra 240 Server Release Notes</i>	817-3142-xx
Conformité et sécurité	<i>Important Safety Information for Sun Hardware Systems</i>	816-7190-10
	<i>Netra 240 Server Safety and Compliance Manual</i>	817-5018-11
Emplacement du site Web de documentation	<i>Sun Netra 240 Server Documentation</i>	817-2697-10
Installation	<i>Guide d'installation du serveur Netra 240</i>	817-4994-11
Lights-Out Management	<i>Guide l'utilisateur du logiciel Advanced Lights Out Manager de Sun pour le serveur Netra 240</i>	817-5002-11
Maintenance	<i>Netra 240 Server Service Manual</i>	817-2699-xx

Accès à la documentation Sun

Vous pouvez afficher, imprimer ou acquérir un grand choix de documentation Sun, y compris des versions localisées, à l'adresse :

<http://www.sun.com/documentation>

Sites Web tiers

Sun n'est pas responsable de la disponibilité des sites Web tiers cités dans le présent document. Sun n'approuve pas et ne peut être tenu responsable du contenu, des publicités, des produits ou autres documents rendus disponibles directement ou indirectement par lesdits sites ou ressources. Sun ne peut être tenu responsable des pertes ou dégâts avérés ou présumés découlant directement ou indirectement de l'utilisation de tout contenu, bien ou service rendu disponible directement ou indirectement par lesdits sites ou ressources.

Contactez l'assistance technique de Sun

Si ce document ne répond pas à toutes vos questions techniques sur ce produit, consultez le site Web suivant :

<http://www.sun.com/service/contacting>

Vos commentaires sont les bienvenus chez Sun

Dans le souci d'améliorer notre documentation, tous vos commentaires et suggestions sont les bienvenus. Vous pouvez nous soumettre vos commentaires à l'adresse :

<http://www.sun.com/hwdocs/feedback>

Veillez à mentionner le titre et la référence de votre document :

Guide d'administration système du serveur Netra 240, 817-5010-11

Outils de dépannage

Ce chapitre décrit les outils de diagnostic disponibles pour le serveur Netra 240. Il comprend les sections suivantes :

- « Aperçu des outils de diagnostic » à la page 2
- « Invites système » à la page 3
- « Advanced Lights Out Manager » à la page 4
- « Diagnostics d'auto-test à la mise sous tension » à la page 8
- « Commandes OpenBoot » à la page 11
- « OpenBoot Diagnostics » à la page 14
- « Outils de diagnostic du système d'exploitation » à la page 18
- « Résultats de tests de diagnostic récents » à la page 27
- « Variables de configuration OpenBoot » à la page 27
- « Reprise automatique du système » à la page 29

Aperçu des outils de diagnostic

Sun offre une gamme d'outils de diagnostic à utiliser avec le serveur Netra 240 de la manière résumée dans la table suivante.

TABLEAU 1-1 Résumé d'outils de dépannage

Outil de diagnostic	Type	Description	Accessibilité et disponibilité	Capacité à distance
ALOM	Matériel et logiciel	Surveille les conditions environnementales, effectue l'isolation d'erreurs de base et permet l'accès à distance à la console.	Peut fonctionner en mode veille et sans système d'exploitation.	Conçu pour l'accès à distance.
Voyants	Matériel	Indiquent l'état général du système et de certains composants.	Accès à partir du châssis du système. Disponibles dès la mise sous tension.	Accès local, mais peut être visualisé via ALOM.
Auto-test à la mise sous tension (POST)	Micro-programme	Teste les composants essentiels du système.	S'exécute automatiquement au démarrage. Disponibles quand le système d'exploitation ne fonctionne pas.	Accès local, mais peut être visualisé via ALOM.
Commandes OpenBoot	Micro-programme	Affichent différents types d'informations système.	Disponibles quand le système d'exploitation ne fonctionne pas.	Accès local, mais accessible via ALOM.
Diagnostics OpenBoot	Micro-programme	Teste les composants du système tout en se concentrant sur les périphériques et les périphériques d'E/S.	S'exécute automatiquement ou par interaction. Disponibles quand le système d'exploitation ne fonctionne pas.	Accès local, mais peut être visualisé via ALOM.
Commandes Solaris software	Logiciel	Affichent différents types d'informations système.	Système d'exploitation obligatoire.	Accès local, mais accessible via ALOM.
Logiciel SunVTS™	Logiciel	Vérifie et éprouve le système en exécutant des tests en parallèle.	Système d'exploitation obligatoire. Module en option.	Affichable et contrôlable en réseau.

Invites système

Les invites par défaut suivantes sont utilisées par le serveur Netra 240 :

- `ok` : Invite OpenBoot PROM
- `sc>` : invite Advanced Lights Out Manager (ALOM)
- `#` : invite superutilisateur du logiciel Solaris (Bourne shell et Korn shell)

La FIGURE 1-1 illustre la relation entre ces trois invites et indique comment passer de l'une à l'autre.

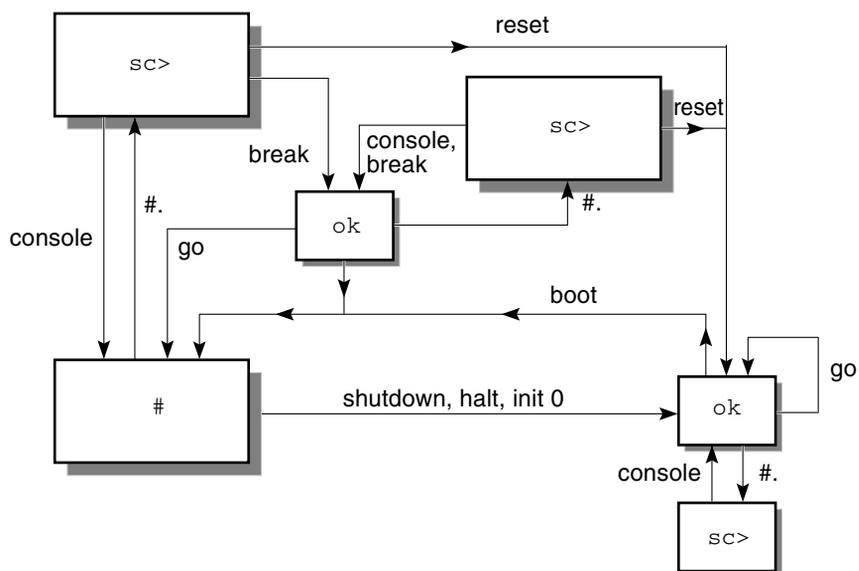


FIGURE 1-1 Circulation des invites système

Les commandes suivantes figurent dans le diagramme de circulation représenté à la FIGURE 1-1 :

- Commandes ALOM : `console`, `reset`, `break`
- Séquence d'échappement : `#.`
- Commandes logicielles de Solaris : `shutdown`, `halt`, `init 0`
- Commandes OpenBoot : `go`, `boot`

Advanced Lights Out Manager

Le logiciel Advanced Lights Out Manager (ALOM) de Sun™ pour le serveur Netra 240 propose différents voyants d'état. Cette section précise la signification de leur état et indique comment les activer et les désactiver. Pour plus d'informations sur ALOM, reportez-vous au Chapitre 3.

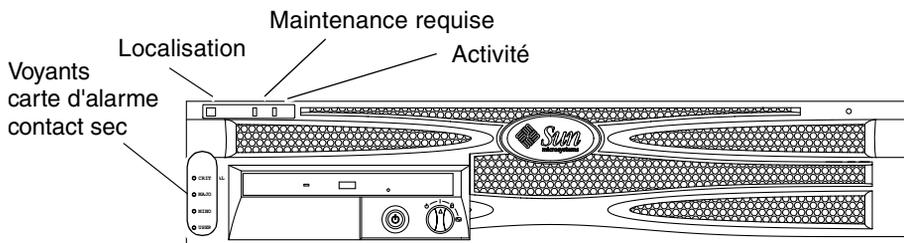


FIGURE 1-2 Emplacement des indicateurs du panneau avant

Indicateurs d'état des serveurs

Les serveurs comportent trois voyants d'état. Ils sont situés sur le cache avant (FIGURE 1-2) et reproduits sur le panneau arrière. Le TABLEAU 1-2 résume les fonctions des voyants.

TABLEAU 1-2 Voyants d'état du serveur (avant et arrière)

Voyant	Couleur du voyant	État du voyant	Description
Activité	Vert	Allumé	Le serveur est sous tension et exécute l'environnement d'exploitation Solaris.
		Éteint	Soit le serveur n'est pas alimenté, soit le système d'exploitation Solaris ne fonctionne pas.
Maintenance requise	Jaune	Allumé	Le serveur a détecté un problème et doit être vérifié par le personnel technique.
		Éteint	Le serveur ne présente aucune défaillance.
Localisation	Blanc	Allumé	Une lumière continue s'allume et identifie le serveur parmi les autres éléments d'un châssis lorsque la commande <code>setlocator</code> est utilisée.

Vous pouvez activer et désactiver le voyant Localisation à partir de la console système ou de l'interface de ligne de commande ALOM.

▼ Pour afficher l'état du voyant Localisation

- Saisissez l'une des commandes suivantes :
 - En tant que superutilisateur, tapez :

```
# /usr/sbin/locator
```

- A l'interface de ligne de commande ALOM, tapez :

```
sc> showlocator
```

▼ Pour allumer le voyant Locator

- Saisissez l'une des commandes suivantes :
 - En tant que superutilisateur, tapez :

```
# /usr/sbin/locator -n
```

- A l'interface de ligne de commande ALOM, tapez :

```
sc> setlocator on
```

▼ Pour éteindre le voyant Locator

- Saisissez l'une des commandes suivantes :
 - En tant que superutilisateur, tapez :

```
# /usr/sbin/locator -f
```

- A l'interface de ligne de commande ALOM, tapez :

```
sc> setlocator off
```

Indicateurs d'état des alarmes

La carte d'alarme de contact sec comporte quatre voyants d'état pris en charge par ALOM. Ils sont placés verticalement sur le cache avant (FIGURE 1-2). Pour de plus amples informations sur les voyants d'alarme et les états d'alarmes du contact sec, consultez le TABLEAU 1-3. Pour de plus amples informations sur les voyants d'alarme, consultez le *Guide de l'utilisateur du logiciel Advanced Lights Out Manager Software de Sun pour le serveur Netra 240* (817-5002-11). Pour plus d'informations sur une API permettant de gérer les indicateurs d'alarme, reportez-vous à l'Annexe A.

TABLEAU 1-3 Indicateurs d'alarme et états de l'alarme du contact sec

Voyant et relais Étiquettes	Couleur du voyant	État de l'application ou du serveur	État ou action	État du voyant système	État du voyant d'alarme	Relais NC ^{IV} Etat	Relais NO ^V Etat	Commentaires	
Critique (Alarm0)	Rouge	État du serveur (Alimentation activée/désactivée et SE Solaris fonctionnel/non fonctionnel)	Aucune alimentation électrique	Éteint	Éteint	Fermé	Ouvert	État par défaut	
			Mise hors tension du système	Éteint	Éteint ⁱⁱⁱ	Fermé	Ouvert	Alimentation électrique raccordée	
			Mise sous tension du système ; SE Solaris pas entièrement chargé	Éteint	Éteint ⁱⁱⁱ	Fermé	Ouvert	État passager	
			SE Solaris chargé avec succès	Allumé	Éteint	Ouvert	Fermé	État de fonctionnement normal	
			Temporisation de la surveillance	Éteint	Allumé	Fermé	Ouvert	État passager, réinitialisation du SE Solaris	
			Fermeture du SE Solaris initiée par l'utilisateur ⁱ	Éteint	Éteint ⁱⁱⁱ	Fermé	Ouvert	État passager	
			Alimentation électrique perdue.	Éteint	Éteint	Fermé	Ouvert	État par défaut	
			Mise hors tension du système par l'utilisateur	Éteint	Éteint ⁱⁱⁱ	Fermé	Ouvert	État passager	
			État de l'application	L'utilisateur active l'alarme critique ⁱⁱ	—	Allumé	Fermé	Ouvert	Panne critique détectée
			L'utilisateur désactive l'alarme critique ⁱⁱ	—	Éteint	Ouvert	Fermé	Panne critique corrigée	

TABEAU 1-3 Indicateurs d'alarme et états de l'alarme du contact sec (*suite*)

Voyant et relais Étiquettes	Couleur du voyant	État de l'application ou du serveur	État ou action	État du voyant système	État du voyant d'alarme	Relais NC ^{iv} Etat	Relais NO ^v Etat	Commentaires
Majeure (Alarm1)	Rouge	État de l'application	L'utilisateur active l'alarme majeure ⁱⁱ	—	Allumé	Ouvert	Fermé	Panne majeure détectée
			L'utilisateur désactive l'alarme majeure ⁱⁱ	—	Éteint	Fermé	Ouvert	Panne majeure corrigée
Mineure (Alarm2)	Orange	État de l'application	L'utilisateur active l'alarme mineure ⁱⁱ	—	Allumé	Ouvert	Fermé	Panne mineure détectée
			L'utilisateur désactive l'alarme mineure ⁱⁱ	—	Éteint	Fermé	Ouvert	Panne mineure corrigée
Utilisateur (Alarm3)	Orange	État de l'application	L'utilisateur active l'alarme utilisateur ⁱⁱ	—	Allumé	Ouvert	Fermé	Panne utilisateur détectée
			L'utilisateur désactive l'alarme utilisateur ⁱⁱ	—	Éteint	Fermé	Ouvert	Panne utilisateur corrigée

i L'utilisateur peut fermer le système à l'aide de commandes telles que `init0` et `init6`. Cette opération ne comprend pas la mise hors tension du système.

ii Sur la base d'une détermination des pannes, l'utilisateur peut activer l'alarme à l'aide de l'API d'alerte de la plate-forme Solaris ou de l'ILC ALOM. Pour de plus amples informations sur l'API d'alerte, reportez-vous à l'Annexe A. Pour plus d'informations sur l'interface de ligne de commandes ALOM, reportez-vous au *Guide de l'utilisateur du logiciel Advanced Lights Out Manager Software de Sun pour le serveur Netra 240 (817-5002-11)*.

iii La mise en œuvre de cet état du voyant d'alarme peut changer.

iv « NC » signifie fermé normalement (« Normally Closed »). Cet état représente le mode par défaut des contacts de relais fermés normalement.

v « NO » signifie ouvert normalement (« Normally Open »). Cet état représente le mode par défaut des contacts de relais ouverts normalement.

Dans tous les cas où l'utilisateur définit une alarme, un message s'affiche sur la console. Par exemple, lorsque l'alarme critique est définie, le message suivant s'affiche sur la console :

```
SC Alert: CRITICAL ALARM is set
```

Remarquez que, dans certains cas, lorsque l'alarme critique est activée, le voyant associé ne s'allume pas. La mise en œuvre des alarmes sera modifiée dans les versions futures. (Voir la note en bas de page ⁱⁱⁱ du TABLEAU 1-3.)

Diagnostics d'auto-test à la mise sous tension

POST (Power-on self-test, auto-test à la mise sous tension) est un microprogramme qui vise à déterminer si une portion du système est défectueuse. POST vérifie les fonctions principales du système, notamment le(s) module(s) des processeurs, la carte mère, la mémoire et certains périphériques d'E/S intégrés. Il émet des messages qui peuvent aider à déterminer la nature d'une défaillance matérielle. Vous pouvez exécuter POST même s'il est impossible de démarrer le système.

POST détecte la plupart des erreurs système ; il est situé dans la mémoire OpenBoot PROM de la carte mère. Vous pouvez programmer le logiciel OpenBoot de manière à ce qu'il exécute POST au démarrage. Pour ce faire, il suffit de définir deux variables d'environnement : `diag-switch?` et `diag-level`. Ces deux variables sont enregistrées sur la carte de configuration du système.

POST s'exécute automatiquement dès la mise sous tension, ou à la suite d'une réinitialisation automatique du système, si les deux conditions suivantes sont respectées :

- `diag-switch?` a la valeur `true` (valeur par défaut : `false`)
- `diag-level` a la valeur `min`, `max` ou `menus` (valeur par défaut : `min`)
- `post-trigger` correspond à la classe de réinitialisation (valeur par défaut `power-on-reset`)

Si `diag-level` a la valeur `min` ou `max`, POST exécute un test abrégé ou étendu, respectivement.

Si `diag-level` est définie à `menus`, un menu de tous les tests exécutés à la mise sous tension apparaît.

Les diagnostics et les messages d'erreur POST sont affichés sur une console.

Contrôle des diagnostics POST

Pour contrôler les diagnostics POST (et d'autres aspects du processus de démarrage), vous devez définir les variables de configuration OpenBoot. Les modifications apportées aux variables de configuration OpenBoot prennent effet après le redémarrage du système seulement. Le TABLEAU 1-4 répertorie les variables les plus importantes et les plus utiles. Pour connaître les instructions liées à la modification des variables de configuration OpenBoot, reportez-vous à la section « Pour afficher et définir les variables de configuration OpenBoot » à la page 28.

TABLEAU 1-4 Variables de configuration OpenBoot

Variable de configuration OpenBoot	Description et mots-clés
auto-boot	Détermine si le système d'exploitation démarre automatiquement. La valeur par défaut est <code>true</code> . <ul style="list-style-type: none">• <code>true</code> : le système d'exploitation démarre automatiquement une fois que les tests du microprogramme sont terminés.• <code>false</code> : le système affiche l'invite <code>ok</code> jusqu'à ce que vous tapiez <code>boot</code>.
diag-level	Détermine le niveau ou le type de diagnostics exécutés. La valeur par défaut est <code>min</code> . <ul style="list-style-type: none">• <code>off</code> : aucun test.• <code>min</code> : seuls des tests de base sont exécutés.• <code>max</code> : vous pouvez exécuter des tests plus élaborés, en fonction du périphérique.• <code>menus</code> : les tests commandés par des menus aux niveaux des POST peuvent être exécutés sur une base individuelle.
diag-script	Détermine les périphériques testés par OpenBoot Diagnostics. La valeur par défaut est <code>none</code> . <ul style="list-style-type: none">• <code>none</code> : aucun périphérique n'est testé.• <code>normal</code> : les périphériques intégrés (basés sur le plan médian) dotés d'auto-tests sont testés.• <code>all</code> : tous les périphériques dotés d'auto-tests sont contrôlés.
diag-switch?	Active et désactive le mode de diagnostic. La valeur par défaut est <code>false</code> . <ul style="list-style-type: none">• <code>true</code> : mode diagnostic: des diagnostics POST et des tests OpenBoot Diagnostics peuvent être exécutés.• <code>false</code> : mode par défaut : N'exécutez pas de test POST ni OpenBoot Diagnostics.

TABLEAU 1-4 Variables de configuration OpenBoot (*suite*)

Variable de configuration OpenBoot	Description et mots-clés
<code>post-trigger</code> <code>obdiag-trigger</code>	<p>Ces deux variables spécifient la classe d'événement de réinitialisation qui déclenche les autotests à la mise sous tension (ou les tests OpenBoot Diagnostics) à exécuter. Ces variables acceptent les mots-clés uniques et les combinaisons des trois premiers mots-clés séparés par des espaces. Pour plus d'informations, reportez-vous au « Pour afficher et définir les variables de configuration OpenBoot » à la page 28.</p> <ul style="list-style-type: none">• <code>error-reset</code> : réinitialisation provoquée par certaines conditions d'erreur matérielle non récupérable. En général, une réinitialisation d'erreur se produit lorsqu'un problème matériel altère les données d'état du système. Exemples : réinitialisation de l'UC et du mécanisme de surveillance, erreurs fatales et événements de réinitialisation de certaines UC (par défaut).• <code>power-on-reset</code> : réinitialisation provoquée par une pression sur le bouton Marche/Veille (par défaut).• <code>user-reset</code> : réinitialisation lancée par l'utilisateur ou par le système d'exploitation.• <code>all-resets</code> : tous les types de réinitialisation système.• <code>none</code> : aucun auto-test à la mise sous tension (ni test OpenBoot Diagnostics) n'est exécuté.
<code>input-device</code>	<p>Sélectionne l'emplacement d'origine de l'entrée de console. La valeur par défaut est <code>ttya</code>.</p> <ul style="list-style-type: none">• <code>ttya</code> : du port intégré SERIAL MGT• <code>ttyb</code> : du port série intégré d'usage général (10101)• <code>keyboard</code> : clavier connecté faisant partie d'un terminal graphique.
<code>output-device</code>	<p>Sélectionne l'emplacement d'affichage du diagnostic et de la sortie de console. La valeur par défaut est <code>ttya</code>.</p> <ul style="list-style-type: none">• <code>ttya</code> : vers le port intégré SERIAL MGT• <code>ttyb</code> : vers le port série intégré d'usage général (10101)• <code>screen</code> : vers l'écran raccordé faisant partie d'un terminal graphique.ⁱ

ⁱ Les messages POST ne peuvent pas être affichés sur un terminal graphique. Ils sont envoyés vers `ttya` même lorsque `output-device` est réglé sur `screen`.

Remarque : Ces variables influencent les tests OpenBoot Diagnostics et les diagnostics POST.

Une fois les diagnostics POST exécutés, POST fournit un compte-rendu de l'état de chaque test exécuté sur le microprogramme OpenBoot. Le contrôle passe ensuite au code du microprogramme OpenBoot.

Si les diagnostics POST ne signalent pas de défaillance et que votre serveur ne démarre toujours pas, exécutez les tests OpenBoot Diagnostics.

▼ Pour lancer les diagnostics POST

1. Rendez-vous à l'invite `ok`.

2. Saisissez :

```
ok setenv diag-switch? true
```

3. Saisissez :

```
ok setenv diag-level value
```

Où la *valeur* est min, max ou menus selon la quantité des informations de diagnostic à afficher.

4. Saisissez :

```
ok reset-all
```

Le système exécute des diagnostics POST si `post-trigger` a la valeur `user-reset`. Des messages d'état et d'erreur s'affichent dans la fenêtre de la console. Si POST détecte une erreur, un message d'erreur décrit la défaillance.

5. Lorsque vous avez terminé d'exécuter POST, restaurez la valeur de `diag-switch?` à `false` en tapant :

```
ok setenv diag-switch? false
```

La réinitialisation de `diag-switch?` à `false` réduit le temps de démarrage.

Commandes OpenBoot

Les commandes OpenBoot sont saisies à partir de l'invite `ok`. Les commandes OpenBoot susceptibles de donner des informations de diagnostic utiles sont les suivantes :

- `probe-scsi` et `probe-scsi-all`
- `probe-ide`
- `show-devs`

Commandes `probe-scsi` et `probe-scsi-all`

Les commandes `probe-scsi` et `probe-scsi-all` permettent de diagnostiquer les problèmes liés aux périphériques SCSI.



Attention : Si vous avez utilisé la commande `halt` ou la séquence de touches `Stop-A` pour atteindre l'invite `ok`, l'exécution de la commande `probe-scsi` ou `probe-scsi-all` peut arrêter le système.

La commande `probe-scsi` communique avec tous les périphériques SCSI connectés aux contrôleurs embarqués SCSI. De plus, la commande `probe-scsi-all` permet d'accéder à des périphériques connectés à des adaptateurs hôtes installés dans des emplacements PCI.

Pour tous les périphériques SCSI connectés et actifs, les commandes `probe-scsi` et `probe-scsi-all` permettent d'afficher l'ID en boucle, l'adaptateur hôte, le numéro d'unité logique, le WWN (Word Wide Name) unique et une description du périphérique incluant le type et le fabricant.

Voici un exemple de sortie de la commande `probe-scsi`.

EXEMPLE DE CODE 1-1 Sortie de la commande `probe-scsi`

```
{1} ok probe-scsi
Target 0
  Unit 0   Disk      SEAGATE ST373307LSUN72G 0207
Target 1
  Unit 0   Disk      SEAGATE ST336607LSUN36G 0207
{1} ok
```

Voici un exemple de sortie de la commande `probe-scsi-all`.

EXEMPLE DE CODE 1-2 Sortie de la commande `probe-scsi-all`

```
{1} ok probe-scsi-all
/pci@1c,600000/scsi@2,1

/pci@1c,600000/scsi@2
Target 0
  Unit 0   Disk      SEAGATE ST373307LSUN72G 0207
Target 1
  Unit 0   Disk      SEAGATE ST336607LSUN36G 0207

{1} ok
```

Commande probe-ide

La commande `probe-ide` communique avec tous les périphériques IDE (Integrated Drive Electronics) connectés au bus IDE. Il s'agit du bus système interne de tous les périphériques de support (par exemple, le lecteur de DVD-ROM).



Attention : Si vous avez utilisé la commande `halt` ou la séquence de touches Stop-A pour atteindre l'invite `ok`, l'exécution de la commande `probe-ide` peut arrêter le système.

Voici un exemple de sortie de la commande `probe-ide`.

EXEMPLE DE CODE 1-3 Sortie de la commande `probe-ide`

```
{1} ok probe-ide
Device 0 ( Primary Master )
        Not Present

Device 1 ( Primary Slave )
        Not Present

Device 2 ( Secondary Master )
        Not Present

Device 3 ( Secondary Slave )
        Not Present

{1} ok
```

Commande show-devs

La commande `show-devs` répertorie les chemins de chaque périphérique matériel dans l'arborescence des périphériques basés sur microprogrammes. L'EXEMPLE DE CODE 1-4 illustre un exemple de sortie.

EXEMPLE DE CODE 1-4 Sortie de la commande `show-devs`

```
/pci@1d,700000
/pci@1c,600000
/pci@1e,600000
/pci@1f,700000
/memory-controller@1,0
/SUNW,UltraSPARC-IIIi@1,0
/memory-controller@0,0
```

EXEMPLE DE CODE 1-4 Sortie de la commande `show-devs` (*suite*)

```
/SUNW,UltraSPARC-IIi@0,0
/virtual-memory
/memory@m0,0
/aliases
/options
/openprom
/chosen
/packages
/pci@1d,700000/network@2,1
/pci@1d,700000/network@2
/pci@1c,600000/scsi@2,1
/pci@1c,600000/scsi@2
/pci@1c,600000/scsi@2,1/tape
/pci@1c,600000/scsi@2,1/disk
/pci@1c,600000/scsi@2/tape
/pci@1c,600000/scsi@2/disk
/pci@1e,600000/ide@d
/pci@1e,600000/usb@a
/pci@1e,600000/pmu@6
/pci@1e,600000/isa@7
/pci@1e,600000/ide@d/cdrom
/pci@1e,600000/ide@d/disk.....
```

▼ Pour lancer les commandes OpenBoot

1. **Arrêtez le système pour atteindre l'invite `ok`.**
Informez les utilisateurs avant de fermer le système.
2. **Saisissez la commande appropriée à l'invite de la console.**

OpenBoot Diagnostics

A l'instar des diagnostics POST, le code OpenBoot Diagnostics est intégré au microprogramme et réside dans la mémoire Boot PROM.

▼ Pour lancer OpenBoot Diagnostics

1. Tapez :

```
ok setenv diag-switch? true  
ok setenv auto-boot? false  
ok reset-all
```

2. Tapez :

```
ok obdiag
```

Cette commande affiche le menu OpenBoot Diagnostics.

```
ok obdiag
```

```
o b d i a g  
-----  
1 i2c@0,320          2 ide@d          3 network@2  
4 network@2,1       5 rtc@0,70       6 scsi@2  
7 scsi@2,1          8 serial@0,2e8   9 serial@0,3f8  
10 usb@a            11 usb@b         12 flashprom@2,0  
-----  
Commands: test test-all except help what printenvs setenv versions exit  
-----
```

Remarque : Si une carte PCI est installée dans le serveur, des tests supplémentaires apparaîtront dans le menu obdiag.

3. Saisissez :

```
obdiag> test n
```

Où *n* représente le numéro du test que vous souhaitez exécuter.

Un résumé des tests est disponible. À l'invite obdiag>, tapez :

```
obdiag> help
```

Contrôle des tests OpenBoot Diagnostics

La plupart des variables de configuration OpenBoot que vous utilisez pour contrôler POST (voir le TABLEAU 1-4) influencent également les tests OpenBoot Diagnostics.

- Utilisez la variable `diag-level` pour contrôler le niveau de test OpenBoot Diagnostics.
- Utilisez `test-args` pour personnaliser l'exécution des tests.

La variable `test-args` est configurée par défaut de manière à contenir une chaîne vierge. Vous pouvez modifier `test-args` à l'aide d'un ou plusieurs des mots-clés réservés présentés dans le TABLEAU 1-5.

TABLEAU 1-5 Mots clés associés à la variable de configuration OpenBoot `test-args`

Mot-clé	Description
<code>bist</code>	Invoque l'autotest intégré (BIST, Built-In Self-Test) sur les périphériques externes et autres.
<code>debug</code>	Affiche tous les messages de débogage.
<code>iopath</code>	Vérifie l'intégrité des interconnexions et des bus.
<code>loopback</code>	Vérifie le chemin de boucle externe du périphérique.
<code>media</code>	Vérifie l'accessibilité des supports de périphérique externe et autre.
<code>restore</code>	Tente de restaurer l'état d'origine du périphérique, en cas d'échec du test précédent.
<code>silent</code>	Affiche uniquement les erreurs, plutôt que l'état de chaque test.
<code>subtests</code>	Affiche le test principal et chaque sous-test appelé.
<code>verbose</code>	Affiche des messages d'état détaillés pour tous les tests.
<code>callers=<i>n</i></code>	Affiche la trace inverse de <i>N</i> appelants lorsqu'une erreur se produit. <code>callers=0</code> - affiche la trace inverse de tous les appelants avant l'erreur.
<code>errors=<i>n</i></code>	Poursuit l'exécution du test jusqu'à l'obtention de <i>N</i> erreurs. <code>errors=0</code> - affiche tous les comptes-rendus d'erreurs sans terminer le test.

Si vous souhaitez personnaliser le test OpenBoot Diagnostics, vous pouvez régler `test-args` sur une liste de mots-clés séparés par une virgule. Exemple :

```
ok setenv test-args debug,loopback,media
```

Commandes `test` et `test-all`

Vous pouvez également exécuter des tests OpenBoot Diagnostics directement à partir de l'invite `ok`. Pour cela, tapez la commande `test`, suivie du chemin matériel complet du périphérique (ou de l'ensemble de périphériques) à tester. Par exemple :

```
ok test /pci@x,y/SUNW,q1c@2
```

Pour personnaliser un test individuel, vous pouvez utiliser `test-args` comme suit :

```
ok test /usb@1,3:test-args={verbose,debug}
```

Seul le test en cours est affecté et la valeur de la variable de configuration OpenBoot `test-args` n'est pas modifiée.

Vous pouvez tester tous les périphériques de l'arborescence à l'aide de la commande `test-all` :

```
ok test-all
```

Si vous réglez un argument de chemin sur `test-all`, seul le périphérique indiqué et ses enfants sont testés. L'exemple suivant représente la commande permettant de tester le bus USB et tous les périphériques dotés d'auto-tests et connectés à ce bus :

```
ok test-all /pci@9,700000/usb@1,3
```

Messages d'erreur OpenBoot Diagnostics

Les résultats renvoyés par OpenBoot Diagnostics figurent dans un tableau contenant un bref résumé du problème, le périphérique matériel concerné, le sous-test qui a échoué ainsi que d'autres informations de diagnostic. L'EXEMPLE DE CODE 1-5 affiche un exemple de message d'erreur OpenBoot Diagnostics.

EXEMPLE DE CODE 1-5 Message d'erreur OpenBoot Diagnostics

```
Testing /pci@1e,600000/isa@7/flashprom@2,0

ERROR   : FLASHPROM CRC-32 is incorrect
SUMMARY : Obs=0x729f6392 Exp=0x3d6cdf53 XOR=0x4ff3bcc1 Addr=0xfeebbffc
DEVICE  : /pci@1e,600000/isa@7/flashprom@2,0
SUBTEST : selftest:crc-subtest
MACHINE : Netra 240
SERIAL#  : 52965531
DATE    : 03/05/2003 01:33:59 GMT
CONTROLS: diag-level=max test-args=

Error: /pci@1e,600000/isa@7/flashprom@2,0 selftest failed, return code = 1
Selftest at /pci@1e,600000/isa@7/flashprom@2,0 (errors=1) .....
échec
Pass:1 (of 1) Errors:1 (of 1) Tests Failed:1 Elapsed Time: 0:0:0:27
```

Outils de diagnostic du système d'exploitation

Lorsqu'un système réussit les tests OpenBoot Diagnostics, il tente de démarrer le système d'exploitation Solaris. Une fois que le serveur fonctionne en mode multiutilisateurs, vous avez accès aux outils de diagnostic logiciels et au logiciel SunVTS. Ces outils vous permettent de surveiller le serveur, de le vérifier et d'isoler ses erreurs.

Remarque : Si vous réglez la variable de configuration OpenBoot `auto-boot?` sur `false`, le système d'exploitation ne s'initialise *pas* après l'exécution des tests basés sur les microprogrammes.

Outre les outils mentionnés précédemment, vous pouvez consulter les fichiers journaux de messages d'erreur et messages système, ainsi que les commandes d'informations du logiciel Solaris.

Fichiers journaux de messages d'erreur et messages système

Des messages d'erreur et d'autres messages système sont enregistrés dans le fichier `/var/adm/messages`. Les messages sont journalisés dans ce fichier à partir de nombreuses sources, y compris le système d'exploitation, le sous-système de surveillance de l'environnement et diverses applications logicielles.

Commandes d'information du système logiciel Solaris

Les commandes d'information du système logiciel Solaris affichent les données que vous pouvez utiliser lors de l'évaluation d'un serveur Netra 240 :

- `prtconf`
- `prtdiag`
- `prtfu`
- `psrinfo`
- `showrev`

Cette section décrit les informations communiquées par ces commandes. Pour obtenir plus d'informations sur l'utilisation de ces commandes, reportez-vous à la page de manuel appropriée.

Commande `prtconf`

La commande `prtconf` affiche l'arborescence des périphériques du logiciel Solaris. Cette arborescence comprend tous les périphériques sondés par le microprogramme OpenBoot, ainsi que des périphériques supplémentaires, tels que les disques individuels, que seul le logiciel du système d'exploitation reconnaît. La sortie de `prtconf` comprend également la mémoire totale du système. L'EXEMPLE DE CODE 1-6 illustre un extrait de la sortie `prtconf`.

EXEMPLE DE CODE 1-6 Sortie de la commande prtconf

```
# prtconf

System Configuration: Sun Microsystems sun4u
Memory size: 5120 Megabytes
System Peripherals (Software Nodes):

SUNW,Netra-240
  packages (driver not attached)
    SUNW,builtin-drivers (driver not attached)
    deblocker (driver not attached)
    disk-label (driver not attached)
    terminal-emulator (driver not attached)
    dropins (driver not attached)
    kbd-translator (driver not attached)
    obp-tftp (driver not attached)
    SUNW,i2c-ram-device (driver not attached)
    SUNW,fru-device (driver not attached)
    ufs-file-system (driver not attached)
  chosen (driver not attached)
  openprom (driver not attached)
    client-services (driver not attached)
  options, instance #0
  aliases (driver not attached)
  memory (driver not attached)
  virtual-memory (driver not attached)
  SUNW,UltraSPARC-IIIi (driver not attached)
  memory-controller, instance #0
  SUNW,UltraSPARC-IIIi (driver not attached)
  memory-controller, instance #1
  pci, instance #0.....
```

L'option `-p` de la commande `prtconf` fournit une sortie analogue à celle de la commande `OpenBoot show-devs`. Cette sortie répertorie uniquement les périphériques compilés par le microprogramme système.

Commande prtdiag

La commande `prtdiag` affiche un tableau d'informations de diagnostic récapitulant l'état des composants du système. Le format d'affichage utilisé par cette commande `prtdiag` peut varier en fonction de la version du système d'exploitation Solaris exécutée sur votre système. Voici un extrait de l'une des sorties générées par la commande `prtdiag` sur un serveur Netra 240 opérationnel exécutant le logiciel Solaris.

EXEMPLE DE CODE 1-7 Sortie de la commande prtdiag

```
# prtdiag
System Configuration: Sun Microsystems sun4u Netra 240
System clock frequency: 160 MHz
Memory size: 2 Go
===== CPUs =====
      CPU  Freq      E$      CPU      CPU      Temperature      Fan
      CPU  Freq      Size      Impl.  Mask      Die      Ambient      Speed  Unit
-----
      MB/P0 1280 MHz 1MB      US-IIIi  2,3      -      -
      MB/P1 1280 MHz 1MB      US-IIIi  2,3      -      -
===== IO Devices =====
      Bus  Freq
Brd  Type  MHz  Slot      Name      Model
-----
0    pci   66      2  network-pci14e4,1648.108e.16+
0    pci   66      2  network-pci14e4,1648.108e.16+
0    pci   66      2  scsi-pci1000,21.1000.1000.1 +
0    pci   66      2  scsi-pci1000,21.1000.1000.1 +
0    pci   66      2  network-pci14e4,1648.108e.16+
0    pci   66      2  network-pci14e4,1648.108e.16+
0    pci   33      7  isa/serial-su16550 (serial)
0    pci   33      7  isa/serial-su16550 (serial)
0    pci   33      7  isa/rmc-comm-rmc_comm (seria+
0    pci   33      13  ide-pci10b9,5229.c4 (ide)
===== Memory Configuration =====
Segment Table:
-----
Base Address      Size      Interleave Factor  Contains
-----
0x0               1GB      1                  GroupID 0
0x1000000000      1GB      1                  GroupID 0
```

EXEMPLE DE CODE 1-7 Sortie de la commande `prtdiag` (suite)

```

Memory Module Groups:
-----
ControllerID  GroupID  Labels
-----
0              0        MB/P0/B0/D0,MB/P0/B0/D1
Memory Module Groups:
-----
ControllerID  GroupID  Labels
-----
1              0        MB/P1/B0/D0,MB/P1/B0/D1

```

Outre les informations de l'EXEMPLE DE CODE 1-7, la commande `prtdiag` combinée à l'option `verbose (-v)` fournit également un compte-rendu sur l'état du panneau avant, l'état du disque, l'état du ventilateur, les blocs d'alimentation, les révisions matérielles et les températures du système (voir l'EXEMPLE DE CODE 1-8).

EXEMPLE DE CODE 1-8 Sortie de l'option `Verbose prtdiag`

```

-----
Location      Sensor          Temperature  Lo LoWarn HiWarn  Hi Status
-----
MB            T_ENC           22C         -7C  -5C    55C   58C okay
MB/P0        T_CORE          57C         -    -    110C  115C okay
MB/P1        T_CORE          54C         -    -    110C  115C okay
PS0          FF_OT           -           -    -    -     -   okay
PS1          FF_OT           -           -    -    -     -   okay

```

En cas de surchauffe, `prtdiag` signale une erreur dans la colonne `Status` (EXEMPLE DE CODE 1-9).

EXEMPLE DE CODE 1-9 Sortie de l'indication de surchauffe `prtdiag`

```

-----
Location      Sensor          Temperature  Lo LoWarn HiWarn  Hi Status
-----
MB            T_ENC           22C         -7C  -5C    55C   58C okay
MB/P0        T_CORE          118C        -    -    110C  115C failed
MB/P1        T_CORE          112C        -    -    110C  115C warning
PS0          FF_OT           -           -    -    -     -   okay
PS1          FF_OT           -           -    -    -     -   okay

```

De la même façon, en cas de panne d'un composant spécifique, `prtdiag` signale une erreur dans la colonne `Status` appropriée (EXEMPLE DE CODE 1-10).

EXEMPLE DE CODE 1-10 Sortie de l'indication d'erreur prtdiag

```
Fan Speeds:
-----
Location      Sensor      Status      Speed
-----
MB/P0/F0      RS          failed      0 rpm
MB/P0/F1      RS          okay        3994 rpm
F2            RS          okay        2896 rpm
PS0           FF_FAN      okay
F3            RS          okay        2576 rpm
PS1           FF_FAN      okay
-----
```

Commande prtfru

Le serveur Netra 240 gère une liste hiérarchique de toutes les unités interchangeables sur site (FRU) du système, ainsi que des informations spécifiques sur d'entre elles.

La commande `prtfru` peut afficher cette liste hiérarchique, ainsi que des données contenues dans les périphériques SEEPRAM (serial electrically-erasable programmable read-only memory) situés sur de nombreuses unités interchangeables sur site. L'EXEMPLE DE CODE 1-11 illustre un extrait de liste hiérarchique de FRU produit par la commande `prtfru` avec l'option `-l`.

EXEMPLE DE CODE 1-11 Sortie de la commande prtfru -l

```
# prtfru -l
/frutree
/frutree/chassis (fru)
/frutree/chassis/MB?Label=MB
/frutree/chassis/MB?Label=MB/system-board (container)
/frutree/chassis/MB?Label=MB/system-board/SC?Label=SC
/frutree/chassis/MB?Label=MB/system-board/SC?Label=SC/sc (fru)
/frutree/chassis/MB?Label=MB/system-board/BAT?Label=BAT
/frutree/chassis/MB?Label=MB/system-board/BAT?Label=BAT/battery (fru)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu (fru)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F0?Label=F0
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F0?Label=F0/fan-unit
(fru)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F1?Label=F1
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F1?Label=F1/fan-unit
(fru).....
```

L'EXEMPLE DE CODE 1-12 représente un extrait des données SEEPROM générées par la commande `prtfriu` à l'aide de l'option `-c`. Cette sortie n'affiche que les conteneurs et leurs données et n'imprime pas la hiérarchie de l'arborescence des unités interchangeables sur site.

EXEMPLE DE CODE 1-12 Sortie de la commande `prtfriu -c`

```
# prtfriu -c
/frutree/chassis/MB?Label=MB/system-board (container)
  SEGMENT: SD
    /ManR
    /ManR/UNIX_Timestamp32: Mon Dec  2 19:47:38 PST 2002
    /ManR/Fru_Description: FRUID, INSTR, M'BD, 2X1.28GHZ, CPU
    /ManR/Manufacture_Loc: Hsinchu, Taiwan
    /ManR/Sun_Part_No: 3753120
    /ManR/Sun_Serial_No: 000615
    /ManR/Vendor_Name: Mitac International
    /ManR/Initial_HW_Dash_Level: 02
    /ManR/Initial_HW_Rev_Level: 0E
    /ManR/Fru_Shortname: MOTHERBOARD
    /SpecPartNo: 885-0076-11
/frutree/chassis/MB?Label=MB/system-board/P0?Label=
P0/cpu/B0?Label=B0/bank/D0?Label=D0/mem-module (container)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=
P0/cpu/B0?Label=B0/bank/D1?Label=D1/mem-module (container).....
```

Les données affichées par la commande `prtfriu` varient en fonction du type d'unité interchangeable sur site. En général, elles comprennent :

- la description de la FRU ;
- le nom et l'adresse du fabricant ;
- le numéro de référence et numéro de série ;
- les niveaux de révision matérielle.

Commande `psrinfo`

La commande `psrinfo` affiche la date et l'heure de mise sous tension de chaque UC. Grâce à l'option `verbose (-v)`, elle affiche également des informations supplémentaires sur les UC, y compris la fréquence d'horloge. L'EXEMPLE DE CODE 1-13 affiche une sortie de la commande `psrinfo` avec l'option `-v`.

EXEMPLE DE CODE 1-13 Sortie de la commande psrinfo -v

```
# psrinfo -v
Status of processor 0 as of: 07/28/2003 14:43:29
  Processor has been on-line since 07/21/2003 18:43:37.
  The sparcv9 processor operates at 1280 MHz,
    and has a sparcv9 floating point processor.
Status of processor 1 as of: 07/28/2003 14:43:29
  Processor has been on-line since 07/21/2003 18:43:36.
  The sparcv9 processor operates at 1280 MHz,
    and has a sparcv9 floating point processor
```

Commande showrev

La commande `showrev` affiche des informations de révision concernant les composants matériels et logiciels courants. L'EXEMPLE DE CODE 1-14 affiche une sortie de la commande `showrev`.

EXEMPLE DE CODE 1-14 Sortie de la commande showrev

```
# showrev
Hostname: vsp78-36
Hostid: 8328c87b
Release: 5.8
Kernel architecture: sun4u
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain: vsplab.SFBay.Sun.COM
Kernel version: SunOS 5.8 Generic 108528-18 November 2002
```

Lorsqu'elle est combinée à l'option `-p`, la commande `showrev` affiche les patches installés. L'EXEMPLE DE CODE 1-15 représente un exemple de sortie partielle de la commande `showrev` avec l'option `-p`.

EXEMPLE DE CODE 1-15 Sortie de la commande showrev -p

```
Patch: 109729-01 Obsolete: Requires: Incompatibles: Packages: SUNWcsu
Patch: 109783-01 Obsolete: Requires: Incompatibles: Packages: SUNWcsu
Patch: 109807-01 Obsolete: Requires: Incompatibles: Packages: SUNWcsu
Patch: 109809-01 Obsolete: Requires: Incompatibles: Packages: SUNWcsu
Patch: 110905-01 Obsolete: Requires: Incompatibles: Packages: SUNWcsu
Patch: 110910-01 Obsolete: Requires: Incompatibles: Packages: SUNWcsu
Patch: 110914-01 Obsolete: Requires: Incompatibles: Packages: SUNWcsu
Patch: 108964-04 Obsolete: Requires: Incompatibles: Packages: SUNWcsr
```

▼ Pour exécuter les commandes d'informations système de la plate-forme Solaris

- À l'invite de commande, entrez la commande du type d'informations système à afficher.

Pour plus d'informations, reportez-vous à la section « Commandes d'information du système logiciel Solaris » à la page 19. Voir le TABLEAU 1-6 pour obtenir un résumé des commandes.

TABLEAU 1-6 Commandes d'affichage des informations de la plate-forme Solaris

Commande	Informations affichées	Commande à saisir	Remarques
prtconf	Informations de configuration système	/usr/sbin/prtconf	—
prtdiag	Diagnostic et informations de configuration	/usr/platform/sun4u/sbin/prtdiag	Pour plus d'informations, utilisez l'option -v.
prtfru	Contenus de la hiérarchie des FRU et de la mémoire SEEPROM	/usr/sbin/prtfru	Utilisez l'option -l pour afficher la hiérarchie. Utilisez l'option -c pour afficher les données SEEPROM.
psrinfo	Date et heure d'apparition en ligne de chaque processeur ; vitesse d'horloge des processeurs	/usr/sbin/psrinfo	Utilisez l'option -v pour obtenir la fréquence et les autres données.
showrev	Informations de révision matérielle et logicielle	/usr/bin/showrev	Utilisez l'option -p pour afficher les patches logiciels.

Résultats de tests de diagnostic récents

Les résumés des résultats des derniers tests POST et OpenBoot Diagnostics sont enregistrés d'un cycle d'alimentation à l'autre.

▼ Pour afficher les résultats des tests récents

1. **Rendez-vous à l'invite** `ok`.
2. **Suivez l'une des procédures suivantes :**
 - Pour afficher un résumé des derniers résultats POST, saisissez :

```
ok show-post-results
```

- Pour afficher un résumé des derniers résultats OpenBoot Diagnostics, saisissez :

```
ok show-obdiag-results
```

Cette commande produit une liste des composants matériels du système et indique les composants qui ont réussi et échoué aux tests POST ou OpenBoot Diagnostics.

Variables de configuration OpenBoot

Les commutateurs et variables de configuration de diagnostic conservés dans IDPROM déterminent le mode et le moment d'exécution des tests POST et OpenBoot Diagnostics. Cette section explique comment accéder aux variables de configuration OpenBoot et les modifier. Pour obtenir la liste des variables de configuration OpenBoot importantes, reportez-vous au TABLEAU 1-4.

Les modifications apportées aux variables de configuration OpenBoot prennent généralement effet après le redémarrage.

▼ Pour afficher et définir les variables de configuration OpenBoot

- Arrêtez le serveur pour afficher l'invite `ok`.
 - Pour afficher les valeurs courantes des variables de configuration OpenBoot, utilisez la commande `printenv`.
L'exemple suivant présente un bref aperçu des résultats de cette commande :

```
ok printenv
Variable Name          Value          Default Value
diag-level             min           min
diag-switch?          false         false
```

- Pour définir ou modifier la valeur d'une variable de configuration, utilisez la commande `setenv` :

```
ok setenv diag-level max
diag-level =          max
```

- Pour permettre aux variables de configuration OpenBoot d'accepter plusieurs mots clés, séparez les mots clés par des espaces :

Utilisation des commandes `watch-net` et `watch-net-all` pour vérifier les connexions réseau

Le test de diagnostic `watch-net` surveille les paquets Ethernet sur l'interface réseau principale. Le test de diagnostic `watch-net-all` surveille les paquets Ethernet sur l'interface réseau principale et sur les interfaces réseau supplémentaires connectées à la carte système. Les bons paquets reçus par le système sont indiqués par un point (.). Les erreurs, notamment celles de cadrage et les erreurs CRC (Cyclic Redundancy Check), sont indiquées par un X et associées à une description.

- **Lancez le test de diagnostic watch-net en saisissant la commande watch-net à l'invite ok (EXEMPLE DE CODE 1-16).**

EXEMPLE DE CODE 1-16 Message de sortie de diagnostic watch-net

```
{0} ok watch-net
Internal loopback test -- succeeded.
Link is -- up
Looking for Ethernet Packets.
`.` is a Good Packet. `X' is a Bad Packet.
Type any key to stop.....
```

- **Pour le test de diagnostic watch-net-all, tapez watch-net-all à l'invite ok (EXEMPLE DE CODE 1-17).**

EXEMPLE DE CODE 1-17 Message de sortie de diagnostic watch-net-all

```
{0} ok watch-net-all
/pci@1f,0/pci@1,1/network@c,1
Internal loopback test -- succeeded.
Link is -- up
Looking for Ethernet Packets.
`.` is a Good Packet. `X' is a Bad Packet.
Type any key to stop.
```

Reprise automatique du système

Remarque : La fonction Automatic System Recovery (ASR) est différente de celle de redémarrage automatique du serveur, également prise en charge par le serveur Netra 240. Pour plus d'informations sur le redémarrage automatique du serveur, reportez-vous au Chapitre 3.

La fonction Automatic System Recovery (ASR) consiste en des autotests et une capacité de configuration automatique visant à détecter les composants matériels défectueux et à les déconfigurer. Ainsi, le serveur est capable de reprendre son fonctionnement après la survenue de certaines erreurs ou défaillances matérielles non fatales.

Si un composant est surveillé par ASR et que le serveur est capable de fonctionner sans lui, le serveur redémarrera automatiquement si ce composant subissait une erreur ou une défaillance. Ainsi, il est impossible pour un composant matériel défectueux d'immobiliser le système ou de le mettre en panne continuellement.

Si une erreur est détectée au cours de la séquence de mise sous tension, le composant défectueux est désactivé. Si le système est encore capable de fonctionner, la séquence de démarrage se poursuit.

Pour prendre en charge cette fonction d'initialisation en mode dégradé, les microprogrammes OpenBoot utilisent l'interface cliente 1275 (via l'arborescence des périphériques) pour marquer un périphérique comme *défaillant* ou *désactivé*, en créant une propriété d'état appropriée, dans le nœud correspondant de l'arborescence des périphériques. Le système d'exploitation Solaris n'activera pas le pilote d'un sous-système ainsi marqué.

Tant qu'un composant défectueux est électriquement inactif (par exemple, il ne provoque pas d'erreurs de bus aléatoires ni de bruit de signal), le système redémarrera automatiquement et reprendra son fonctionnement jusqu'à l'arrivée d'un technicien.

Une fois un périphérique défaillant ou désactivé remplacé par un nouveau, le microprogramme OpenBoot modifie automatiquement l'état du périphérique au redémarrage.

Remarque : ASR n'est pas active tant que vous ne l'activez pas (voir « Pour activer ASR » à la page 32).

Options Auto-Boot

Le paramètre `auto-boot?` définit si le microprogramme démarre automatiquement le système d'exploitation après chaque réinitialisation. La valeur par défaut est `true`.

Le paramètre `auto-boot-on-error?` définit si le système tentera un démarrage en mode dégradé après la détection d'un sous-système défectueux. Les paramètres `auto-boot?` et `auto-boot-on-error?` doivent être définis à `true` pour permettre un démarrage automatique en mode dégradé.

- Pour définir les commutateurs, saisissez :

```
ok setenv auto-boot? true  
ok setenv auto-boot-on-error? true
```

Remarque : Le paramètre par défaut pour `auto-boot-on-error?` est `false`. Par conséquent, le système n'essaie pas d'effectuer une initialisation en mode dégradé si vous n'affectez pas la valeur `true` à ce paramètre. En outre, le système ne tentera pas un démarrage en mode dégradé à la suite d'une erreur irréversible fatale, même si le démarrage en mode dégradé est activé. Pour obtenir des exemples d'erreurs bloquantes non réparables, reportez-vous à la section « Récapitulatif sur le traitement des erreurs » à la page 31.

Récapitulatif sur le traitement des erreurs

Les mesures correctives entreprises au cours de la séquence de mise sous tension se divisent en trois cas de figure :

- Lorsque POST ou les diagnostics OpenBoot ne détectent aucune erreur, le système essaie de s'initialiser si `auto-boot?` est `true`.
- Si des erreurs non fatales uniquement sont détectées par POST ou OpenBoot Diagnostics, le système tente de démarrer si `auto-boot?` est définie à `true` et que `auto-boot-on-error?` est définie à `true`.

Remarque : Si POST ou les diagnostics OpenBoot détectent une erreur non bloquante associée au périphérique d'initialisation normal, les microprogrammes OpenBoot désactivent automatiquement le périphérique défaillant et essaient d'utiliser le prochain périphérique d'initialisation spécifié par la variable de configuration `boot-device`.

- Lorsqu'une erreur bloquante est détectée par POST ou les diagnostics OpenBoot, le système ne se réinitialise pas, quelles que soient les valeurs des paramètres `auto-boot?` ou `auto-boot-on-error?` Les erreurs fatales non récupérables incluent les erreurs suivantes :
 - Défaillance de toutes les UC
 - Défaillance de tous les bancs de mémoire logique
 - Défaillance CRC (Cyclic Redundancy Check) de la mémoire Flash RAM
 - Défaillance des données de configuration PROM d'une FRU critique
 - Défaillance ASIC (Application-Specific Integrated Circuit) critique

Scénarios de réinitialisation

Trois variables de configuration OpenBoot, `diag-switch?`, `obdiag-trigger` et `post-trigger`, permettent de déterminer comment le système exécute les diagnostics intégrés aux microprogrammes, suite aux événements de réinitialisation du système.

Le protocole de réinitialisation système standard ignore POST et OpenBoot Diagnostics, sauf si `diag-switch?` a la valeur `true`. Le paramètre de cette variable par défaut est `false`. Etant donné que ASR dépend des diagnostics de microprogramme pour détecter les périphériques défectueux, `diag-switch?` doit avoir la valeur `true` pour que ASR fonctionne. Pour plus d'instructions, reportez-vous à la section « Pour activer ASR » à la page 32.

Pour contrôler quels événements de réinitialisation, le cas échéant, déclenchent automatiquement les diagnostics de microprogramme, utilisez `obdiag-trigger` et `post-trigger`. Pour obtenir des instructions détaillées sur ces variables et leur utilisation, reportez-vous aux sections « Contrôle des diagnostics POST » à la page 9 et « Contrôle des tests OpenBoot Diagnostics » à la page 16.

▼ Pour activer ASR

1. A l'invite `ok` du système, tapez la commande suivante :

```
ok setenv diag-switch? true
ok setenv auto-boot? true
ok setenv auto-boot-on-error? true
```

2. Affectez à la variable `obdiag-trigger` la valeur `power-on-reset`, `error-reset` ou `user-reset`.

Par exemple, saisissez :

```
ok setenv obdiag-trigger user-reset
```

3. Saisissez :

```
ok reset-all
```

Le système stocke en permanence les modifications apportées aux paramètres et est automatiquement initialisé si la variable OpenBoot `auto-boot?` est réglée sur `true` (valeur par défaut).

Remarque : Pour stocker les changements de paramètre, vous pouvez également lancer le cycle d'alimentation du système à l'aide du bouton Marche/Veille du panneau avant.

▼ Pour désactiver ASR

1. A l'invite `ok` du système, tapez la commande suivante :

```
ok setenv diag-switch? false
```

2. Saisissez :

```
ok reset-all
```

Le système stocke définitivement le changement de paramètre.

Remarque : Pour stocker les changements de paramètre, vous pouvez également lancer le cycle d'alimentation du système à l'aide du bouton Marche/Veille du panneau avant.

Logiciel SunVTS

Ce chapitre décrit le logiciel SunVTS. Ce chapitre couvre les sujets suivants :

- « Présentation du logiciel SunVTS » à la page 35
- « Tests SunVTS » à la page 36
- « Logiciel SunVTS et sécurité » à la page 37
- « Installation du logiciel SunVTS » à la page 38
- « Consultation de la documentation SunVTS » à la page 39

Présentation du logiciel SunVTS

Le logiciel SunVTS 5.1 Patch Set 5 (PS5) et les prochaines versions compatibles sont pris en charge sur le serveur Netra 240.

Le logiciel SunVTS (Sun Validation and Test Suite) est un outil de diagnostic en ligne qui permet de vérifier la configuration et la fonctionnalité des contrôleurs matériels, des périphériques et des plates-formes. Il fonctionne sur le système d'exploitation Solaris et comporte les interfaces suivantes :

- Interface de ligne de commande (ILC)
- interface série (tty)

La suite logicielle SunVTS procède à des tests sous contrainte du système et des périphériques. Il est possible d'afficher et de contrôler une session SunVTS sur un réseau. A l'aide d'un système à distance, vous pouvez suivre l'évolution de la session de test, modifier les options de test et contrôler toutes les caractéristiques de test d'un autre système du réseau.

Le logiciel SunVTS peut être exécuté en trois modes :

- Le mode *Connexion* vérifie la présence de contrôleurs périphériques. Cette vérification ne prend généralement que quelques minutes et constitue une bonne « vérification sanitaire » des connexions système.
- Le mode *Fonctionnel* ne vérifie que les sous-systèmes spécifiques sélectionnés. Il s'agit du mode par défaut.
- Le mode *Auto Config* (Configuration automatique) détecte automatiquement tous les sous-systèmes et les vérifie de l'une des deux façons suivantes :
 - *Test Confidence* (test de confiance) : teste une fois tous les sous-systèmes, puis s'arrête. Pour les configurations système classiques, cette opération dure près de deux heures.
 - *Test Comprehensive* (test complet) : teste plusieurs fois tous les sous-systèmes pendant un maximum de 24 heures.

Le logiciel SunVTS pouvant exécuter plusieurs tests en parallèle et consommant de nombreuses ressources système, vous devez l'utiliser avec prudence sur un système de production. Si vous effectuez des tests intenses en utilisant le mode « Comprehensive » du logiciel SunVTS, veillez à n'exécuter aucune autre application en parallèle sur le système à ce moment.

Un serveur doit exécuter le système d'exploitation Solaris pour que le logiciel SunVTS puisse le tester. Étant donné que les progiciels SunVTS sont en option, ils se peut qu'ils ne soient pas installés sur votre système. Pour plus d'instructions, reportez-vous à la section « Pour vérifier l'installation du logiciel SunVTS » à la page 38.

Tests SunVTS

Le logiciel SunVTS vous permet d'afficher et de contrôler une session de test sur un serveur connecté à distance. Le TABLEAU 2-1 répertorie certains des tests disponibles.

TABLEAU 2-1 Tests du logiciel SunVTS

Test du logiciel SunVTS	Description
cputest	Teste l'UC.
disktest	Teste les disques durs locaux.
dvdtest	Teste le lecteur de DVD-ROM.
n240atest	Teste la carte d'alarme afin de vérifier les relais d'alarme, les voyants et l'ID des unités interchangeables sur site.
fptest	Teste l'unité à virgule flottante.

TABLEAU 2-1 Tests du logiciel SunVTS (suite)

Test du logiciel SunVTS	Description
nettest	Teste le matériel Ethernet sur la carte système et le matériel réseau sur les cartes PCI en option.
netlptest	Effectue un essai en boucle afin de vérifier que l'adaptateur Ethernet peut envoyer et recevoir des paquets.
pmem	Teste la mémoire physique (lecture seule).
sutest	Teste les ports série intégrés du serveur.
vmem	Teste la mémoire virtuelle (partition d'échange et mémoire physique associées).
env6test	Teste les périphériques d'environnement.
ssptest	Teste les périphériques matériels ALOM.
i2c2test	Teste le bon fonctionnement des périphériques I ² C.

Logiciel SunVTS et sécurité

Lors de l'installation logicielle de SunVTS, vous devez choisir entre le mode de sécurité de base ou Sun Enterprise Authentication Mechanism™ (SEAM). En mode Basic, un fichier de sécurité local installé dans le répertoire d'installation du logiciel SunVTS limite le nombre d'utilisateurs, de groupes et d'hôtes autorisés à utiliser le logiciel SunVTS. La sécurité SEAM est basée sur le protocole d'authentification réseau standard Kerberos et garantit une authentification utilisateur sécurisée, l'intégrité des données et la confidentialité des transactions sur les réseaux.

Si votre site utilise le mode de sécurité SEAM, vous devez installer le logiciel client et serveur SEAM dans votre environnement en réseau, puis le configurer correctement dans les logiciels Solaris et SunVTS. Si votre site n'utilise pas le mode SEAM, ne sélectionnez pas l'option SEAM lors de l'installation logicielle de SunVTS.

Si vous activez le mauvais mode de sécurité au cours de l'installation, ou si vous configurez incorrectement le mode de sécurité sélectionné, il peut vous être impossible d'exécuter les tests SunVTS. Pour plus d'informations, reportez-vous au document intitulé *SunVTS User's Guide* et aux instructions fournies avec le logiciel SEAM.

▼ Pour vérifier l'installation du logiciel SunVTS

- Saisissez :

```
# pkginfo -l SUNWvts
```

- Si le logiciel SunVTS est chargé, des informations sur les modules s'affichent.
- Si SunVTS n'est pas chargé, le message d'erreur suivant apparaît :

```
ERROR : information for "SUNWvts" was not found
```

Installation du logiciel SunVTS

Par défaut, le logiciel SunVTS n'est pas installé sur les serveurs Netra 240. Il est toutefois disponible sur le CD Solaris OS Supplement et vous pouvez télécharger les toutes dernières versions depuis le site Web suivant :

<http://www.sun.com/oem/products/vts/>

Remarque : Le logiciel SunVTS 5.1 Patch Set 5 (PS5) et les prochaines versions compatibles sont pris en charge sur le serveur Netra 240.

Pour en savoir plus sur l'utilisation du logiciel SunVTS, consultez la documentation SunVTS correspondant à la version de Solaris que vous exécutez. Vous trouverez également des informations complémentaires sur le logiciel SunVTS, ainsi que des instructions d'installation, sur le site Web ci-dessus.

Consultation de la documentation SunVTS

Les documents SunVTS sont compris sur le CD supplémentaire du logiciel, fourni avec chaque kit de supports Solaris. Ils sont également accessibles à l'adresse : <http://docs.sun.com>.

Pour plus d'informations, vous pouvez également consulter les documents SunVTS suivants :

- *SunVTS User's Guide* décrit la procédure d'installation, de configuration et d'exécution du logiciel de diagnostic SunVTS.
- *SunVTS Quick Reference Card* donne un aperçu de l'utilisation de l'interface SunVTS.
- *SunVTS Test Reference Manual* fournit des détails sur chaque test SunVTS.

Advanced Lights Out Manager

Ce chapitre donne un aperçu du logiciel Advanced Lights-Out Manager (ALOM) de Sun™. Ce chapitre traite les sujets suivants :

- « Présentation du logiciel Advanced Lights Out Manager » à la page 41
- « Ports ALOM » à la page 42
- « Définition du mot de passe admin » à la page 43
- « Fonctions ALOM de base » à la page 43
- « Redémarrage automatique du serveur » à la page 45
- « Surveillance et contrôle de l'environnement » à la page 46

Présentation du logiciel Advanced Lights Out Manager

Le serveur Netra 240 livré est équipé en usine du logiciel Sun Advanced Lights Out Manager. La console système est dirigée vers ALOM par défaut et est configurée afin d'afficher les informations sur la console serveur au démarrage.

ALOM vous permet de surveiller votre serveur grâce à une connexion série (à l'aide du port SERIAL MGT) ou une connexion Ethernet (à l'aide du port NET MGT). Pour de plus amples informations sur la configuration de la connexion Ethernet ALOM, reportez-vous au *Guide de l'utilisateur du logiciel Advanced Lights Out Manager de Sun pour le serveur Netra 240 (817-5002-11)*.

Remarque : Le port série ALOM, appelé SERIAL MGT, est destiné uniquement à la gestion du serveur. Si vous avez besoin d'un port série à usage général, utilisez le port 10101.

ALOM peut être configuré pour l'envoi électronique de notifications de défaillances matérielles et d'autres événements liés au serveur ou à ALOM.

La circuiterie ALOM utilise l'alimentation de veille du serveur, ce qui signifie que :

- ALOM est actif dès que le serveur est connecté à une source d'alimentation et jusqu'à ce que le câble d'alimentation soit débranché ;
- le microprogramme et le logiciel ALOM continuent de fonctionner quand le système d'exploitation du serveur ferme la session.

Le TABLEAU 3-1 répertorie les composants surveillés par le logiciel ALOM et les informations fournies pour chacun par le logiciel.

TABLEAU 3-1 Composants surveillés par ALOM

Composant	Informations fournies
Disques durs	Présence et état
Ventilateurs du système et des processeurs	Vitesse et état
Processeurs	Présence, température et avertissements thermiques ou défaillances
Unités d'alimentation	Présence et état
Température système	Température ambiante et avertissements thermiques ou défaillances
Panneau avant du serveur	Position du commutateur rotatif et état des indicateurs
Tension	État et seuils
Disjoncteurs SCSI et USB	État
Alarmes avec relais à contacts secs	État

Ports ALOM

Le port de gestion par défaut est SERIAL MGT. Ce port utilise un connecteur RJ-45 et est destiné *uniquement* à la gestion du serveur ; il ne prend en charge que les connexions ASCII vers une console externe. Utilisez ce port lors de la première utilisation du serveur.

Un autre port série (10101) est destiné au transfert de données série d'usage général. Ce port utilise un connecteur DB-9. Pour plus d'informations sur les brochages, reportez-vous au *Guide d'installation du serveur Netra 240* (817-4994-11).

En outre, le serveur possède une interface de domaine de gestion Ethernet 10BASE-T : NET MGT. Pour utiliser ce port, la configuration ALOM est obligatoire. Pour plus d'informations, reportez-vous au *Guide de l'utilisateur du logiciel Advanced Lights Out Manager Software de Sun pour le serveur Netra 240* (817-5002-11).

Définition du mot de passe `admin`

Lorsque vous passez au logiciel ALOM après la mise sous tension initiale, l'invite `sc>` s'affiche. À ce stade, vous pouvez exécuter les commandes qui ne requièrent aucune autorisation utilisateur. (Pour obtenir la liste de ces commandes, reportez-vous au *Guide de l'utilisateur du logiciel Advanced Lights Out Manager Software de Sun pour le serveur Netra 240* (817-5002-11). Lorsque vous tentez d'exécuter une commande nécessitant des permissions utilisateur, un message vous invite à définir un mot de passe pour l'utilisateur `admin`.

- **À l'invite, le cas échéant, définissez un mot de passe pour l'utilisateur `admin`.**

Le mot de passe doit contenir :

- au moins deux caractères alphabétiques ;
- au moins un caractère numérique ou spécial ;
- entre six et huit caractères.

Une fois le mot de passe défini, l'utilisateur `admin` détient tous les droits et peut exécuter toutes les commandes de l'interface de ligne de commandes ALOM.

L'utilisateur est invité à se connecter à l'aide du mot de passe `admin` lorsqu'il accède par la suite au logiciel ALOM.

Fonctions ALOM de base

Cette section aborde certaines fonctions ALOM de base. Pour une documentation complète, reportez-vous au *Guide de l'utilisateur du logiciel Advanced Lights Out Manager Software de Sun pour le serveur Netra 240* (817-5002-11) et au *Netra 240 Server Release Notes* (817-3142-xx).

▼ Pour passer à l'invite ALOM

- À l'invite, appuyez sur les touches # . :

```
# #.
```

Remarque : Lorsque vous passez à l'invite ALOM, vous êtes connecté avec l'ID d'utilisateur `admin`. Reportez-vous à la section « Définition du mot de passe `admin` » à la page 43.

▼ Pour passer à l'invite de la console serveur

- Saisissez :

```
sc> console
```

Plusieurs utilisateurs ALOM peuvent se connecter simultanément à la console serveur, mais un seul utilisateur a le droit de saisir des caractères d'entrée sur la console.

Si un autre utilisateur est connecté et détient des droits en écriture, le message suivant apparaît après la saisie de la commande `console` :

```
sc> Console session already in use. [view mode]
```

▼ Pour supprimer les droits d'écriture d'un utilisateur sur la console

- Saisissez :

```
sc> console -f
```

Redémarrage automatique du serveur

Remarque : La fonction Automatic System Recovery (ASR) est différente de celle de redémarrage automatique du serveur, également prise en charge par le serveur Netra 240.

Le redémarrage automatique du serveur est un composant d'ALOM. Cette fonction surveille l'environnement d'exploitation Solaris au cours de son exécution, synchronise par défaut les systèmes de fichiers et redémarre le serveur si ce dernier connaît une défaillance.

ALOM utilise un processus de surveillance pour surveiller le noyau *uniquement*. ALOM ne redémarrera pas le serveur si un processus est suspendu et que le noyau est en cours d'exécution. Les paramètres de surveillance ALOM pour l'intervalle de vérification et le délai de surveillance ne sont pas configurables par l'utilisateur.

Si le noyau est suspendu et que le délai de surveillance expire, ALOM rapporte et consigne l'événement et effectue l'une des trois actions configurables par l'utilisateur.

- *xir* – il s'agit de l'action par défaut ; le serveur effectue la synchronisation des systèmes de fichiers et redémarre. Si le système se bloque, ALOM effectue de nouveau une réinitialisation matérielle au bout de 15 minutes.
- *Reset* – il s'agit d'une réinitialisation matérielle qui entraîne une reprise rapide du système. Toutefois, les données concernant la suspension ne sont pas stockées et d'importants dommages peuvent se produire.
- *None* – cette action laisse indéfiniment le serveur en état de suspension après le signalement de l'expiration du délai de surveillance.

Pour plus d'informations, reportez-vous à la section `sys_autorestart` du *Guide de l'utilisateur du logiciel Advanced Lights Out Manager Software de Sun pour le serveur Netra 240 (817-5002-11)*.

Pour apprendre à utiliser Automatic System Recovery (ASR), reportez-vous au Chapitre 1.

Surveillance et contrôle de l'environnement

Le serveur Netra 240 comporte un sous-système de surveillance de l'environnement conçu pour protéger le serveur et ses composants contre :

- les températures extrêmes ;
- une ventilation insuffisante du système ;
- une utilisation avec des composants manquants ou mal configurés ;
- les pannes d'alimentation ;
- des erreurs matérielles internes.

Les fonctions de surveillance et de contrôle sont gérées par le microprogramme ALOM, qui garantit que les fonctions de surveillance sont toujours opérationnelles, même si le système est arrêté ou ne peut pas s'initialiser. Par ailleurs, le fait de surveiller le système depuis le microprogramme ALOM libère le système afin d'affecter des ressources d'UC et mémoire au système d'exploitation et aux logiciels applicatifs.

Le sous-système de surveillance de l'environnement utilise un bus I²C normalisé. Le bus I²C est un bus série bifilaire permettant de surveiller et de contrôler les capteurs thermiques, les ventilateurs, les blocs d'alimentation, les voyants d'état et le commutateur de contrôle du système du panneau avant.

Le serveur contient trois capteurs de température qui surveillent la température ambiante du serveur et la température des deux UC. Le sous-système de surveillance interroge les capteurs et utilise les températures échantillonnées pour signaler les conditions de température excessive ou insuffisante et déclencher les actions appropriées. D'autres périphériques I²C détectent la présence et les problèmes des composants.

Les composants matériels et logiciels du sous-système de surveillance garantissent que la température ambiante du serveur ne s'écarte pas d'une plage de températures prédéfinie. Si la température observée par un capteur dépasse un seuil d'avertissement de température insuffisante ou excessive, le logiciel du sous-système de surveillance allume les voyants Maintenance requise situés sur les panneaux avant et arrière. Si une telle température persiste et atteint un seuil entraînant un arrêt progressif (température élevée ou faible), le système procède à l'arrêt progressif du système. Si une telle température persiste et atteint un seuil entraînant un arrêt forcé (température faible ou élevée), le système procède à l'arrêt forcé du système.

Tous les messages d'erreur et d'avertissement sont affichés sur la console du système et consignés dans le fichier `/var/adm/messages`. Les voyants Maintenance requise restent allumés pour faciliter le diagnostic des problèmes.

Les types de messages transmis à la console du système et consignés dans le fichier `/var/adm/messages` dépendent du paramétrage des variables utilisateur ALOM `sc_clieventlevel` et `sys_eventlevel`. Pour de plus amples informations sur la configuration de ces variables, reportez-vous au *Guide de l'utilisateur du logiciel Advanced Lights Out Manager de Sun pour le serveur Netra 240* (817-5002-11).

TABLEAU 3-2 Seuils de température du système du serveur Netra 240

Seuils de température	Température	Action du serveur
Température basse - Arrêt forcé	-11 °C	Le serveur procède à un arrêt forcé du système.
Température basse - Arrêt progressif	-9 °C	Le serveur procède à un arrêt progressif du système.
Température basse - Avertissement	-7 °C	Le serveur allume les voyants Maintenance requise situés sur les panneaux avant et arrière.
Température élevée - Avertissement	57 °C	Le serveur allume les voyants Maintenance requise situés sur les panneaux avant et arrière.
Température élevée - Arrêt progressif	60 °C	Le serveur procède à un arrêt progressif du système.
Température élevée - Arrêt forcé	63 °C	Le serveur procède à un arrêt forcé du système.

Le sous-système de surveillance est également conçu pour détecter les pannes des ventilateurs. Si l'un des ventilateurs tombe en panne, le sous-système de surveillance détecte la panne, affiche un message d'erreur sur la console du système, le consigne dans le fichier `/var/adm/messages` et allume les voyants Maintenance requise.

Le sous-système d'alimentation est contrôlé de la même manière. En interrogeant régulièrement l'état des blocs d'alimentation, le sous-système de surveillance indique l'état des sorties, entrées et présence de chaque bloc.

Si un problème d'alimentation est détecté, la console affiche un message d'erreur, puis le consigne dans le fichier `/var/adm/messages`. De plus, les voyants placés sur chaque bloc d'alimentation s'allument en cas de pannes. Les voyants Maintenance requise du système s'allument en cas d'erreurs système. Les alertes de la console ALOM enregistrent les pannes d'alimentation.

Utilisez la commande ALOM `showenvironment` pour afficher les seuils d'avertissement du sous-système d'alimentation et les vitesses des ventilateurs. Pour de plus amples informations sur l'utilisation de cette commande, reportez-vous au *Guide de l'utilisateur du logiciel Advanced Lights Out Manager de Sun pour le serveur Netra 240* (817-5002-11).

API de sortie des relais d'alerte

Cette annexe propose un exemple de programme (EXEMPLE DE CODE A-1) montrant comment obtenir/définir (get/set) l'état des alarmes. L'application peut utiliser la fonction `ioctl LOMIOCALSTATE` pour obtenir l'état de chaque alarme et la fonction `ioctl LOMIOCALCTL` pour les définir sur une base individuelle. Pour plus d'informations sur les voyants d'alerte, reportez-vous au *Netra 240 Server Service Manual* (817-2699-xx).

EXEMPLE DE CODE A-1 Exemple de programme pour l'obtention et la définition de l'état des alarmes (get/set)

```
#include <sys/types.h>
#include <string.h>
#include <stdlib.h>
#include <sys/unistd.h>
#include <fcntl.h>
#include "lom_io.h"

#define ALARM_INVALID -1
#define LOM_DEVICE "/dev/lom"

static void usage();
static void get_alarm(const char *alarm);
static int set_alarm(const char *alarm, const char *alarmval);
static int parse_alarm(const char *alarm);
static int lom_ioctl(int ioc, char *buf);
static char *get_alarmval(int state);
static void get_alarmvals();

main(int argc, char *argv[])
{
    if (argc < 3) {
        usage();
        if (argc == 1)
```

EXEMPLE DE CODE A-1 Exemple de programme pour l'obtention et la définition de l'état des alarmes
(get/set) (suite)

```
#include <sys/types.h>
        get_alarmvals();
        exit (1);
    }

    if (strcmp(argv[1], "get") == 0) {
        if (argc != 3) {
            usage();
            exit (1);
        }
        get_alarm(argv[2]);
    }
    else
    if (strcmp(argv[1], "set") == 0) {
        if (argc != 4) {
            usage();
            exit (1);
        }
        set_alarm(argv[2], argv[3]);
    } else {
        usage();
        exit (1);
    }
}

static void
usage()
{
    printf("usage: alarm [get|set] [crit|major|minor|user] [on|off]\n");
}

static void
get_alarm(const char *alarm)
{
    ts_aldata_t    ald;
    int altype = parse_alarm(alarm);
    char *val;

    if (altype == ALARM_INVALID) {
        usage();
        exit (1);
    }

    ald.alarm_no = altype;
    ald.alarm_state = ALARM_OFF;
```

EXEMPLE DE CODE A-1 Exemple de programme pour l'obtention et la définition de l'état des alarmes
(get/set) (suite)

```
#include <sys/types.h>
    lom_ioctl(LOMIOCALSTATE, (char *)&ald);

    if ((ald.alarm_state != ALARM_OFF) &&
        (ald.alarm_state != ALARM_ON)) {
        printf("Invalid value returned: %d\n", ald.alarm_state);
        exit (1);
    }

    printf("ALARM.%s = %s\n", alarm, get_alarmval(ald.alarm_state));
}

static int
set_alarm(const char *alarm, const char *alarmstate)
{
    ts_aldata_t    ald;
    int alarmval = ALARM_OFF, altype = parse_alarm(alarm);

    if (altype == ALARM_INVALID) {
        usage();
        exit (1);
    }

    if (strcmp(alarmstate, "on") == 0)
        alarmval = ALARM_ON;
    else
        if (strcmp(alarmstate, "off") == 0)
            alarmval = ALARM_OFF;
        else {
            usage();
            exit (1);
        }

    ald.alarm_no = altype;
    ald.alarm_state = alarmval;

    if (lom_ioctl(LOMIOCALCTL, (char *)&ald) != 0) {
        printf("Setting ALARM.%s to %s failed\n", alarm, alarmstate);
        return (1);
    } else {
        printf("Setting ALARM.%s successfully set to %s\n", alarm,
alarmstate);
        return (1);
    }
}
```

EXEMPLE DE CODE A-1 Exemple de programme pour l'obtention et la définition de l'état des alarmes
(get/set) *(suite)*

```
#include <sys/types.h>
static int
parse_alarm(const char *alarm)
{
    int altype;

    if (strcmp(alarm, "crit") == 0)
        altype = ALARM_CRITICAL;
    else
    if (strcmp(alarm, "major") == 0)
        altype = ALARM_MAJOR;
    else
    if (strcmp(alarm, "minor") == 0)
        altype = ALARM_MINOR;
    else
    if (strcmp(alarm, "user") == 0)
        altype = ALARM_USER;
    else {
        printf("invalid alarm value: %s\n", alarm);
        altype = ALARM_INVALID;
    }

    return (altype);
}

static int
lom_ioctl(int ioc, char *buf)
{
    int fd, ret;

    fd = open(LOM_DEVICE, O_RDWR);

    if (fd == -1) {
        printf("Error opening device: %s\n", LOM_DEVICE);
        exit (1);
    }

    ret = ioctl(fd, ioc, (void *)buf);

    close (fd);

    return (ret);
}

static char *
```

EXEMPLE DE CODE A-1 Exemple de programme pour l'obtention et la définition de l'état des alarmes
(get/set) *(suite)*

```
#include <sys/types.h>
get_alarmval(int state)
{
    if (state == ALARM_OFF)
        return ("off");
    else
        if (state == ALARM_ON)
            return ("on");
        else
            return (NULL);
}
static void
get_alarmvals()
{
    get_alarm("crit");
    get_alarm("major");
    get_alarm("minor");
    get_alarm("user");
}
}
```


Index

A

adaptateur hôte (`probe-scsi`), 12
Advanced Lights Out Manager
 voir ALOM
alarme
 API de sortie de relais, 49 à 53
 état `get`, 49 à 53
 état `set`, 49 à 53
ALOM
 définition des mots de passe, 43
 fonctions de base, 43
 outil de diagnostic, 2
 ports, 42
 présentation, 41
 Redémarrage automatique du serveur, 45
 sous-système de surveillance de
 l'environnement, 46
 voyants d'état, 4
arborescence, logiciel Solaris, affichage, 19
ASR, 29
Automatic System Recovery, *voir aussi* ASR, 29
autotest intégré, variable `test-args`, 16

B

BIST, *voir* autotest intégré
bus IDE, 13

C

capteurs de température, 46
carte d'alarme
 états d'alarme, 6
 voyants d'alarme, 6

carte de configuration système, 8
chemins de périphérique matériel, 13, 17
commande `prtconf`, Solaris, 19
commande `prtdiag`, Solaris, 21
commande `prtfru`, Solaris, 23
commande `psrinfo`, Solaris, 24
commande `show-devs`, OpenBoot, 13
commande `showrev`, Solaris, 25
commande `test` (tests OpenBoot Diagnostics), 17
commande `test-all` (tests OpenBoot
 Diagnostics), 17
Commandes OpenBoot
 exécuter, 14
 outil de diagnostic, 2
 `probe-ide`, 13
 `probe-scsi` et `probe-scsi-all`, 12
 `show-devs`, 13
Commandes Solaris
 outil de diagnostic, 2
 `prtconf`, 19
 `prtdiag`, 21
 `prtfru`, 23
 `psrinfo`, 24
 `showrev`, 25

D

diagnostics
 Microprogramme, 14
 POST, 9
 SE Solaris, 18
 SunVTS, 35

E

état des relais
 fermé normalement (NC), 7
 ouvert normalement (NO), 7
états d'alarme, contact sec, 6
événements de réinitialisation, types, 10

F

fermé normalement (NC), état des relais, 7
fichiers journaux, 19
 messages d'erreur, 19
 messages système, 19
fréquence d'horloge, UC, 24
fréquence du processeur, affichage, 24

I

I²C, bus, 46
I²C, capteurs, 46
ID en boucle (*probe-scsi*), 12
Integrated Drive Electronics (IDE), *voir* bus IDE
invite serveur
 invite Advanced Lights-Out Manager, 3
 invite OpenBoot, 3
 invite superutilisateur du logiciel Solaris, 3

M

Maintenance requise, voyant, 46
majeure, voyant d'alarme, 7
mémoire système, détermination de la taille, 19
message
 interprétation des erreurs, 17
 POST, erreur, 8
messages d'erreur
 fichier journal, 47
 liés à l'alimentation, 47
 OpenBoot Diagnostics, interprétation, 17
 tests OpenBoot Diagnostics, 17
mineure, voyant d'alarme, 7

N

numéro d'unité logique (*probe-scsi*), 12

O

OpenBoot (diagnostics), 14
 contrôle des tests, 16
 lancement, 15
 outil de diagnostic, 2
outil de diagnostic
 ALOM, 2
 auto-test à la mise sous tension, 2
 commande logicielle Solaris, 2
 commande OpenBoot, 2
 OpenBoot (diagnostics), 2
 SunVTS, 2
 voyants, 2
outils de dépannage, 2
ouvert normalement (NO), état des relais, 7

P

paramètres OpenBoot PROM, variable
 diag-level, 8
patches, installés, *showrev*, 25
périphérique matériel, chemins, 13, 17
périphériques SCSI, diagnostics des problèmes, 12
périphériques USB, auto-tests OpenBoot
 Diagnostics, 17
POST
 démarrer les diagnostics, 11
 diagnostics, contrôle, 9
 messages d'erreur, 8
 outil de diagnostic, 2
Auto-test à la mise sous tension
 voir POST
probe-ide, commande (OpenBoot), 13
probe-scsi et *probe-scsi-all*, commandes
 (OpenBoot), 12

R

Redémarrage automatique du serveur, 45
révision matériel, affichage avec *showrev*, 25
révision, logiciel et matériel, affichage avec
 showrev, 25

S

- SE Solaris
 - arborescence des périphériques, 19
 - SunVTS, 36
- SEAM, 37
- sous-système de surveillance de l'environnement, 46
- Sun Enterprise Authentication Mechanism, *voir* SEAM
- Sun Validation Test Suite, *voir* SunVTS
- SunVTS, 35 à 39
 - documentation, 39
 - installation, 38
 - interfaces, 35
 - logiciel, modes de test, 36
 - outil de diagnostic, 2
 - présentation, 35
 - progiciels en option, 36
 - SE Solaris, 36
 - sécurité de base, 37
 - sécurité SEAM, 37
 - tests disponibles, 36
 - vérifier l'installation, 38
 - version compatible, 35, 38
- surchauffe, 22
- surchauffe, sous-système de surveillance, 46
- surveillance, sous-système
 - surchauffe, 46
 - température insuffisante, 46
- système, voyants d'état
 - voyants d'erreur liée à un problème d'environnement, 47
 - Voir aussi* Voyants

T

- température insuffisante, sous-système de surveillance, 46
- tests de diagnostic, ignorer, 10
- tests OpenBoot Diagnostics
 - à l'invite `ok`, 17
 - chemins de périphérique matériel, 17
 - commande `test`, 17
 - commande `test-all`, 17
 - messages d'erreur, interprétation, 17

U

- UC
 - affichage des informations, 24
 - fréquence d'horloge, 24
- unité centrale, *voir* UC
- unité interchangeable sur site, 23 à 24
- unités d'alimentation, surveillance de pannes, 47
- utilisateur, voyant d'alarme, 7

V

- variable `auto-boot?`, 9
- variable `diag-level`, 9, 16
- variable `diag-script`, 9
- variable `diag-switch?`, 8, 9
- variable `input-device`, 10
- variable `obdiag-trigger`, 10
- variable `output-device`, 10
- variable `post-trigger`, 10
- variable `test-args`, 16
 - mots-clés, 16
 - personnaliser les tests, 16
- Variables de configuration OpenBoot
 - description, 9
 - mots-clés, 9
- voyant Localisation, 4
 - allumé, 5
 - état, 5
 - off, 5
- voyants d'alarme, 6
 - critical, 6
 - majeure, 7
 - mineure, 7
 - user, 7
- voyants d'état du serveur, avant et arrière, 4
- voyants, outil de diagnostic, 2

W

- `watch-net`, 28
- World Wide Name (`probe-scsi`), 12

