



Guide de la fonction de contrôle de Sun Java Enterprise System 5



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Référence : 820-1589-10
Mars 2007

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée au produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs des brevets américains et une ou plusieurs des demandes de brevet en attente aux États-Unis et dans d'autres pays.

Droits soumis à la loi américaine – Logiciel de commerce. Les utilisateurs gouvernementaux sont soumis au contrat de licence standard de Sun Microsystems, Inc. et aux dispositions applicables du FAR et de ses compléments.

Cette distribution peut comprendre des composants développés par des tierces parties.

Des parties de ce produit peuvent être dérivées des systèmes Berkeley BSD concédés sous licence par l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays et concédée exclusivement sous licence par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, JavaScript, Java, JavaServer Pages, JSP et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux États-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 États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface graphique OPEN LOOK et SunTM a été développée par Sun Microsystems, Inc. pour ses utilisateurs et détenteurs de licence. Sun reconnaît les efforts novateurs de Xerox en ce qui concerne la recherche et le développement du concept des interfaces visuelles ou graphiques dans le domaine informatique. Sun détient une licence non exclusive de Xerox pour l'interface utilisateur graphique Xerox, qui couvre aussi les détenteurs de licences Sun qui implémentent des IG OPEN LOOK et respectent autrement les contrats de licence écrits de Sun.

Les produits mentionnés dans cette publication et les informations fournies sont soumis à la législation américaine en matière de contrôle des exportations et peuvent être soumis à la réglementation en vigueur dans d'autres pays dans le domaine des exportations et importations. L'utilisation d'armes nucléaires, de missiles, d'armes biologiques et chimiques ou d'armes nucléaires maritimes, qu'elle soit directe ou indirecte, est strictement interdite. Les exportations ou réexportations vers les pays sous embargo américain, ou vers des entités figurant sur les listes d'exclusion des exportations américaines, y compris, mais de manière non exhaustive, la liste des personnes refusées et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

CETTE DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTE CONDITION, DÉCLARATION ET GARANTIE EXPRESSE OU TACITE, Y COMPRIS TOUTE GARANTIE IMPLICITE DE QUALITÉ MARCHANDE, D'ADÉQUATION À UN BUT PARTICULIER OU D'ABSENCE DE CONTREFAÇON, EST EXCLUE, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.

Table des matières

Préface	7
1 Présentation de la fonction de contrôle de Java ES	15
Présentation des composants Monitoring Framework et Monitoring Console	15
Principe de fonctionnement de la fonction de contrôle de Java ES	16
Modèle CMM (Modèle de contrôle commun)	17
Instrumentation CMM	17
Agents de noeud	18
Agent maître	19
Ordre d'installation conseillé	20
2 Activation et configuration du composant Monitoring Framework	21
Organisation des répertoires installés	22
Utilisation du composant Monitoring Framework avec Access Manager	23
▼ Pour activer la fonction de contrôle dans Access Manager	23
Utilisation du composant Monitoring Framework avec Application Server	24
▼ Pour activer la fonction de contrôle dans Application Server	24
Utilisation du composant Monitoring Framework avec Calendar Server	25
▼ Pour activer la fonction de contrôle dans Calendar Server	25
Utilisation du composant Monitoring Framework avec Directory Server	26
▼ Pour activer la fonction de contrôle dans Directory Server	26
Utilisation du composant Monitoring Framework avec Instant Messaging	26
▼ Pour activer la fonction de contrôle dans Instant Messaging	26
Utilisation du composant Monitoring Framework avec Messaging Server	27
▼ Pour activer la fonction de contrôle dans Messaging Server	27
Utilisation du composant Monitoring Framework avec Portal Server	28
▼ Pour activer la fonction de contrôle dans Portal Server	28

Utilisation du composant Monitoring Framework avec Web Server	28
▼ Pour activer la fonction de contrôle dans Web Server	28
Configuration du Conteneur d'agents communs	29
▼ Pour activer la fonction de contrôle dans le Conteneur d'agents communs	29
Dépannage du composant Monitoring Framework	30
Utilisation du composant Monitoring Framework sur les plates-formes HP-UX	30
Utilisation du composant Monitoring Framework sous Microsoft Windows	30
▼ Pour redémarrer l'agent de noeud	31
La commande mfwkadm	31
Descriptif	31
Description	33
Options	34
Sous-commandes	34
Exemples	46
État de sortie	51
Attributs	51
Voir aussi	52
3 Installation et utilisation du composant Monitoring Console	53
Installation du composant Monitoring Console	53
▼ Pour installer le composant Monitoring Console avec le programme d'installation Java ES	54
▼ Pour installer le composant Monitoring Console dans une zone Solaris	55
▼ Pour configurer le composant Monitoring Console	56
▼ Pour annuler la configuration du composant Monitoring Console	56
Organisation des répertoires installés	57
Démarrage du composant Monitoring Console	57
▼ Pour démarrer le composant Monitoring Console	58
▼ Pour se connecter aux agents de noeud	59
Utilisation du composant Monitoring Console	61
▼ Pour désactiver et réactiver le contrôle sélectif des composants	61
▼ Pour créer une nouvelle règle de contrôle	62
Dépannage du composant Monitoring Console	70

A	Références sur les objets CMM	73
	Présentation des objets CMM	73
B	Objets contrôlés exposés par chaque composant	77
	Instrumentation du Conteneur d'agents communs	77
	Instrumentation du composant Access Manager	77
	Instrumentation du composant Application Server	77
	Instrumentation du composant Calendar Server	77
	Instrumentation du composant Directory Server	78
	Instrumentation du composant Instant Messaging	78
	Instrumentation du composant Messaging Server	78
	Instrumentation du composant Portal Server	78
	Instrumentation du composant Web Server	78
	Index	79

Préface

Ce manuel décrit la nouvelle fonction de contrôle intégrée à Sun Java™ Enterprise System 5 (Java ES). Cette fonction est implémentée par le biais des composants Sun Java System Monitoring Framework 2.0 et Sun Java System Monitoring Console 1.0.

Les procédures présentées dans ce manuel expliquent comment configurer et activer Monitoring Framework pour chacun des composants installés, puis afficher les données contrôlées à partir du composant Monitoring Console. Le présent manuel n'aborde pas les fichiers journaux ni aucun autre mécanisme de contrôle des composants individuels en dehors de la structure.

À qui ce manuel s'adresse-t-il ?

Ce manuel s'adresse :

- Aux architectes logiciels qui doivent établir un programme de maintenance pour les déploiements de Java ES.
- Aux administrateurs système chargés de l'installation et de la configuration de Java ES.
- Aux administrateurs système et aux techniciens qui assurent la surveillance et la maintenance des déploiements de Java ES.

Avant de lire ce manuel

Vous devez être familiarisé avec la documentation disponible pour Java ES qui est décrite à la section suivante. Vous devez également comprendre les principes de conception et de fonctionnement des composants de Java ES à contrôler.

Par ailleurs, si vous envisagez d'installer et de configurer les composant de contrôle, vous devez préalablement installer tous les autres composants. Avant d'entreprendre toute installation ou configuration, consultez les *Notes de version de Sun Java Enterprise System 5 pour UNIX* .

Documentation Java ES

Cette documentation décrit la planification du déploiement et l'installation du système. La documentation système est disponible à l'adresse suivante : <http://docs.sun.com/coll/1286.2>. Pour une présentation de Java ES, reportez-vous aux ouvrages présentés ci-dessous en suivant l'ordre dans lequel ils figurent.

TABLEAU P-1 Documentation Java Enterprise System

Titre du document	Contenu
<i>Notes de version de Sun Java Enterprise System 5 pour UNIX</i>	Ce manuel contient les informations les plus récentes sur Java ES, y compris les problèmes connus. De plus, des notes de version existent pour les différents composants, comme indiqué dans la Collection de notes de version (http://docs.sun.com/coll/1315.2).
<i>Sun Java Enterprise System 5 Release Notes for Microsoft Windows</i>	
<i>Présentation technique de Sun Java Enterprise System 5</i>	Ce manuel présente les principes techniques et conceptuels de Java ES. Il décrit les composants, l'architecture, les processus et les fonctions.
<i>Sun Java Enterprise System Deployment Planning Guide</i>	Ce manuel indique brièvement les modalités de planification et de conception des solutions de déploiement d'entreprise basées sur Java ES. Il présente quelques principes et concepts fondamentaux de la planification et de la conception de déploiement, traite du cycle de vie d'une solution et fournit des exemples et des stratégies de haut niveau à utiliser lors de la planification de solutions basées sur Java ES.
<i>Guide de planification pour l'installation de Sun Java Enterprise System 5</i>	Ce manuel vous aide à développer les spécifications d'implémentation pour le matériel, le système d'exploitation et les différents aspects réseau de votre déploiement Java ES. Il décrit les aspects à prendre en compte dans votre plan d'installation et de configuration tels que les dépendances entre les composants.
<i>Guide d'installation de Sun Java Enterprise System 5 pour UNIX</i>	
<i>Sun Java Enterprise System 5 Installation Guide for Microsoft Windows</i>	Ce manuel vous guide tout au long du processus d'installation de Java ES. Il explique également comment configurer les composants après l'installation et vérifier leur bon fonctionnement.
<i>Référence de l'installation de Sun Java Enterprise System 5 pour UNIX</i>	Ce manuel fournit des informations supplémentaires sur les paramètres de configuration, des feuilles de travail destinées à être utilisées pour la planification de votre configuration, ainsi qu'une liste des matériaux de référence tels que les répertoires par défaut et les numéros de port pour le système d'exploitation Solaris et l'environnement d'exploitation Linux.

TABLEAU P-1 Documentation Java Enterprise System (Suite)

Titre du document	Contenu
<i>Guide de mise à niveau de Sun Java Enterprise System 5 pour UNIX</i>	Ce manuel fournit les instructions pour effectuer la mise à jour vers Java ES 5 à partir de versions installées précédentes.
<i>Sun Java Enterprise System 5 Upgrade Guide for Microsoft Windows</i>	
<i>Guide de la fonction de contrôle de Sun Java Enterprise System 5</i>	Ce manuel fournit des instructions de configuration du composant Monitoring Framework pour chaque composant du produit et la procédure d'utilisation du composant Monitoring Console pour afficher les données en temps réel et créer des règles de contrôle.
<i>Sun Java Enterprise System Glossary</i>	Ce glossaire contient les définitions des termes utilisés dans la documentation Java ES.

Chemins par défaut et noms de fichier

Le tableau suivant présente les chemins et les noms de fichiers par défaut des composants de Java ES qui implémentent la fonction de contrôle.

TABLEAU P-2 Chemins et noms de fichiers par défaut

Paramètre substituable	Description	Valeur par défaut
<i>base-mfwk</i>	Répertoire dans lequel les composants partagés de Monitoring Framework sont installés automatiquement. Ce chemin est également utilisé comme répertoire de configuration.	Systèmes Solaris : /opt/SUNWmfwk Systèmes Linux : /opt/sun/mfwk
<i>base-MConsole</i>	Répertoire d'installation choisi pour le composant Monitoring Console.	Systèmes Solaris : /opt/SUNWjesmc Systèmes Linux : /opt/sun/jesmc
<i>base-WebConsole</i>	Répertoire dans lequel le composant partagé Web Console est installé automatiquement.	Systèmes Solaris : /etc/webconsole/console Systèmes Linux : /etc/opt/webconsole/console
<i>base-AccessMgr</i>	Répertoire d'installation choisi pour le composant Sun Java System Access Manager.	Systèmes Solaris : /opt/SUNWam Systèmes Linux : /opt/sun/identity
<i>base-AppServer</i>	Répertoire d'installation choisi pour le composant Sun Java System Application Server.	Systèmes Solaris : /opt/SUNWappserver/appserver Systèmes Linux : /opt/sun/appserver

TABLEAU P-2 Chemins et noms de fichiers par défaut (Suite)

Paramètre substituable	Description	Valeur par défaut
<i>base-CalServ</i>	Répertoire d'installation choisi pour le composant Sun Java System Calendar Server.	Systèmes Solaris : /opt/SUNWics5 Systèmes Linux : /opt/sun/calendar
<i>base-DirServ</i>	Répertoire d'installation choisi pour le composant Sun Java System Directory Server.	Systèmes Solaris : /opt/SUNWdsee/ds6 Systèmes Linux : /opt/sun/ds6
<i>base-IM</i>	Répertoire d'installation choisi pour le composant Sun Java System Instant Messaging.	Systèmes Solaris : /opt/SUNWiim Systèmes Linux : /opt/sun/im
<i>base-MsgServ</i>	Répertoire d'installation choisi pour le composant Sun Java System Messaging Server.	Systèmes Solaris : /opt/SUNWmsgsr Systèmes Linux : /opt/sun/messaging
<i>base-Portal</i>	Répertoire d'installation choisi pour Sun Java System Portal Server.	Systèmes Solaris : /opt/SUNWportal Systèmes Linux : /opt/sun/portal
<i>base-WebServer</i>	Répertoire d'installation choisi pour le composant Sun Java System Web Server.	Systèmes Solaris : /opt/SUNWwbsvr7 Systèmes Linux : /opt/sun/webserver

Conventions typographiques

Les conventions typographiques employées dans ce document sont décrites dans le tableau suivant.

TABLEAU P-3 Conventions typographiques

Police	Description	Exemple
AaBbCc123	Nom de commandes, fichiers et répertoires. Messages système s'affichant à l'écran.	Modifiez votre fichier <code>.login</code> . Utilisez <code>ls -a</code> pour afficher la liste de tous les fichiers. nom-machine% Vous avez reçu du courrier.
AaBbCc123	Ce que vous entrez, et non ce qui apparaît sur l'écran de l'ordinateur.	nom-machine% su Mot de passe :
<i>AaBbCc123</i>	Marque de réservation à remplacer par un vrai nom ou une valeur réelle	La commande à utiliser pour supprimer un fichier est <code>rm nom-fichier</code> .

TABLEAU P-3 Conventions typographiques (Suite)

Police	Description	Exemple
<i>AaBbCc123</i>	Titres de manuels, nouveaux termes et termes importants (notez que certains éléments importants sont affichés en gras en ligne)	Lisez le chapitre 6 du <i>Guide de l'utilisateur</i> . Un <i>cache</i> est une copie de ce qui est stocké localement. N'enregistrez <i>pas</i> le fichier.

Invites de Shell dans des exemples de commande

Le tableau suivant présente les invites système par défaut, ainsi que les invites de superutilisateur.

TABLEAU P-4 Invites de shell

Shell	Invite
C shell sur les systèmes UNIX et Linux	nom-machine%
Superutilisateur C shell sur les systèmes UNIX et Linux	nom-machine#
Bourne shell et Korn shell sur les systèmes UNIX et Linux	\$
Superutilisateur Bourne shell et Korn shell sur les systèmes UNIX et Linux	#
Ligne de commande Microsoft Windows	C:\

Conventions de symboles

Le tableau suivant présente les symboles susceptibles d'être utilisés dans ce manuel.

TABLEAU P-5 Conventions de symboles

Symbole	Description	Exemple	Signification
[]	Contient des arguments optionnels et des options de commande.	ls [-l]	L'option -l n'est pas obligatoire.
{ }	Délimite un ensemble de choix pour une option de commande obligatoire.	-d {y n}	L'option -d nécessite l'utilisation de l'argument y ou l'argument n.
\${ }	Indique une référence de variable.	\${com.sun.javaRoot}	Fait référence à la valeur de la variable com.sun.javaRoot.

TABLEAU P-5 Conventions de symboles (Suite)

Symbole	Description	Exemple	Signification
-	Joint plusieurs touches utilisées simultanément.	Ctrl-A	Appuyez sur la touche Ctrl tout en maintenant la touche A enfoncée.
+	Joint plusieurs touches utilisées de manière consécutive.	Ctrl+A+N	Appuyez sur la touche Ctrl, relâchez-la, puis appuyez sur les touches suivantes.
→	Indique une sélection d'options de menu dans une interface graphique.	Fichier → Nouveau → Modèles	Dans le menu Fichier, sélectionnez Nouveau. Dans le sous-menu Nouveau, choisissez Modèles.

Documentation, support et formation

Le site Web de Sun fournit des informations sur les ressources supplémentaires suivantes :

- Documentation (<http://www.sun.com/documentation/>)
- Support (<http://www.sun.com/support/>)
- Formation (<http://www.sun.com/training/>)

Recherche de documentation sur les produits Sun

La recherche de documentation sur les produits Sun peut s'effectuer à partir du site Web docs.sun.comSM, mais aussi à l'aide d'un moteur de recherche. Dans ce cas, il vous suffit de saisir le texte suivant dans le champ de recherche :

```
terme-recherche site:docs.sun.com
```

Par exemple, pour lancer une recherche sur “courtier,” tapez le texte suivant :

```
courtier site:docs.sun.com
```

Pour inclure d'autres sites Web de Sun dans votre recherche (par exemple, java.sun.com, www.sun.com et developers.sun.com), remplacez docs.sun.com par sun.com dans le champ de recherche.

Références aux sites Web tiers associés

Des adresses URL de sites tiers, qui renvoient à des informations complémentaires connexes, sont référencées dans ce document.

Remarque – Sun décline toute responsabilité quant à la disponibilité de sites Web de tiers mentionnés dans ce document. Sun ne garantit pas le contenu, la publicité, les produits et autres matériaux disponibles sur ces sites ou dans ces ressources, ou accessibles par leur intermédiaire, et ne saurait en être tenu pour responsable. Par ailleurs, la responsabilité de Sun ne saurait être engagée en cas de dommages ou de pertes, réels ou supposés, occasionnés par, ou liés à, l'utilisation du contenu, des produits ou des services disponibles sur ces sites ou dans ces ressources, ou accessibles par leur biais, ou encore à la confiance qui a pu leur être accordée.

Vos commentaires sont les bienvenus

Afin d'améliorer sa documentation, Sun vous encourage à faire des commentaires et à apporter des suggestions. Pour nous faire part de vos remarques, rendez-vous sur le site <http://docs.sun.com> et cliquez sur Send Comments. Dans le formulaire en ligne, indiquez le titre complet et le numéro de référence du document. Ce numéro comporte 7 ou 9 chiffres et se trouve sur la page de titre du manuel ou dans l'URL du document. Par exemple, le numéro de référence de ce manuel est 819-5081-10.

Présentation de la fonction de contrôle de Java ES

Ce manuel décrit les composants Monitoring Framework 2.0 et Monitoring Console 1.0 de Sun Java™ Enterprise System (Java ES). Utilisés conjointement, ces deux composants permettent d'implémenter la nouvelle fonction de contrôle introduite dans la version 5 du produit.

Les procédures présentées dans ce manuel expliquent comment activer le composant Monitoring Framework pour chacun des composants installés, puis afficher les données contrôlées à partir du composant Monitoring Console. Ce manuel n'aborde pas les fichiers journaux, les messages d'erreur, ni aucun autre mécanisme de contrôle implémenté par les composants individuels en dehors de la structure. Les composants Monitoring Framework et Monitoring Console n'intègrent pas de fonction de gestion ou d'administration des composants contrôlés. Pour plus d'informations sur l'administration d'un composant spécifique, reportez-vous à la documentation associée.

Ce chapitre présente les concepts de contrôle et l'architecture du composant Monitoring Framework.

Il se compose des sections suivantes :

- “Présentation des composants Monitoring Framework et Monitoring Console” à la page 15
- “Principe de fonctionnement de la fonction de contrôle de Java ES” à la page 16
- “Ordre d'installation conseillé” à la page 20

Présentation des composants Monitoring Framework et Monitoring Console

Sun Java System Monitoring Framework fournit l'infrastructure pour instrumenter des composants et exposer leurs attributs à des fins de contrôle. Elle définit une hiérarchie d'objets contrôlés appelée *Modèle de contrôle commun (CMM)*, basée sur la spécification du modèle d'information commune (CIM), qui est une norme industrielle. Les différents composants du produit exposent des objets qui représentent leurs attributs contrôlables et *l'agent de noeud* créé

une vue regroupant tous les composants d'un hôte donné. Le composant Monitoring Framework intègre également des mécanismes qui assurent la collecte de statistiques opérationnelles et le déclenchement d'alarmes en fonction des valeurs seuil définies par l'utilisateur.

Le composant Sun Java System Monitoring Console est l'interface graphique utilisée pour surveiller les composants de Java ES. Il comprend un *agent maître* qui se connecte à tous les agents de noeud au sein d'un déploiement Java ES. Monitoring Console est une application Web dont l'accès en tout point via HTTP est rendu possible par le composant Sun Java System Web Console. L'écran principal de cette application présente un récapitulatif de l'état de tous les composants activés et de toutes les alarmes déclenchées. Vous pouvez parcourir la hiérarchie des objets contrôlés pour chaque composant et afficher l'état détaillé, ainsi que les valeurs en temps réel, de tous les attributs contrôlés. L'interface de l'application Monitoring Console vous permet d'afficher le détail d'une alarme, d'en accuser réception ou de créer de nouvelles règles de contrôle basées sur l'attribut de votre choix.

Principe de fonctionnement de la fonction de contrôle de Java ES

Le fonction de contrôle fait référence au processus complet qui consiste à collecter les données d'exécution, les exposer et calculer des critères de qualité de service pour permettre à l'administrateur de pouvoir accéder aux performances du système et d'être informé du déclenchement d'alarmes. Pendant les opérations d'exécution, l'application Monitoring Console permet uniquement d'afficher les statistiques de performances, de créer des règles de contrôle automatique et d'accuser réception des alarmes. En revanche, elle constitue un précieux outil dans le cadre des opérations de configuration, de dépannage et de contrôle avancé car elle permet de mieux comprendre l'architecture du composant Monitoring Framework et ses interactions avec le composant Monitoring Console.

La fonction de contrôle de Java ES repose sur les principes suivants :

- Le modèle CMM (Modèle de contrôle commun) veille à ce que tous les composants de Java ES exposent des objets et des valeurs universels afin de disposer d'attributs comparables.
- Les objets Java définis par les interfaces CMM fournissent une instrumentation standardisée pour les composants du produit.
- Un agent de noeud expose tous les objets contrôlés pour l'ensemble des composants installés sur un système et assure la gestion des statistiques, des règles et des alarmes pour ces mêmes objets.
- Un agent maître résidant sur un hôte distinct regroupe tous les objets contrôlés à partir des différents agents de noeud et met les données résultantes à la disposition de l'application Monitoring Console.

La section suivante décrit plus en détail chacun de ces principes sur lesquels repose l'architecture de contrôle.

Modèle CMM (Modèle de contrôle commun)

Tout mécanisme de contrôle standardisé repose sur la définition des objets à contrôler et sur l'adoption de ces derniers par tous les composants contrôlés. Pour ce faire, l'architecture de contrôle définit le modèle CMM (Modèle de contrôle commun) comme l'extension du modèle d'information commune (CIM, Common Information Model) développé et géré par le consortium DMTF (Distributed Management Task Force). Le modèle CMM est à la fois un modèle d'information qui spécifie les objets contrôlés (tels que les ordinateurs, les applications, etc.) et un modèle de données qui définit des valeurs universelles (comme les valeurs d'état opérationnel). Dans son rôle de modèle d'information, le modèle CMM définit également les attributs d'un objet (par exemple, le nombre de requêtes traitées par un service) et les relations entre les objets (par exemple, le fait qu'un service soit hébergé sur un ordinateur donné).

Grâce au modèle CMM, les concepts comme les applications, les services, les points d'accès et autres, sont les mêmes pour tous les composants du produit et ce, même si l'implémentation sous-jacente diffère. Par exemple, le composant Web Server peut exposer un service qui gère les requêtes HTTP, tandis que le composant Directory Server expose un service de traitement des requêtes LDAP. L'objet standard permettra de capturer les données communes à ces deux fonctions, comme la possibilité de mesurer le nombre de requêtes traitées, le temps moyen de réponse à une requête sur un intervalle spécifique, etc.

De plus, certaines valeurs sont standardisées, ce qui permet de leur conférer un sens universel à l'échelle de tout le système. À titre d'exemple, l'état opérationnel ENDOMMAGÉ signifie toujours qu'un service reste disponible, mais que ses performances ont considérablement chuté, indépendamment du composant contrôlé.

La spécification CMM est intégrée aux interfaces Java et des classes sont utilisées pour l'instrumentation, comme décrit dans l'[Annexe A](#).

Instrumentation CMM

Dans Monitoring Framework, l'instrumentation désigne l'ensemble d'interfaces et de classes Java qui implémentent les définitions du modèle CMM. Pour les besoins de la nouvelle fonction de contrôle de Java ES, le code des composants du produit a été instrumenté afin de pouvoir instancier les objets CMM et exposer des valeurs d'exécution au travers des attributs des objets contrôlés. Les objets CMM implémentés par chaque composant déterminent les éléments contrôlables, ce qui explique que certains composants exposent moins d'attributs que d'autres. La liste des objets et des attributs associés exposés à des fins de contrôle par chaque composant est fournie dans l'[Annexe B](#).

Agents de noeud

Dans le contexte de la fonction de contrôle, un noeud désigne un hôte logique identifié par un nom de domaine complet unique ou une adresse IP. Il peut s'agir d'un système complet ou d'une zone Solaris configurée en tant que système virtuel. L'agent de noeud communique avec tous les composants instrumentés qui résident sur cet hôte et expose l'ensemble des objets contrôlés associés. Il peut également gérer l'ensemble de la structure logique pour collecter les statistiques de performances, surveiller les seuils définis dans les règles et générer des alarmes pour les objets contrôlés qu'il contient.

Le schéma ci-dessous représente le contenu d'un agent de noeud sur un seul hôte où existent des instances de trois composants de Java ES. Il explique également comment l'instrumentation est générée au niveau de l'agent de noeud pour exposer les valeurs transmises par les composants du produit.

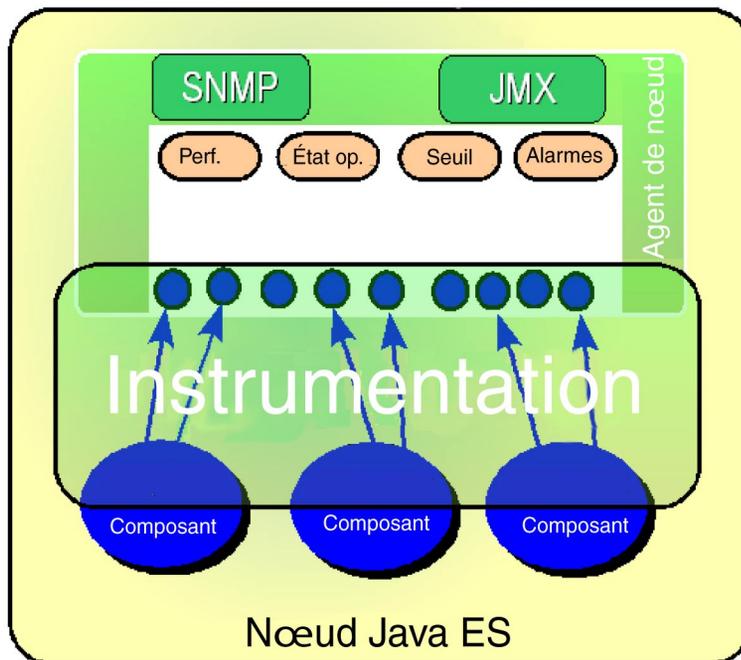


FIGURE 1-1 Schéma d'un agent de noeud

L'agent de noeud est implémenté sous forme de module dans le Conteneur d'agents communs, lequel est lui-même une machine virtuelle Java. L'implémentation de l'agent de noeud est basée sur la technologie JMX (Java Management Extensions), l'extension Java standard utilisée pour le contrôle et la gestion à distance. Toute application de contrôle compatible JMX qui comprend le modèle CMM peut accéder aux objets contrôlés contenus dans l'agent de noeud.

La fonction JMX permet également à l'agent de noeud d'exposer certains objets contrôlés via le protocole SNMP (Simple Network Monitoring Protocol).

Agent maître

L'agent maître est déployé sur une machine distincte dans le cadre de l'installation de l'application Monitoring Console. Il est configuré avec le nom ou l'adresse de tous les noeuds afin de pouvoir regrouper tous les objets contrôlés des différents agents de noeud. L'agent de noeud utilise également la technologie JMX pour communiquer avec les agents de noeud et il est aussi chargé dans son Conteneur d'agents communs local.

Le schéma suivant montre un agent maître connecté à deux noeuds. L'application Monitoring Console se connecte à l'agent maître pour contrôler les trois composants sur chaque noeud. Si vous souhaitez utiliser le protocole SNMP à des fins de contrôle, vous devez vous connecter à chaque noeud de façon individuelle car l'agent maître n'assure pas le regroupement des attributs SNMP. L'agent maître est destiné à être utilisé avec l'application Monitoring Console uniquement, c'est pourquoi il n'est pas accessible via d'autres applications de contrôle.

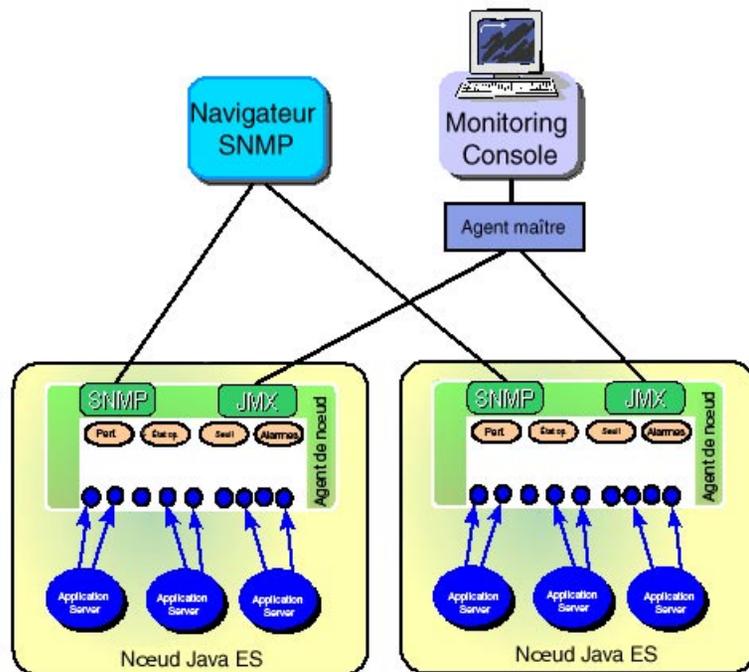


FIGURE 1-2 Schéma de l'architecture de contrôle globale

Ordre d'installation conseillé

Si vous souhaitez tester ou déployer la fonction de contrôle de Java ES, il est plus simple d'effectuer l'installation dans l'ordre suivant :

1. Installez et configurez tous les composants de votre déploiement conformément aux recommandations et instructions fournies dans le *Guide d'installation de Sun Java Enterprise System 5 pour UNIX*.
2. Activez et configurez le composant Monitoring Framework pour tous les composants contrôlés, comme décrit au [Chapitre 2](#).
3. Installez le composant Monitoring Console sur un hôte distinct, démarrez l'agent maître, puis le serveur Web, conformément aux instructions fournies au [Chapitre 3](#). Tous les composants contrôlés doivent être visibles et leur contrôle actif dans l'application Monitoring Console.

Remarque – En raison d'une incompatibilité qui existe entre l'agent de noeud et l'agent maître de cette version, le composant Monitoring Console doit être installé sur un hôte où aucun autre composant de Java ES ne réside. Pour plus de détails à ce sujet, reportez-vous à la section “[Dépannage du composant Monitoring Console](#)” à la page 70.

Toute modification apportée aux composants déployés après l'activation de la fonction de contrôle nécessite le redémarrage du conteneur de l'agent maître et du serveur Web utilisées par le composant Monitoring Console, comme expliqué à la section “[Dépannage du composant Monitoring Framework](#)” à la page 30.

Activation et configuration du composant Monitoring Framework

Comme décrit à la section [“Principe de fonctionnement de la fonction de contrôle de Java ES” à la page 16](#), le composant Monitoring Framework fournit l’instrumentation et l’agent de noeud nécessaire pour chaque composant contrôlé. De par sa nature, Monitoring Framework est donc un composant partagé qui est automatiquement installé chaque fois que vous procédez à l’installation d’un composant contrôlé à l’aide du programme d’installation de Java Enterprise System.

Cependant, pour plusieurs composants contrôlés, la fonction de contrôle n’est pas activée par défaut, tandis que d’autres nécessitent une configuration plus avancée pour devenir visibles dans l’agent de noeud. Suivez les procédures décrites dans ce chapitre pour chacun des composants du produit que vous avez installé.

Remarque – Il est conseillé d’installer et de configurer tous les composants du produit que vous envisagez d’exécuter sur un hôte donné avant d’effectuer une quelconque procédure présentée dans ce chapitre. Avant d’entreprendre toute installation ou configuration, consultez les *Notes de version de Sun Java Enterprise System 5 pour UNIX*.

Ces procédures utilisent la commande `mfwksetup` qui n’est généralement pas employée et qui, de ce fait, n’a pas été documentée.

Ce chapitre se compose des sections suivantes :

- “Organisation des répertoires installés” à la page 22
- “Utilisation du composant Monitoring Framework avec Access Manager” à la page 23
- “Utilisation du composant Monitoring Framework avec Application Server” à la page 26
- “Utilisation du composant Monitoring Framework avec Calendar Server” à la page 25
- “Utilisation du composant Monitoring Framework avec Directory Server” à la page 26
- “Utilisation du composant Monitoring Framework avec Instant Messaging” à la page 26
- “Utilisation du composant Monitoring Framework avec Messaging Server” à la page 27
- “Utilisation du composant Monitoring Framework avec Portal Server” à la page 28
- “Utilisation du composant Monitoring Framework avec Web Server” à la page 28

- “Configuration du Conteneur d’agents communs” à la page 29
- “Dépannage du composant Monitoring Framework” à la page 30
- “La commande `mfwkadm`” à la page 31

Organisation des répertoires installés

En tant que composant partagé, Monitoring Framework est automatiquement installé en cas de besoin. Pour connaître le nom du package installé sur votre système d’exploitation, reportez-vous au Chapitre 5, “Liste des packages pouvant être installés” du *Référence de l’installation de Sun Java Enterprise System 5 pour UNIX*. Le tableau ci-dessous répertorie les répertoires du package Monitoring Framework. Le répertoire d’installation par défaut *base-mfwk* fait référence au répertoire suivant, comme indiqué à la section “Chemins par défaut et noms de fichier” à la page 9 :

- Systèmes Solaris : `/opt/SUNWmfwk`
- Systèmes Linux : `/opt/sun/mfwk`

TABLEAU 2-1 Répertoires utilisés par le composant Monitoring Framework

Chemin	Description du contenu
<i>base-mfwk/config</i>	Modèle pour un fichier de configuration
Systèmes Solaris : <i>base-mfwk/lib</i> Systèmes Linux : <i>base-mfwk/share/lib</i>	Fichiers archive (.jar) Java
Systèmes Solaris : <i>base-mfwk/lib</i> Systèmes Linux : <i>base-mfwk/share/lib</i>	Fichiers de bibliothèque d’exécution 32 bits (.so)
Systèmes Solaris SPARC® : <i>base-mfwk/lib/sparcv9</i> Systèmes Solaris x86 : <i>base-mfwk/amd64</i> Systèmes Linux : <i>base-mfwk/lib64</i>	Fichiers de bibliothèque d’exécution 64 bits (.so)
<i>base-mfwk/bin</i>	Scripts publics et fichiers binaires privés
<i>base-mfwk/mib</i>	Versions texte des MIB SNMP prises en charge par le composant Monitoring Framework
<i>base-mfwk/xml</i>	Modèles de descripteur du Conteneur d’agents communs pour agent et agent maître (déployés par la commande <code>mfwksetup</code>)
<i>base-mfwk/dtd</i>	Fichiers DTD pour la fonction OSS/J
<i>/etc/base-mfwk/config</i>	Fichiers de configuration, y compris les fichiers liés à la sécurité

TABLEAU 2-1 Répertoires utilisés par le composant Monitoring Framework (Suite)

Chemin	Description du contenu
<code>/etc/base-mfwk/xml</code>	Descripteurs du Conteneur d'agents communs pour les agents et exemples
<code>/var/base-mfwk/logs</code>	Fichiers journaux du composant Monitoring Framework
<code>/var/base-mfwk/reports</code>	Répertoire de base pour les rapports sur les règles de contrôle
<code>/var/base-mfwk/alarms</code>	Référentiel des fichiers d'alarme

Utilisation du composant Monitoring Framework avec Access Manager

La fonction de contrôle est activée par défaut dans Access Manager, mais une limitation empêche les objets contrôlés d'apparaître dans l'application Monitoring Console.

Reportez-vous à la section [“Instrumentation du composant Access Manager”](#) à la page 77 pour obtenir la liste des objets et des attributs que vous pouvez contrôler.

▼ Pour activer la fonction de contrôle dans Access Manager

- 1 Désactivez temporairement la fonction de contrôle dans Access Manager en utilisant les commandes suivantes :

```
cacaoadm unregister-module com.sun.cmm.am.xml
cacaoadm restart
```

- 2 Ouvrez le fichier descripteur XML du composant Access Manager à des fins d'édition :

```
vi /etc/base-AccessMgr/config/com.sun.cmm.am.xml
```

- 3 Localisez les lignes contenant :

```
<param-name>Product Name</param-name>
<param-value>Access Manager</param-value>
```

et remplacez la seconde ligne par ce qui suit :

```
<param-value>Java ES Access Manager</param-value>
```

Enregistrez le fichier et fermez l'éditeur.

4 Enregistrez le module XML modifié :

```
base-mfwk/bin/mfwksetup -u /etc/base-AccessMgr/config/com.sun.cmm.am.xml  
base-mfwk/bin/mfwksetup -r /etc/base-AccessMgr/config/com.sun.cmm.am.xml
```

5 Redémarrez le Conteneur d'agents communs :

```
cacaoadm restart
```

**Erreurs
fréquentes**

En l'absence de test du comportement de la fonction de contrôle avec les conteneurs We de fabricants tiers, celle-ci est désactivée par défaut en cas de déploiement de Access Manager dans un environnement Websphere ou Weblogic. Vous pouvez cependant l'activer en suivant la procédure [“Pour désactiver et réactiver le contrôle sélectif des composants”](#) à la page 61, bien que cette configuration ne soit pas prise en charge.

Utilisation du composant Monitoring Framework avec Application Server

Reportez-vous à la section [“Instrumentation du composant Application Server”](#) à la page 77 pour obtenir la liste des objets et des attributs que vous pouvez contrôler.

▼ Pour activer la fonction de contrôle dans Application Server

1 Modifiez le fichier `/var/base-AppServer/domains/domain1/config/domain.xml` en changeant toutes les valeurs du paramètre `module-monitoring-level` de OFF en HIGH. Sinon :

- a. Connectez-vous à la console d'administration Application Server en tapant l'URL `https://nom-hôte:4849`.
- b. Sélectionnez Configurations, puis server-config (Admin Config).
- c. Définissez la valeur de contrôle sur HIGH.
- d. Définissez toutes les autres valeur sur HIGH.

2 Redémarrez Application Server en utilisant les commandes suivantes :

```
cd base-AppServer/appserv/bin  
asadmin stop-domain domain1  
asadmin start-domain user nom-utilisateur domain1
```

Tapez le mot de passe de `nom-utilisateur` lorsque vous y êtes invité.

- 3 Si vous avez déployé et contrôlé une instance de Portal Server avec Application Server, le processus de redémarrage du composant Application Server interfère avec le contrôle de Portal Server. Pour faire apparaître l'instance de Portal Server dans l'application Monitoring Console, vous devez afficher une page du portail dans un navigateur. Par exemple, chargez la page `http://portalserv.example.com:8080/portal` pour permettre le contrôle de `portalserv.example.com`.

Erreurs fréquentes

En raison d'une limitation, les objets contrôlés pour le composant Application Server sont supprimés de Monitoring Framework en cas de panne ou d'arrêt du composant Application Server. Dans ce cas, Application Server n'est plus visible dans l'application Monitoring Console et ne peut plus être contrôlé.

Utilisation du composant Monitoring Framework avec Calendar Server

Reportez-vous à la section [“Instrumentation du composant Calendar Server”](#) à la page 77 pour obtenir la liste des objets et des attributs que vous pouvez contrôler.

▼ Pour activer la fonction de contrôle dans Calendar Server

- 1 Modifiez le fichier `ics.conf` :

```
vi base-CalServ/cal/config/ics.conf
```

- 2 Ajoutez la ligne suivante :

```
local.mfagent.enable="yes"
```

- 3 Enregistrez le module XML Calendar Server :

```
base-mfwk/bin/mfwksetup -r /opt/SUNWics5/cal/lib/com.sun.cmm.cs.xml
```

- 4 Définissez la variable d'environnement `LD_LIBRARY_PATH` comme suit :

```
LD_LIBRARY_PATH=base-mfwk/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

- 5 Redémarrez le composant Calendar Server :

```
cd base-CalServ/cal/sbin/
./stop-cal
./start-cal
```

6 Redémarrez le Conteneur d'agents communs :

```
cacaoadm restart
```

Utilisation du composant Monitoring Framework avec Directory Server

Reportez-vous à la section [“Instrumentation du composant Directory Server”](#) à la page 78 pour obtenir la liste des objets et des attributs que vous pouvez contrôler.

▼ Pour activer la fonction de contrôle dans Directory Server

1 Créez un fichier de mots de passe temporaire :

```
echo -n mot-de-passe > /tmp/pwd
```

2 Activez le plug-in Monitoring en utilisant la commande suivante :

```
base-DirServ/ds6/bin/dscfg enable-plugin -e -p 389 -w /tmp/pwd "Monitoring Plugin"
```

3 Redémarrez le composant Directory Server:

```
cd base-DirServ/ds6/bin  
./dsadm restart /var/base-DirServ/DSinstance/
```

Utilisation du composant Monitoring Framework avec Instant Messaging

Reportez-vous à la section [“Instrumentation du composant Instant Messaging”](#) à la page 78 pour obtenir la liste des objets et des attributs que vous pouvez contrôler.

▼ Pour activer la fonction de contrôle dans Instant Messaging

1 Ouvrez le fichier descripteur XML du composant Instant Messaging à des fins d'édition :

```
vi /etc/base-IM/default/com.sun.cmm.im.xml
```

2 Changez la valeur du paramètre Install Location de *base-IM* en /etc/*base-IM*/default.

- 3 **Enregistrez le descripteur XML modifié du composant Instant Messaging :**
`base-mfwk/bin/mfwksetup -r /etc/base-IM/default/com.sun.cmm.im.xml`
- 4 **Activez l'instrumentation en ajoutant la ligne suivante dans le fichier**
`base-IM/config/iim.conf:`
`iim_server.monitor.enable = true`
- 5 **Redémarrez le composant Instant Messaging en utilisant les commandes suivantes :**
`cd base-IM/sbin`
`./imadmin stop`
`./imadmin start`
- 6 **Redémarrez le Conteneur d'agents communs :**
`cacaoadm restart`

Utilisation du composant Monitoring Framework avec Messaging Server

Reportez-vous à la section “[Instrumentation du composant Messaging Server](#)” à la page 78 pour obtenir la liste des objets et des attributs que vous pouvez contrôler.

▼ Pour activer la fonction de contrôle dans Messaging Server

- 1 **Activez l'instrumentation en utilisant la commande suivante :**
`base-MsgServ/sbin/configutil -o local.mfagent.enable -v 1`
- 2 **Enregistrez le module XML Messaging Server :**
`base-mfwk/bin/mfwksetup -r base-MsgServ/lib/com.sun.cmm.ms.xml`
- 3 **Redémarrez le composant Messaging Server :**
`cd base-MsgServ/sbin`
`./stop-msg`
`./start-msg`
- 4 **Redémarrez le Conteneur d'agents communs :**
`cacaoadm restart`

Utilisation du composant Monitoring Framework avec Portal Server

Reportez-vous à la section “[Instrumentation du composant Portal Server](#)” à la page 78 pour obtenir la liste des objets et des attributs que vous pouvez contrôler.

▼ Pour activer la fonction de contrôle dans Portal Server

- Pour activer le composant Portal Server, vous devez vous connecter à

`http://nom-hôte-complet:8080/portal/dt`

Cela compile le JSP du portail qui crée l'instance du portail prête pour le contrôle.

Erreurs fréquentes

À chaque redémarrage du composant Application Server qui héberge Portal Server, vous devez réactiver manuellement la fonction de contrôle à l'aide de cette procédure.

Utilisation du composant Monitoring Framework avec Web Server

Reportez-vous à la section “[Instrumentation du composant Web Server](#)” à la page 78 pour obtenir la liste des objets et des attributs que vous pouvez contrôler.

▼ Pour activer la fonction de contrôle dans Web Server

- 1 Démarrez Web Server en utilisant la commande suivante :

```
cd /var/base-WebServer/https-nom-hôte-complet/bin
./startserv
```

- 2 Démarrez le serveur d'administration :

```
cd /var/base-WebServer/admin-server/bin
./startserv
```

Configuration du Conteneur d'agents communs

Le Conteneur d'agents communs est un autre composant partagé qui est nécessaire au composant Monitoring Framework pour exécuter l'agent de noeud. Suivant la procédure d'installation utilisée, l'arrêt et le redémarrage du Conteneur d'agents communs peuvent être nécessaires. Par ailleurs, le Conteneur d'agents communs ayant été instrumenté, il peut lui aussi être contrôlé. Pour une description des objets contrôlés, reportez-vous à la section [“Instrumentation du Conteneur d'agents communs” à la page 77](#).

Pour vérifier si le Conteneur d'agents communs et l'agent de noeud ont déjà été lancés, exécutez la commande suivante :

```
cacaoadm status
```

Si le message suivant s'affiche, cela signifie que l'agent de noeud est déjà en cours d'exécution :

```
default instance is DISABLED at system startup.
Smf monitoring process:
26996
Uptime: 0 day(s), 0:57
```

En revanche, si le message suivant s'affiche, cela signifie que l'agent de noeud n'est pas en cours d'exécution :

```
default instance is DISABLED at system startup.
default instance is not running.
```

▼ Pour activer la fonction de contrôle dans le Conteneur d'agents communs

Le Conteneur d'agents communs est un composant partagé qui a été instrumenté pour prendre en charge la fonction de contrôle. Comme décrit à la section [“Agents de noeud” à la page 18](#), tous les composants de Java ES qui ont été installés sur un même hôte ou dans la même zone partagent le Conteneur d'agents communs et l'agent de noeud. Exécutez la procédure ci-dessous en tant que superutilisateur sur chaque hôte logique de votre déploiement où vous souhaitez contrôler le Conteneur d'agents communs.

- 1 **Si le Conteneur d'agents communs est en cours d'exécution, arrêtez-le en utilisant la commande suivante :**

```
cacaoadm stop
```

- 2 **Activez l'instrumentation du conteneur proprement dit :**

```
cacaoadm set-param enable-instrumentation=true
```

- 3 Assurez-vous d'avoir correctement défini la valeur du paramètre que vous venez de définir, puis redémarrez le Conteneur d'agents communs :**

```
cacaoadm get-param enable-instrumentation
cacaoadm start
```

- 4 Créez un mot de passe key :**

```
echo -n mot-de-passe > /etc/base-mfwk/config/security/password.cacao
```

- 5 Générez votre clé :**

```
base-mfwk/bin/cpgenkey -n cacao -p /etc/base-mfwk/config/security/password.cacao
```

- 6 Enregistrez les modules de contrôle spécifiques au Conteneur d'agents communs :**

```
cacaoadm register-module /usr/lib/cacao/ext/instrum/config/com.sun.cacao.instrum.xml
cacaoadm register-module /usr/lib/cacao/ext/instrum_jesmfm/config/com.sun.cacao.instrum.jesmfm.xml
cacaoadm register-module /usr/lib/cacao/ext/instrum_jesmfm/config/com.sun.cacao.cmm.xml
```

Dépannage du composant Monitoring Framework

Consultez également les problèmes connus répertoriés dans les *Notes de version de Sun Java Enterprise System 5 pour UNIX*.

Utilisation du composant Monitoring Framework sur les plates-formes HP-UX

Sur les plates-formes HP-UX, la machine virtuelle Java (JVM) n'est pas configurée par défaut pour le traitement d'un nombre important de tâches requis par le composant Monitoring Framework, ce qui peut entraîner des exceptions de mémoire saturée (OutOfMemory). Pour configurer la machine virtuelle Java, téléchargez et exécutez l'outil HPjconfig à partir de l'URL suivant : http://h21007.www2.hp.com/dspp/tech/tech_TechDocumentDetailPage_IDX/1,1701,1620,00.html.

Utilisation du composant Monitoring Framework sous Microsoft Windows

Le contrôle des composants de Java ES sur la plate-forme Windows via le composant Monitoring Framework est totalement pris en charge, même s'il présente quelques différences. Par exemple, vous devez effectuer la mise à jour vers Java 1.5 ou ultérieur pour ne pas rencontrer certains problèmes connus. Pour obtenir la liste des autres problèmes connus, consultez les *Sun Java Enterprise System 5 Release Notes for Microsoft Windows*.

▼ Pour redémarrer l'agent de noeud

Si vous devez redémarrer le Conteneur d'agents communs qui contient un agent de noeud, les composants contrôlés par le biais de cet agent ne seront pas visibles dans l'application Monitoring Console si vous n'effectuez pas la procédure suivante :

1 Redémarrez le Conteneur d'agents communs qui contient l'agent de *noeud* :

```
cacaoadm restart
```

2 Redémarrez le Conteneur d'agents communs qui contient l'agent *maître*. L'agent maître s'exécute dans le composant Monitoring Framework sur l'hôte ou dans la zone où vous avez installé le composant Monitoring Console.

```
cacaoadm restart
```

L'agent maître se reconnecte automatiquement à tous les agents de noeud dont il assurait précédemment le contrôle.

3 Redémarrez le serveur Web sur lequel le composant Monitoring Console est installé :

```
/usr/sbin/smcwebserver restart
```

La commande `mfwkadm`

Cette section présente la page de manuel disponible pour la commande `mfwkadm`, qui constitue un utilitaire de maintenance. Les informations ci-dessous sont accessibles dans la section 1M des pages de manuel. Utilisez cette commande pour gérer le contenu d'un agent de noeud, notamment l'ensemble des modules des composants contrôlés et les règles de contrôle, également appelées « tâches », que vous avez définies pour ce noeud. Certains des termes et descriptions de la page de manuel ont été modifiés pour refléter la terminologie utilisée dans le présent manuel.

Descriptif

```
mfwkadm --help
```

```
mfwkadm start
```

```
mfwkadm stop
```

```
mfwkadm restart
```

```
mfwkadm list-params
```

```
mfwkadm list-modules
```

```
mfwkadm info instance-exécutée
```

Contrôle des performances

mfwkadm pm-job observable-classes

mfwkadm pm-job observable-objects [class=*classe-objets*] [domain=*domaine-objet*]

mfwkadm pm-job observable-attributes class=*classe-objets*

mfwkadm pm-job list

mfwkadm pm-job info *nom-tâche*

mfwkadm pm-job create *nom-tâche* granularity=*valeur-entière* object=*nom-objet*
[object=*nom-objet* ...]

mfwkadm pm-job delete *nom-tâche*

mfwkadm pm-job suspend *nom-tâche*

mfwkadm pm-job resume *nom-tâche*

Contrôle de l'état opérationnel

mfwkadm opstat-job observable-classes

mfwkadm opstat-job observable-objects [class=*classe-objets*] [domain=*domaine-objet*]

mfwkadm opstat-job observable-attributes class=*classe-objets*

mfwkadm opstat-job list

mfwkadm opstat-job info *nom-tâche*

mfwkadm opstat-job create *nom-tâche* granularity=*valeur-entière* object=*nom-objet*
[object=*nom-objet* ...]

mfwkadm opstat-job delete *nom-tâche*

mfwkadm opstat-job suspend *nom-tâche*

mfwkadm opstat-job resume *nom-tâche*

Contrôle des valeurs seuil

mfwkadm thrsh-job observable-classes

mfwkadm thrsh-job observable-objects [class=*classe-objets*] [domain=*domaine-objet*]

mfwkadm thrsh-job observable-attributes class=*classe-objets*

mfwkadm thrsh-job list

mfwkadm thrsh-job info *nom-tâche*

mfwkadm thrsh-job create *nom-tâche* granularity=*valeur-entière*
 attributeName=*nom-attribut* attributeType=*type-attribut* thresholdValue=*valeur-seuil*
 thresholdOffset=*valeur-décalage* thresholdDirection=[ÉLÉVATION | CHUTE]
 object=*nom-objet*

mfwkadm thrsh-job delete *nom-tâche*

mfwkadm thrsh-job suspend *nom-tâche*

mfwkadm thrsh-job resume *nom-tâche*

Description

L'utilitaire mfwkadm est l'interface de ligne de commande que vous pouvez utiliser pour gérer l'agent Monitoring Framework, également appelé « agent de noeud ». L'agent de noeud s'exécute dans le Conteneur d'agents communs. Vous pouvez utiliser la commande mfwkadm pour arrêter et redémarrer l'agent de noeud ou pour gérer les tâches de contrôle que celui-ci exécute. Elle doit être exécutée sur le même hôte que celui de l'agent de noeud. L'ordre des arguments de cette commande présenté dans cette section doit être respecté.

Pour changer la langue dans laquelle s'affichent les messages de sortie, définissez la variable d'environnement LC_MESSAGE sur votre *langue*. La commande mfwkadm utilise les messages stockés dans le fichier JesmfMessages_ *langue* .pm qui se trouve dans le répertoire lib/resources. Si aucun fichier de messages n'existe pour votre langue, ou si aucune langue n'est spécifiée, la commande mfwkadm utilise le groupe de messages du fichier JesmfMessages .pm.

L'utilitaire mfwkadm est associé aux sous-commandes suivantes. Celles marquées d'un astérisque (*) nécessite l'exécution du Conteneur d'agents communs et le chargement de l'agent de noeud :

- start
- stop
- restart
- list-params (*)
- list-modules (*)
- info (*)
- pm-job (*)
- opstat-job (*)
- thrsh-job (*)

Suivant le nombre de modules du Conteneur d'agents communs à charger, un délai de quelques secondes à plusieurs minutes peut être observé entre le démarrage de l'agent de noeud et la disponibilité de l'utilitaire mfwkadm. Au cours de cette période, les commandes exécutées échouent et un message explicite s'affiche.

Options

Les options suivantes sont prises en charge.

`--help` Affiche le résumé des utilisations.

Sous-commandes

`start`

Cette sous-commande démarre l'agent de noeud du composant Monitoring Framework, ainsi que les modules des composants associés, sans arrêter le Conteneur d'agents communs.

Elle procède au déploiement de l'agent de noeud, puis à celui des modules des composants associés dans le Conteneur d'agents communs. Il s'agit d'un utilitaire wrapper qui prévaut sur les sous-commandes `lock` et `unlock` de l'utilitaire `cacaoadm`.

La sous-commande `start` se limite à démarrer l'agent de noeud et les modules des composants de Java ES associés au composant Monitoring Framework. Les modules des composants ont le préfixe `com.sun.cmm`.

Sécurité : la sous-commande `start` peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs. Si un autre utilisateur l'exécute, le message suivant s'affiche :

```
Error occured in mfwkadm
Problem running /usr/sbin/cacaoadm unlock com.sun.mfwk 2>&1.
Stdout/Stderr: This command must be run by user: [root].
```

`stop`

Cette sous-commande arrête l'agent de noeud Monitoring Framework et les modules des composants de Java ES associés dans le Conteneur d'agents communs.

Elle arrête tout d'abord les modules des composants de Java ES déployés dans le Conteneur d'agents communs, puis l'agent de noeud. Il s'agit d'un utilitaire wrapper qui prévaut sur les sous-commandes `lock` et `unlock` de l'utilitaire `cacaoadm`.

La sous-commande `stop` arrête uniquement les modules des composants de Java ES qui sont associés au composant Monitoring Framework, puis l'agent de noeud proprement dit. Les modules des composants ont le préfixe `com.sun.cmm`.

Sécurité : la sous-commande `stop` peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs. Si un autre utilisateur l'exécute, le message suivant s'affiche :

```
Error occured in mfwkadm
Problem running /usr/sbin/cacaoadm unlock com.sun.mfwk 2>&1.
Stdout/Stderr: This command must be run by user: [root].
```

restart

Cette sous-commande redémarre l'agent de noeud Monitoring Framework et les modules des composants de Java ES dans le Conteneur d'agents communs.

Elle tente d'arrêter, puis de redémarrer l'agent de noeud et les modules qui lui sont associés dans le Conteneur d'agents communs de la même façon que les sous-commandes `stop` et `start`.

Sécurité : la sous-commande `restart` peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs. Si un autre utilisateur l'exécute, le message suivant s'affiche :

```
Error occured in mfwkadm
Problem running //usr/sbin/cacaoadm unlock com.sun.mfwk 2>&1.
Stdout/Stderr: This command must be run by user: [root].
```

list-params

Cette sous-commande affiche la liste de tous les paramètres de configuration associés à l'agent de noeud Monitoring Framework.

Sécurité : il n'existe aucune restriction d'utilisation pour cette commande.

list-modules

Cette sous-commande affiche la liste des modules des composants qui implémentent le modèle de contrôle commun (CMM) et qui sont chargés dans le Conteneur d'agents communs. Elle fournit également la liste de toutes les instances en cours d'exécution pour chaque composant de Java ES installé. Pour chaque composant, zéro, une ou plusieurs instances peuvent être en cours d'exécution.

Sécurité : pour les utilisateurs autres que celui qui a démarré le Conteneur d'agents communs, la liste des composants de Java ES installés ne spécifie pas les instances en cours d'exécution des composants.

info *instance-exécutée*

Cette sous-commande affiche des informations sur l'*instance-exécutée* désignée. *instance-exécutée* doit correspondre à une instance en cours d'exécution qui figure dans la liste affichée par la sous-commande `list-modules`. Les informations affichées incluent :

- Pour chaque type de tâche de contrôle, tous les objets contrôlables associés à l'instance en cours d'exécution, triés par nom de classe. Les objets contrôlables sont ceux pour lesquels vous pouvez créer des tâches de contrôle des performances, des tâches de contrôle de l'état opérationnel ou encore des tâches de contrôle de seuil en utilisant respectivement les sous-commandes `pm-job`, `opstat-job` ou `thrsh-job`.
- Pour chaque classe d'objets contrôlables, tous les objets contrôlables associés, avec leur nom et type indiqués.

Sécurité : pour les utilisateurs autres que celui qui a démarré le Conteneur d'agents communs, aucune information n'est affichée.

Contrôle des performances

pm- job observable-classes

Cette sous-commande affiche la liste de toutes les classes d'objets actuellement contrôlables pour lesquelles il est possible de créer des tâches de contrôle des performances.

pm- job observable-objects [class=*classe-objets*] [domain=*domaine-objets*]

Cette sous-commande affiche la liste de tous les objets actuellement contrôlables pour lesquels il est possible de créer des tâches de contrôle des performances. Par défaut, tous les objets de toutes les classes contrôlables et de tous les domaines sont affichés. La liste des objets est triée par nom de classe.

class=*classe-objets*

Si vous spécifiez le paramètre *classe-objets* optionnel, cela permet de limiter la sortie aux objets contrôlables de la classe désignée. La valeur de *classe-objets* doit correspondre à l'une des classes qui figure dans la liste affichée par la sous-commande pm- job observable-classes.

domain=*domaine-objets*

Si vous spécifiez le paramètre *domaine-objets* optionnel, cela permet de limiter la sortie aux objets contrôlables du domaine désigné. Le domaine d'un objet correspond à la chaîne qui précède les deux-points (:) dans le nom de cet objet.

pm- job observable-attributes class=*classe-objets*

Cette sous-commande affiche la liste de tous les objets contrôlables dans la *classe-objets* spécifiée. Les attributs sont affichés avec leur nom et type. La valeur de *classe-objets* doit correspondre à l'une des classes qui prend en charge les tâches de contrôle des performances, telles que listées par la sous-commande pm- job observable-classes.

pm- job list

Cette sous-commande affiche la liste de toutes les tâches de contrôle des performances actuellement définies. Les tâches sont affichées pour chacun des objets auquel est associé une tâche de contrôle des performances. De plus, les objets sont triés par nom de classe. Les informations affichées pour chaque tâche sont identiques à celles obtenues avec la sous-commande pm- job info.

Sécurité : pour les utilisateurs autres que celui qui a démarré le Conteneur d'agents communs, aucune tâche n'est affichée.

`pm-job info nom-tâche`

Cette sous-commande affiche les informations détaillées relatives à la tâche de contrôle des performances *nom-tâche*. La valeur de *nom-tâche* doit correspondre à l'une des tâches affichées par la sous-commande `pm-job list`. Les informations affichées par cette sous-commande sont les suivantes :

- Le nom de la tâche de contrôle des performances.
- Le type de la tâche de contrôle des performances, à savoir « par objet » ou par « par classe ». Les tâches de type par objet contrôlent une ou plusieurs instances de l'objet spécifié, tandis que les tâches de type par classe surveillent chaque instance d'une classe d'objets. Notez que la commande `mfwkadm` ne permet pas de créer des tâches de type par classe.
- L'état de la tâche de contrôle des performances : active en fonctionnement, active hors fonctionnement ou suspendue. Une tâche active en fonctionnement est actuellement planifiée pour être exécutée et collecte des données. Une tâche active hors fonctionnement est en cours d'exécution, mais ne collecte aucune donnée car l'heure courante se situe hors de la planification de traitement. Une tâche suspendue n'est pas exécutée et ne collecte aucune donnée. Utilisez les sous-commandes `pm-job suspend` et `pm-job resume` pour changer l'état d'exécution d'une tâche de contrôle des performances.
- La granularité, exprimée en secondes, de la tâche de contrôle des performances. Il s'agit de l'intervalle défini pour la collecte des données effectuée par cette tâche.
- La période de compte rendu définie pour la tâche de contrôle. La période de compte rendu multipliée par la granularité donne la fréquence de notification. Par exemple, si la période de granularité est définie sur 10 secondes et la période de compte rendu sur 6, un compte rendu de tâche par événement collectera les données toutes les 10 secondes et enverra une notification comprenant 6 comptes rendus toutes les 60 secondes (10*6). Si la tâche effectue également un compte rendu par fichier, elle enverra un événement toutes les 60 secondes avec l'emplacement des 6 fichiers générés.
- Si la tâche de contrôle des performances effectue un compte rendu par événement. Cela signifie que les résultats de la tâche de contrôle des performances sont envoyés sous forme de notifications à un client enregistré.
- Si la tâche de contrôle des performances effectue un compte rendu par fichier. Cela signifie que les comptes rendus de la tâche de contrôle des performances sont écrits dans des fichiers locaux et que les notifications contenant les noms de fichiers sont envoyées aux clients enregistrés.
- Le format du compte rendu de la tâche de contrôle des performances, qui est toujours XML.

- La planification de la tâche de contrôle des performances. La planification définit les jours et les heures auxquels la tâche est active en fonctionnement ou active hors fonctionnement (et collecte ou non des données, respectivement). Puis, pour une tâche de type par objet :
- La liste des objets contrôlés, triée par nom.
- Si seul un sous-ensemble d'attributs contrôlables est spécifié, les attributs contrôlés des objets contrôlés sont listés par nom et par type. Et, pour une tâche de type par classe :
- La liste des tâches contrôlés, triée par nom.
- Si seul un sous-ensemble d'attributs contrôlables est spécifié, les attributs contrôlés des classes contrôlées sont listés par nom et par type. Ces attributs sont communs à toutes les classes.

Sécurité : pour les utilisateurs autres que celui qui a démarré le Conteneur d'agents communs, aucune information n'est affichée.

`pm-job create nom-tâche granularity= valeur-entière object=nom-objet [object=nom-objet ...]`
 Cette sous-commande crée une nouvelle tâche de contrôle des performances d'un ou plusieurs objets. La commande `mfwkadm` ne permet pas de créer des tâches par classe. Lors de la création de tâches de contrôle des performances, les paramètres suivants peuvent être définis :

nom-tâche

Chaîne qui identifie de façon unique la tâche de contrôle des performances. Le paramètre *nom-tâche* ne peut pas être déjà utilisé par une autre tâche quelconque de contrôle des performances.

granularity=valeur-entière

Intervalle exprimé en secondes qui sépare le début de deux collectes successives de données de mesure alors que la tâche est active en fonctionnement. La période de granularité peut être définie sur 300 secondes (5 minutes), 900 secondes (15 minutes), 1800 secondes (toutes les demi-heures), 3600 secondes (toutes les heures). Une période de granularité de 300 secondes est convient dans la plupart des cas. Pour certaines mesures, l'utilisation de périodes de granularité plus longues pour la collecte des données peut cependant être plus significative.

object=nom-objet [object=nom-objet ...]

Un ou plusieurs objets contrôlables pour lesquels la tâche de contrôle des performances doit collecter des données et fournir un compte rendu. La valeur de *nom-objet* doit correspondre à l'un des noms affichés par les sous-commandes `pm-job list` ou `pm-job observable-objects`. Si vous spécifiez plusieurs paramètres *object= nom-objet*, une seule tâche de contrôle des performances est créée pour plusieurs objets.

Sécurité : cette sous-commande peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs.

pm- job delete *nom-tâche*

Cette sous-commande supprime la tâche de contrôle des performances *nom-tâche*. La valeur de *nom-tâche* doit correspondre à l'une des tâches affichées par la sous-commande `pm- job list`.

Sécurité : cette sous-commande peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs.

pm- job suspend *nom-tâche*

Cette sous-commande suspend la tâche de contrôle des performances *nom-tâche*. Une tâche suspendue n'est pas active et ne collecte aucune donnée, quelle que soit sa planification. Elle reste toutefois définie et peut être réactivée via l'utilisation de la sous-commande `pm- job resume`. La valeur de *nom-tâche* doit correspondre à l'une des tâches affichées par la sous-commande `pm- job list`.

Sécurité : cette sous-commande peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs.

pm- job resume *nom-tâche*

Cette sous-commande reprend l'exécution de la tâche de contrôle des performances *nom-tâche*. Une tâche dont l'exécution est reprise commencera à collecter des données et à envoyer des comptes rendus en fonction de sa planification. La valeur de *nom-tâche* doit correspondre à l'une des tâches affichées par la sous-commande `pm- job list`. Cette sous-commande est l'homologue de la sous-commande `pm- job suspend`.

Sécurité : cette sous-commande peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs.

Contrôle de l'état opérationnel

opstat- job observable-classes

Cette sous-commande affiche la liste de toutes les classes d'objets actuellement contrôlables pour lesquelles il est possible de créer des tâches de contrôle de l'état opérationnel.

opstat- job observable-objects [*class=nom-classe*] [*domain=domaine-objets*]

Cette sous-commande affiche la liste de tous les objets actuellement contrôlables pour lesquels il est possible de créer des tâches de contrôle de l'état opérationnel. Par défaut, tous les objets de toutes les classes contrôlables et de tous les domaines sont affichés. La liste des objets est triée par nom de classe.

class=classe-objets

Si vous spécifiez le paramètre *classe-objets* optionnel, cela permet de limiter la sortie aux objets contrôlables de la classe désignée. La valeur de *classe-objets* doit correspondre à l'une des classes qui figurent dans la liste affichée par la sous-commande `opstat- job observable-classes`.

domain=*domaine-objets*

Si vous spécifiez le paramètre *domaine-objets* optionnel, cela permet de limiter la sortie aux objets contrôlables du domaine désigné. Le domaine d'un objet correspond à la chaîne qui précède les deux-points (:) dans le nom de cet objet.

opstat-job observable-attributes class= *classe-objets*

Cette sous-commande affiche la liste de tous les objets contrôlables dans la *classe-objets* spécifiée. Les attributs sont affichés avec leur nom et type. La valeur de *classe-objets* doit correspondre à l'une des classes qui figurent dans la liste affichée par la sous-commande opstat-job observable-classes.

opstat-job list

Cette sous-commande affiche la liste de toutes les tâches de contrôle de l'état opération actuellement définies. Les tâches sont affichées pour chacun des objets auquel est associé une tâche de contrôle de l'état opérationnel. De plus, les objets sont triés par nom de classe. Les informations affichées pour chaque tâche sont identiques à celles obtenues avec la sous-commande opstat-job info.

Sécurité : pour les utilisateurs autres que celui qui a démarré le Conteneur d'agents communs, aucune tâche n'est affichée.

opstat-job info *nom-tâche*

Cette sous-commande affiche les informations détaillées relatives à la tâche de contrôle de l'état opérationnel *nom-tâche*. La valeur de *nom-tâche* doit correspondre à l'une des tâches affichées par la sous-commande opstat-job list. Les informations affichées par cette sous-commande sont les suivantes :

- Le nom de la tâche de contrôle de l'état opérationnel.
- Le type de la tâche de contrôle de l'état opérationnel, à savoir « par objet » ou par « par classe ». Les tâches de type par objet contrôlent une instance de l'objet spécifié, tandis que les tâches de type par classe surveillent chaque instance d'une classe d'objets. Notez que la commande mfwkadm ne permet pas de créer des tâches de type par classe.
- L'état de la tâche de contrôle de l'état opérationnel : active en fonctionnement, active hors fonctionnement ou suspendue. Une tâche active en fonctionnement est actuellement planifiée pour être exécutée et collecte des données. Une tâche active hors fonctionnement est en cours d'exécution, mais ne collecte pas de données car l'heure courante est en dehors de la planification de traitement. Une tâche suspendue n'est pas exécutée et ne collecte aucune donnée. Utilisez les sous-commandes opstat-job suspend et opstat-job resume pour changer l'état d'exécution d'une tâche de contrôle de l'état opérationnel.
- La granularité, exprimée en secondes, de la tâche de contrôle de l'état opérationnel. Il s'agit de l'intervalle défini pour la collecte des données effectuée par cette tâche.
- Si la tâche de contrôle de l'état opérationnel effectue un compte rendu par événement. Cela signifie que les résultats de la tâche de contrôle de l'état opérationnel sont envoyés sous forme de notifications à un client enregistré.

- Si la tâche de contrôle de l'état opérationnel effectue un compte rendu par fichier. Cela signifie que les comptes rendus de la tâche de contrôle de l'état opérationnel sont écrits dans des fichiers locaux et que les notifications contenant les noms de fichiers sont envoyées aux clients enregistrés.
- Le format du compte rendu de la tâche de contrôle de l'état opérationnel, qui est toujours XML.
- La planification de la tâche de contrôle de l'état opérationnel. La planification définit les jours et les heures auxquels la tâche est active en fonctionnement ou active hors fonctionnement (et collecte ou non des données, respectivement).
- Pour une tâche de type par objet, la liste des objets contrôlés, triée par nom.
- Pour une tâche de type par classe, la liste des classes contrôlées, triée par nom.

Sécurité : pour les utilisateurs autres que celui qui a démarré le Conteneur d'agents communs, aucune information n'est affichée.

`opstat - job create nom-tâche granularity= valeur-entière object=nom-objet
[object=nom-objet ...]`

Cette sous-commande crée une nouvelle tâche de contrôle de l'état opérationnel d'un ou plusieurs objets. La commande `mfwkadm` ne permet pas de créer des tâches par classe. Lors de la création de tâches de contrôle des performances, les paramètres suivants peuvent être définis :

nom-tâche

Chaîne qui identifie de façon unique la tâche de contrôle de l'état opérationnel. Le paramètre *nom-tâche* ne peut pas être déjà utilisé par une autre tâche quelconque de contrôle de l'état opérationnel.

granularity=*valeur-entière*

Intervalle exprimé en secondes qui sépare le début de deux collectes successives de données de mesure alors que la tâche est active en fonctionnement.

object=*nom-objet* [object=*nom-objet* ...]

Un ou plusieurs objets contrôlables pour lesquels la tâche de contrôle de l'état opérationnel doit collecter des données et fournir un compte rendu. La valeur de *nom-objet* doit correspondre à l'un des noms affichés par les sous-commandes `opstat - job list` ou `opstat - job observable-objects`. Si vous spécifiez plusieurs paramètres *object=nom-objet*, une seule tâche de contrôle de l'état opérationnel est créée pour plusieurs objets.

Sécurité : cette sous-commande peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs.

`opstat - job delete nom-tâche`

Cette sous-commande supprime la tâche de contrôle de l'état opérationnel *nom-tâche*. La valeur de *nom-tâche* doit correspondre à l'une des tâches affichées par la sous-commande `opstat - job list`.

Sécurité : cette sous-commande peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs.

opstat - job suspend *nom-tâche*

Cette sous-commande suspend la tâche de contrôle de l'état opérationnel *nom-tâche*. Une tâche suspendue n'est pas active et ne collecte aucune donnée, quelle que soit sa planification. Elle reste toutefois définie et peut être réactivée via l'utilisation de la sous-commande opstat - job resume. La valeur de *nom-tâche* doit correspondre à l'une des tâches affichées par la sous-commande opstat - job list.

Sécurité : cette sous-commande peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs.

opstat - job resume *nom-tâche*

Cette sous-commande reprend l'exécution la tâche de contrôle de l'état opérationnel *nom-tâche*. Une tâche dont l'exécution est reprise commencera à collecter des données et à envoyer des comptes rendus en fonction de sa planification. La valeur de *nom-tâche* doit correspondre à l'une des tâches affichées par la sous-commande opstat - job list. Cette sous-commande est l'homologue de la sous-commande opstat - job suspend.

Sécurité : cette sous-commande peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs.

Contrôle des valeurs seuil

thrsh - job observable - classes

Cette sous-commande affiche la liste de toutes les classes d'objets actuellement contrôlables pour lesquelles il est possible de créer des tâches de contrôle des valeurs seuil.

thrsh - job observable - objects [class=*classe-objets*] [domain=*domaine-objets*]

Cette sous-commande affiche la liste de tous les objets actuellement contrôlables pour lesquels il est possible de créer des tâches de contrôle des valeurs seuil. Par défaut, tous les objets de toutes les classes contrôlables et de tous les domaines sont affichés. La liste des objets est triée par nom de classe.

class=*classe-objets*

Si vous spécifiez le paramètre *classe-objets* optionnel, cela permet de limiter la sortie aux objets contrôlables de la classe désignée. La valeur de *classe-objets* doit correspondre à l'une des classes qui figurent dans la liste affichée par la sous-commande thrsh - job observable - classes.

domain=*domaine-objets*

Si vous spécifiez le paramètre *domaine-objets* optionnel, cela permet de limiter la sortie aux objets contrôlables de le domaine désigné. Le domaine d'un objet correspond à la chaîne qui précède les deux-points (:) dans le nom de cet objet.

`thrsh-job observable-attributes class=classe-objets`

Cette sous-commande affiche la liste de tous les objets contrôlables dans la *classe-objets* spécifiée. Les attributs sont affichés avec leur nom et type. La valeur de *classe-objets* doit correspondre à l'une des classes qui figurent dans la liste affichée par la sous-commande `thrsh-job observable-classes`.

`thrsh-job list`

Cette sous-commande affiche la liste de toutes les tâches de contrôle des valeurs seuil actuellement définies. Les tâches sont affichées pour chacun des objets auquel est associé une tâche de contrôle des valeurs seuil. De plus, les objets sont triés par nom de classe. Les informations affichées pour chaque tâche sont identiques à celles obtenues avec la sous-commande `thrsh-job info`.

Sécurité : pour les utilisateurs autres que celui qui a démarré le Conteneur d'agents communs, aucune tâche n'est affichée.

`thrsh-job info nom-tâche`

Cette sous-commande affiche les informations détaillées relatives à la tâche de contrôle des valeurs seuil *nom-tâche*. La valeur de *nom-tâche* doit correspondre à l'une des tâches affichées par la sous-commande `thrsh-job list`. Les informations affichées par cette sous-commande sont les suivantes :

- Le nom de la tâche de contrôle des valeurs seuil.
- La multiplicité de la tâche de contrôle des valeurs seuil. Cette version prend uniquement en charge des tâches de contrôle des valeurs seuil simples qui surveillent un attribut d'un seul objet.
- L'état de la tâche de contrôle des valeurs seuil : active en fonctionnement, active hors fonctionnement ou suspendue. Une tâche active en fonctionnement est actuellement planifiée pour être exécutée et collecte des données. Une tâche active hors fonctionnement est en cours d'exécution, mais ne collecte pas de données car l'heure courante est en dehors de la planification de traitement. Une tâche suspendue n'est pas exécutée et ne collecte aucune donnée. Utilisez les sous-commandes `thrsh-job suspend` et `thrsh-job resume` pour changer l'état d'exécution d'une tâche de contrôle des valeurs seuil.
- La granularité, exprimée en secondes, de la tâche de contrôle des valeurs seuil. Il s'agit de l'intervalle défini pour la collecte des données effectuée par cette tâche.
- La planification de la tâche de contrôle des valeurs seuil. La planification définit les jours et les heures auxquels la tâche est active en fonctionnement ou active hors fonctionnement (et collecte ou non des données, respectivement).
- La configuration de l'alarme pour la tâche de contrôle des valeurs seuil. Il s'agit de l'alarme qui sera déclenchée si la valeur observée de l'attribut contrôlé dépasse la valeur seuil définie. Le type et la gravité de l'alarme sont également affichés.
- L'objet contrôlé par la tâche de contrôle des valeurs seuil.
- Le nom de l'attribut auquel le seuil s'applique.

- La valeur du seuil sur lequel est basé le déclenchement de l'alarme.
- La direction de la progression de la valeur pour laquelle une alarme sera déclenchée en cas de dépassement du seuil, à savoir ÉLEVATION ou CHUTE.
- Le décalage de tolérance pour le seuil. Lorsque la direction définie est ÉLEVATION, l'alarme est redéclenchée lorsque la valeur de l'attribut contrôlé est inférieure à *valeur-seuil-valeur-décalage*. Lorsque la direction définie est CHUTE, l'alarme est redéclenchée lorsque la valeur de l'attribut contrôlé est supérieure à *valeur-seuil+valeur-décalage*. Ce comportement s'applique même si la valeur du décalage est égale à zéro.

Sécurité : pour les utilisateurs autres que celui qui a démarré le Conteneur d'agents communs, aucune information n'est affichée.

```
thrsh-job create nom-tâche object=nom-objet granularity=valeur-entière
attributeName=nom-attribut attributeType=type-attribut thresholdValue=valeur-seuil
thresholdOffset=valeur-décalage thresholdDirection=[ÉLEVATION|CHUTE]
```

Cette sous-commande crée une nouvelle tâche de contrôle des valeurs seuil qui surveille un attribut d'un seul objet. Lors de la création des tâches de contrôle des valeurs seuil, les paramètres suivants peuvent être définis :

nom-tâche

Chaîne qui identifie de façon unique la tâche de contrôle des valeurs seuil. Le paramètre *nom-tâche* ne peut pas être déjà utilisé par une autre tâche quelconque de contrôle des valeurs seuil.

object=*nom-objet*

Objet contrôlable pour lequel la tâche de contrôle des valeurs seuil collectera les valeurs de l'attribut afin de les comparer à la valeur seuil définie. La valeur de *nom-objet* doit correspondre à l'un des noms affichés par les sous-commandes `thrsh-job list` ou `thrsh-job observable-objects`.

granularity=*valeur-entière*

Intervalle exprimé en secondes qui sépare le début de deux contrôles successifs de la valeur de l'attribut alors que la tâche est active en fonctionnement.

attributeName=*nom-attribut*

Nom de l'attribut pour lequel la tâche de contrôle des valeurs seuil collecte les valeurs qu'elle compare à la valeur seuil. La valeur de *nom-attribut* doit correspondre à l'un des noms affichés par les sous-commandes `thrsh-job list` ou `thrsh-job observable-objects`.

attributeType=*type-attribut*

Type de l'attribut contrôlable à contrôler. La valeur de *type-attribut* doit figurer dans la liste affichée par les sous-commandes `thrsh-job info` et `thrsh-job observable-attributes`.

`thresholdValue=valeur-seuil`

Valeur de l'attribut contrôlé pour laquelle cette tâche de contrôle des valeurs seuil déclenchera une alarme en cas de dépassement dans la direction spécifiée par `direction-seuil`.

`thresholdOffset=valeur-décalage`

La valeur de *valeur-décalage* détermine la tolérance de la tâche de contrôle des valeurs seuil pour le déclenchement d'alarmes successives. La valeur de *valeur-décalage* doit être égale à zéro ou à une valeur positive. Après le déclenchement d'un premier événement d'alarme, une autre alarme est déclenchée lorsque la valeur de l'attribut contrôlé se trouve hors de la plage définie par *valeur-décalage* et `direction-seuil`.

`thresholdDirection=[ÉLEVATION|CHUTE]`

Lorsque la direction définie est ÉLEVATION, un événement d'alarme est redéclenché lorsque la valeur de l'attribut contrôlé est inférieure à *valeur-seuil-valeur-décalage*. Lorsque la direction définie est CHUTE, un événement d'alarme est redéclenché lorsque la valeur de l'attribut contrôlé est supérieure à *valeur-seuil+valeur-décalage*. Ce comportement s'applique même si la valeur de *valeur-décalage* est égale à zéro.

Sécurité : cette sous-commande peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs.

`thrsh-job delete nom-tâche`

Cette sous-commande supprime la tâche de contrôle des valeurs seuil *nom-tâche*. La valeur de *nom-tâche* doit correspondre à l'une des tâches affichées par la sous-commande `thrsh-job list`.

Sécurité : cette sous-commande peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs.

`thrsh-job suspend nom-tâche`

Cette sous-commande suspend la tâche de contrôle des valeurs seuil *nom-tâche*. Une tâche suspendue n'est pas active et ne collecte aucune donnée, quelle que soit sa planification. Elle reste toutefois définie et peut être réactivée via l'utilisation de la sous-commande `thrsh-job resume`. La valeur de *nom-tâche* doit correspondre à l'une des tâches affichées par la sous-commande `thrsh-job list`.

Sécurité : cette sous-commande peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs.

`thrsh-job resume nom-tâche`

Cette sous-commande reprend l'exécution de la tâche de contrôle des valeurs seuil *nom-tâche*. Une tâche dont l'exécution est reprise commencera à collecter des données et à envoyer des comptes rendus en fonction de sa planification. La valeur de *nom-tâche* doit correspondre à l'une des tâches affichées par la sous-commande `thrsh-job list`. Cette sous-commande est l'homologue de la sous-commande `thrsh-job suspend`.

Sécurité : cette sous-commande peut uniquement être exécutée par l'utilisateur qui a démarré le Conteneur d'agents communs.

Exemples

Le scénario fictif suivant montre comment utiliser la commande mfwkadm avec ses options et sous-commandes associées.

La sous-commande `list-modules` affiche les instances des composants de Java ES sur l'hôte courant avec les modules correspondants dans le Conteneur d'agents communs. L'exemple ci-dessous présente la liste de deux composants installés : Directory Server pour lequel aucune instance n'est en cours d'exécution et Web Server pour lequel une seule instance existe.

```
$ mfwkadm list-modules
```

```
Installed products and their running instances:
```

```
=====
```

```
Instances for installed product: com.sun.cmm.ds:collectionID=/opt/SUNWdsee/ds6,  
name=Sun Java(TM) System Directory Server,type=CMM_InstalledProduct
```

```
-----
```

```
No instance.
```

```
Instances for installed product: com.sun.cmm.ws:collectionID=/var/opt/SUNWwbsvr7,  
name=WebServer,type=CMM_InstalledProduct
```

```
-----
```

```
/wsPrefix/com.sun.cmm.ws:name=https-hostname.example.com,type=CMM_ApplicationSystem
```

La sous-commande `info` suivante affiche les objets contrôlables dans l'instance Web Server, avec les classes et les attributs pour chaque type de tâche.

```
$ mfwkadm info /wsPrefix/com.sun.cmm.ws:name=https-hostname.example.com,\  
type=CMM_ApplicationSystem
```

```
Information about running instance: /wsPrefix/com.sun.cmm.ws:  
name=https-hostname.example.com,type=CMM_ApplicationSystem
```

```
=====
```

```
Observable objects for performance jobs:
```

```
-----
```

```
+ Objects of class: com.sun.cmm.settings.CMM_ApplicationSystemSetting
```

```
    /wsPrefix/com.sun.cmm.ws:name=https-hostname.example.com-setting,
```

```
type=CMM_ApplicationSystemSetting
```

```
Observable attributes:
```

```
Caption [STRING]
ConfigurationDirectory [STRING]
CreationClassName [STRING]
Description [STRING]
DirectoryName [STRING]
ElementName [STRING]
InstanceID [STRING]
Name [STRING]
URL [STRING]
```

```
+ Objects of class: com.sun.cmm.settings.CMM_KeepAliveSetting
```

```
/wsPrefix/com.sun.cmm.ws:name=process-1-keepalive-setting,
type=CMM_KeepAliveSetting
```

```
Observable attributes:
```

```
AllocationUnit [STRING]
Caption [STRING]
ConnectionsUpperBound [LONG]
CreationClassName [STRING]
Description [STRING]
ElementName [STRING]
InputUnit [STRING]
InstanceID [STRING]
LowerAllocationLimit [LONG]
LowerInputLimit [LONG]
LowerOutputLimit [LONG]
Name [STRING]
OtherAllocationUnit [STRING]
OtherInputUnit [STRING]
OtherLowerAllocationLimit [LONG]
OtherLowerInputLimit [LONG]
OtherLowerOutputLimit [LONG]
OtherOutputUnit [STRING]
OtherUpperAllocationLimit [LONG]
OtherUpperInputLimit [LONG]
OtherUpperOutputLimit [LONG]
OutputUnit [STRING]
QueuedUpperBound [LONG]
SecondsTimeout [LONG]
TimeoutUpperBound [LONG]
UpperAllocationLimit [LONG]
UpperInputLimit [LONG]
```

```
UpperOutputLimit [LONG]
...
```

La commande suivante affiche la liste des tâches de contrôle des performances définies. Dans cet exemple, il n'y a qu'une seule tâche de contrôle des performances appelée myPerfJob qui surveille un seul objet :

```
$ mfwkadm pm-job list
```

```
BY_OBJECTS performance jobs:
```

```
=====
```

```
Performance job information for: myPerfJob
```

```
-----
```

```
Type:                BY_OBJECTS
State:               ACTIVE_ON_DUTY
Granularity period: 30
Reporting period:   1
By event:           EVENT_SINGLE
By file:           EVENT_SINGLE
Report format:     XML
Schedule:
  Global start time: Immediately
  Global stop time: Forever
  Weekly schedule: Everyday
  Daily schedule: All day
```

```
Observed objects:
```

```
      /wsPrefix/com.sun.cmm.ws:name=virtualServer-hostname.example.com-
webApp/-stats,type=CMM_VirtualServerWebModuleStats
```

```
Observed attributes:
```

```
      All available
```

```
BY_CLASSES performance jobs:
```

```
=====
```

```
No jobs found.
```

La commande suivante crée une tâche de contrôle de l'état opérationnel pour deux objets contrôlables dont le nom a été obtenu à l'aide des sous-commandes `opstat -job info` et `opstat -job observable-objects` :

```
$ mfwkadm opstat-job create myOpStatJob granularity=60 \\  
object=/wsPrefix/com.sun.cmm.ws:name=process-1,type=CMM_UnixProcess \\  
object=/wsPrefix/com.sun.cmm.ws:name=process-1-DNSCache1,type=CMM_DnsCache
```

La commande suivante suspend l'exécution de la tâche créée ci-dessus :

```
$ mfwkadm opstat-job suspend myOpStatJob
```

La commande suivante affiche les classes contrôlables pour d'éventuelles tâches de contrôle des valeurs seuil :

```
$ mfwkadm thrsh-job observable-classes
```

```
Threshold jobs observable classes:
```

```
=====
```

```
com.sun.cmm.cim.CIM_ScopedSettingData
com.sun.cmm.cim.CIM_SettingData
com.sun.cmm.cim.CIM_StatisticalData
com.sun.cmm.cim.statistics.CIM_EthernetPortStatistics
com.sun.cmm.cim.statistics.CIM_NetworkPortStatistics
com.sun.cmm.cim.statistics.j2ee.CIM_J2eeJVMStats
com.sun.cmm.cim.statistics.j2ee.CIM_J2eeStatistic
com.sun.cmm.settings.CMM_ApplicationSystemSetting
com.sun.cmm.settings.CMM_KeepAliveSetting
com.sun.cmm.settings.CMM_QueueTimeoutSetting
com.sun.cmm.settings.CMM_RFC2788ApplicationSystemSetting
com.sun.cmm.settings.CMM_ScopedSettingData
com.sun.cmm.settings.CMM_SoftwareResourceSetting
com.sun.cmm.settings.CMM_SWRBufferSetting
com.sun.cmm.settings.CMM_SWRLimitSetting
com.sun.cmm.settings.CMM_SWRQueueSetting
com.sun.cmm.settings.CMM_VirtualServerSetting
com.sun.cmm.statistics.CMM_ApplicationSystemStats
com.sun.cmm.statistics.CMM_ApplicationSystemWatchdogStats
com.sun.cmm.statistics.CMM_ConnectionQueueStats
com.sun.cmm.statistics.CMM_DnsCacheStats
com.sun.cmm.statistics.CMM_EthernetPortStats
com.sun.cmm.statistics.CMM_FileCacheStats
com.sun.cmm.statistics.CMM_HTTPResponsesStats
com.sun.cmm.statistics.CMM_JVMJSR174ExtStats
com.sun.cmm.statistics.CMM_JVMJSR174Stats
com.sun.cmm.statistics.CMM_JVMStats
com.sun.cmm.statistics.CMM_NetworkPortStats
com.sun.cmm.statistics.CMM_OperatingSystemStats
com.sun.cmm.statistics.CMM_ProcessorStats
com.sun.cmm.statistics.CMM_ProcessStats
com.sun.cmm.statistics.CMM_QueueTimeoutStats
com.sun.cmm.statistics.CMM_RFC2788ApplicationTableStats
com.sun.cmm.statistics.CMM_ServiceStats
com.sun.cmm.statistics.CMM_SoftwareResourceStats
com.sun.cmm.statistics.CMM_SolarisEthernetPortStats
com.sun.cmm.statistics.CMM_SolarisNetworkPortStats
com.sun.cmm.statistics.CMM_SolarisOperatingSystemStats
com.sun.cmm.statistics.CMM_SolarisProcessorStats
com.sun.cmm.statistics.CMM_SolarisProcessorSysinfoStats
```

```
com.sun.cmm.statistics.CMM_SolarisProcessorVmStats
com.sun.cmm.statistics.CMM_Statistic
com.sun.cmm.statistics.CMM_SWRBufferStats
com.sun.cmm.statistics.CMM_SWRCacheStats
com.sun.cmm.statistics.CMM_SWRLimitStats
com.sun.cmm.statistics.CMM_SWRQueueStats
com.sun.cmm.statistics.CMM_UnixOperatingSystemStats
com.sun.cmm.statistics.CMM_UnixProcessStats
com.sun.cmm.statistics.CMM_VirtualServerWebModuleStats
com.sun.cmm.statistics.CMM_WebModuleStats
```

La commande suivante affiche les attributs contrôlables pour les tâches de contrôle des valeurs seuil qui surveillent les objets de la classe `com.sun.cmm.statistics.CMM_SWRQueueStats` trouvée à l'exemple précédent :

```
$ mfwkadm thrsh-job observable-attributes \\  
class=com.sun.cmm.statistics.CMM_SWRQueueStats
```

```
Threshold jobs observable attributes:  
=====
```

```
Class: com.sun.cmm.statistics.CMM_SWRQueueStats
```

```
Attributes:
```

```
BufferSize [LONG]
EntriesCount [LONG]
EntriesHighWaterMark [LONG]
EntriesLowWaterMark [LONG]
EntriesTotal [LONG]
ErrorCount [INTEGER]
FailedOperations [LONG]
LowerLimit [LONG]
OperationsCount [LONG]
OtherLowerLimit [LONG]
OtherUpperLimit [LONG]
OverflowsCount [LONG]
QueuedCount [LONG]
QueuedHighWater [LONG]
SampleInterval [LONG]
TotalQueuedCount [LONG]
UpperLimit [LONG]
```

La commande suivante est un autre exemple de création de tâche, mais cette fois pour le contrôle des valeurs seuil :

```
$ mfwkadm thrsh-job create myThreshJob granularity=30 \\  
object=/wsPrefix/com.sun.cmm.ws:name=process-1-threadPool-NativePool-stats,\\
```

```
type=CMM_SWRQueueStats attributeName=EntriesCount attributeType=LONG \\  
thresholdValue=1000 thresholdOffset=10 thresholdDirection=RISING
```

L'exemple ci-dessous reproduit la sortie de la sous-commande `thrsh-job info` pour la tâche de contrôle des valeurs seuil créée à l'exemple précédemment :

```
$ mfwkadm thrsh-job info myThreshJob
```

```
Threshold job information for: myThreshJob
```

```
-----
```

```
Type:                SIMPLE  
State:               ACTIVE_ON_DUTY  
Granularity period: 30  
Schedule:  
    Global start time: Immediately  
    Global stop time: Forever  
    Weekly schedule: Everyday  
    Daily schedule: All day  
Alarm configuration:  
    Type: QualityOfServiceAlarm  
    Severity: INDETERMINATE  
Threshold definition(s):  
    Object: /wsPrefix/com.sun.cmm.ws:name=process-1-threadPool-  
NativePool-stats,type=CMM_SWRQueueStats  
    Attribute: EntriesCount [LONG]  
    Value: 1000  
    Direction: RISING  
    Offset: 10
```

État de sortie

Les valeurs de sortie suivantes sont retournées :

- 0 Exécution réussie
- 1 Une erreur est survenue.

Attributs

Type d'attribut	Valeur d'attribut
Disponibilité	SUNWmfwk

Type d'attribut	Valeur d'attribut
Stabilité de l'interface	Contract Private

Voir aussi

cacao.5, cacaoadm.1m

Installation et utilisation du composant Monitoring Console

Monitoring Console est l'application Web qui affiche toutes les données de contrôle collectées par le biais de l'instrumentation. Ce composant utilise un agent maître pour regrouper toutes les valeurs et notifications d'alarme de chacun des agents de noeud.

Une fois Monitoring Console installé, vous pouvez y accéder de façon sécurisée à partir d'une simple fenêtre de navigateur depuis un hôte quelconque, y compris via Internet, si votre pare-feu est configuré pour autoriser ce type d'accès. Par la suite, il vous suffit d'utiliser l'interface graphique pour afficher les valeurs contrôlées en temps réel, visualiser et accuser réception des alarmes et créer des règles de déclenchement pour les alarmes personnalisées.

Remarque – Avant de procéder à l'installation ou à la configuration de ce composant, consultez les *Notes de version de Sun Java Enterprise System 5 pour UNIX*.

Ce chapitre se compose des sections suivantes :

- [“Installation du composant Monitoring Console” à la page 53](#)
- [“Organisation des répertoires installés” à la page 57](#)
- [“Démarrage du composant Monitoring Console” à la page 57](#)
- [“Utilisation du composant Monitoring Console” à la page 61](#)
- [“Dépannage du composant Monitoring Console” à la page 70](#)

Installation du composant Monitoring Console

En raison des limitations spécifiques à l'agent maître de cette version, celui-ci ne peut pas résider sur le même hôte qu'un agent de noeud. Par conséquent, le composant Monitoring Console ne peut pas être installé sur un hôte où a déjà été installé un quelconque composant contrôlé de Java ES. Il doit être installé sur un hôte spécifique, sauf si des zones Solaris ont été configurées. Pour plus d'informations à ce sujet, reportez-vous à la section [“Pour installer le composant Monitoring Console dans une zone Solaris” à la page 55](#) ci-après.

L'installation du composant Monitoring Console implique également celle du composant Monitoring Framework en tant que dépendance de composant partagé. La console exige que la structure et le Conteneur d'agents communs chargent l'agent maître, mais celui-ci, contrairement à l'agent de noeud, n'est pas configurable par l'utilisateur. Plus précisément, vous ne devez pas utiliser la commande `mfwkadm` sur l'hôte ou dans la zone où vous installez le composant Monitoring Console.

▼ Pour installer le composant Monitoring Console avec le programme d'installation Java ES

En raison d'une limitation spécifique à cette version Bêta, vous devez installer le composant Monitoring Console sur un hôte ou dans une zone Solaris où aucun autre composant de Java ES n'est installé. C'est pour cette raison que Monitoring Console est le seul composant qui sera installé dans le cadre de cette procédure.

Cette procédure utilise l'interface graphique du programme d'installation. Pour plus d'informations sur l'exécution du programme d'installation dans d'autres modes, reportez-vous au Chapitre 4, "Installation à l'aide de l'interface en mode texte" du *Guide d'installation de Sun Java Enterprise System 5 pour UNIX* et au Chapitre 5, "Installation en mode Silencieux" du *Guide d'installation de Sun Java Enterprise System 5 pour UNIX*.

- 1 **Démarrez le programme d'installation à partir du répertoire approprié pour votre plate-forme dans cette version de Java ES. Pour plus de détails, reportez-vous à la section "Lancement de l'installation" du *Guide d'installation de Sun Java Enterprise System 5 pour UNIX*.**
- 2 **Après avoir passé l'écran de bienvenue et accepté la licence, choisissez le type de procédure, mise à jour ou installation, sélectionnez Installer un nouveau logiciel, puis cliquez sur Suivant.**
- 3 **Dans l'écran de sélection des composants à installer, sélectionnez uniquement le composant Sun Java System Monitoring Console. Cliquez sur Suivant.**
- 4 **Le programme d'installation vérifie alors les mises à jour requises pour les composants partagés. Une fois cette vérification terminée, cliquez sur Suivant.**
- 5 **À présent, le programme d'installation vérifie la configuration du système. Si votre système d'exploitation requiert l'application de patches, annulez l'installation, appliquez les patches voulus à votre système et recommencez cette procédure. Si aucun patch ne doit être appliqué, cliquez sur Suivant.**
- 6 **Dans l'écran de sélection du type de configuration, sélectionnez Configurer maintenant, puis dans l'écran de configuration personnalisée qui s'affiche, cliquez sur Suivant.**

- 7 Le programme d'installation est prêt à installer le composant Monitoring Console. Cliquez sur **Suivant pour commencer**. Au cours de l'installation, il peut s'avérer nécessaire d'ouvrir la fenêtre d'enregistrement du produit si vous n'avez pas précédemment enregistré le déploiement de Java ES.
- 8 Une fois l'installation terminée, vérifiez le résumé de l'installation et les fichiers journaux associés, puis cliquez sur **Installation terminée** pour quitter le programme d'installation.

Étapes suivantes Vous pouvez à présent passer à la procédure [“Pour configurer le composant Monitoring Console”](#) à la page 56.

▼ Pour installer le composant Monitoring Console dans une zone Solaris

Si vous avez créé des zones Solaris sur votre système, vous pouvez installer le composant Monitoring Console sur le même hôte physique que les autres composants de Java ES. Dans ce cas, vous devez placer les autres composants dans la zone globale et créer une zone racine locale sparse en tant qu'hôte logique pour le composant Monitoring Console. Pour ce faire, procédez comme suit en respectant l'ordre indiqué.

- 1 Installez et configurez tous les composants de Java ES dans la zone globale, excepté le composant Monitoring Console. Effectuez toutes les tâches de configuration de post-installation pour les composants de la zone globale de sorte que toutes les instances de serveur soient exécutées.
- 2 Dans le cadre de l'installation effectuée dans la zone globale, le composant Monitoring Framework est installé en tant que composant partagé dans cette zone. Suivez toutes les procédures décrites au [Chapitre 2](#) qui s'appliquent aux composants que vous avez installés.
- 3 Sur le même hôte, créez une zone racine locale sparse en tant qu'hôte logique pour le composant Monitoring Console. Comme il s'agit d'une zone racine sparse, le composant Monitoring Framework installé dans *base-mfwk* doit être visible (voir [“Chemins par défaut et noms de fichier”](#) à la page 9).
- 4 Installez le composant Monitoring Console dans la zone racine locale sparse en suivant la procédure [“Pour installer le composant Monitoring Console avec le programme d'installation Java ES”](#) à la page 54.
- 5 Configurez le composant Monitoring Framework dans la zone racine sparse à l'aide des commandes suivantes :

```
cd base-mfwk/bin
./mfwksetup -i
```

Exploitant les fichiers de la zone globale, cette commande crée les fichiers de configuration Monitoring Framework requis dans la zone locale.

Étapes suivantes Vous pouvez à présent passer à la procédure “[Pour configurer le composant Monitoring Console](#)” à la page 56.

▼ Pour configurer le composant Monitoring Console

Cette procédure explique comment configurer le composant Monitoring Console sur un hôte physique distinct. Si vous avez installé le composant Monitoring Console sur un hôte logique créé à l'aide d'une zone Solaris, les commandes utilisées sont les mêmes, mais elles doivent être exécutées au niveau du système de fichiers de cette zone.

- 1 **Utilisez le composant Monitoring Framework pour initialiser l'agent maître à l'aide des commandes suivantes :**

```
cd base-mfwk/bin
./masetup -i
```

- 2 **Redémarrez le Conteneur d'agents communs (cacao) à l'aide de la commande suivante :**

```
cacaoadm restart
```

▼ Pour annuler la configuration du composant Monitoring Console

Si vous avez installé et configuré le composant Monitoring Console sur un hôte où vous envisagez d'installer d'autres composants, vous ne serez pas en mesure de contrôler ces composants en raison d'un conflit généré au niveau du composant Monitoring Framework. Pour contrôler les nouveaux composants par le biais d'un agent de noeud, vous devez annuler la configuration de l'agent maître du composant Monitoring Console.

- **En tant que superutilisateur, exécutez les commandes suivantes pour annuler la configuration du composant Monitoring Console:**

```
cacaoadm stop
cacaoadm unregister-module com.sun.mfwk.masteragent.xml
cacaoadm register-module /etc/base-mfwk/xml/com.sun.mfwk.xml
cacaoadm restart
```

Organisation des répertoires installés

Pour connaître le nom du package installé sur votre système d'exploitation, reportez-vous au Chapitre 5, “Liste des packages pouvant être installés” du *Référence de l'installation de Sun Java Enterprise System 5 pour UNIX*. Le tableau ci-dessous présente les répertoires utilisés par le package Monitoring Console. Le répertoire d'installation par défaut *base-MConsole* fait référence au répertoire ci-dessous, comme indiqué à la section “Chemins par défaut et noms de fichier” à la page 9 :

- Systèmes Solaris : /opt/SUNWjesmc
- Systèmes Linux : /opt/sun/jesmc

TABLEAU 3-1 Répertoires et fichiers utilisés par le composant Monitoring Console

Chemin	Description du contenu
<i>base-MConsole</i> /WEB-INF/classes	Classes de servlets de l'application Web
<i>base-MConsole</i> /WEB-INF/lib	Dépendances JAR de l'application Web
<i>base-MConsole</i> /WEB-INF/*.xml	Descripteurs de l'application Web
<i>base-MConsole</i> /css	Fichiers de feuilles de style
<i>base-MConsole</i> /html	Fichiers HTML
<i>base-MConsole</i> /images	Fichiers image GIF utilisés par l'interface utilisateur
<i>base-MConsole</i> /js	Fichiers JavaScript™
<i>base-MConsole</i> /*.jsp	Fichiers JavaServer Pages™
<i>base-WebConsole</i> /prereg/jesmc/*.reg	Fichiers Web Console utilisés par le composant Monitoring Console

Démarrage du composant Monitoring Console

Le composant Monitoring Console est une application Web accessible à partir de n'importe quel navigateur capable de se connecter à l'hôte sur lequel vous l'avez installé. L'accès au composant Monitoring Console s'effectue via Web Console, qui est automatiquement installé sur le même hôte. La procédure ci-dessous explique comment accéder au composant Monitoring Console et afficher les composants contrôlés.

▼ Pour démarrer le composant Monitoring Console

- 1 Vous devez tout d'abord redémarrer le serveur Web où Web Console est installé. Exécutez la commande suivante sur l'hôte ou dans la zone où le composant Monitoring Console est installé :

```
/usr/sbin/smcwebserver restart
```

- 2 Attendez que Web Console démarre. Utilisez la commande suivante pour connaître son état :

```
/usr/sbin/smcwebserver status
```

Il se peut que vous deviez exécuter cette commande plusieurs fois avant de voir le message suivant :

```
Sun Java(TM) Web Console is running.
```

- 3 Ouvrez Web Console en spécifiant l'URL suivant dans un navigateur quelconque capable de se connecter à l'hôte Monitoring Console. Si vous avez installé le composant dans une zone Solaris, *hôte-MC* est le nom d'hôte logique donné à cette zone :

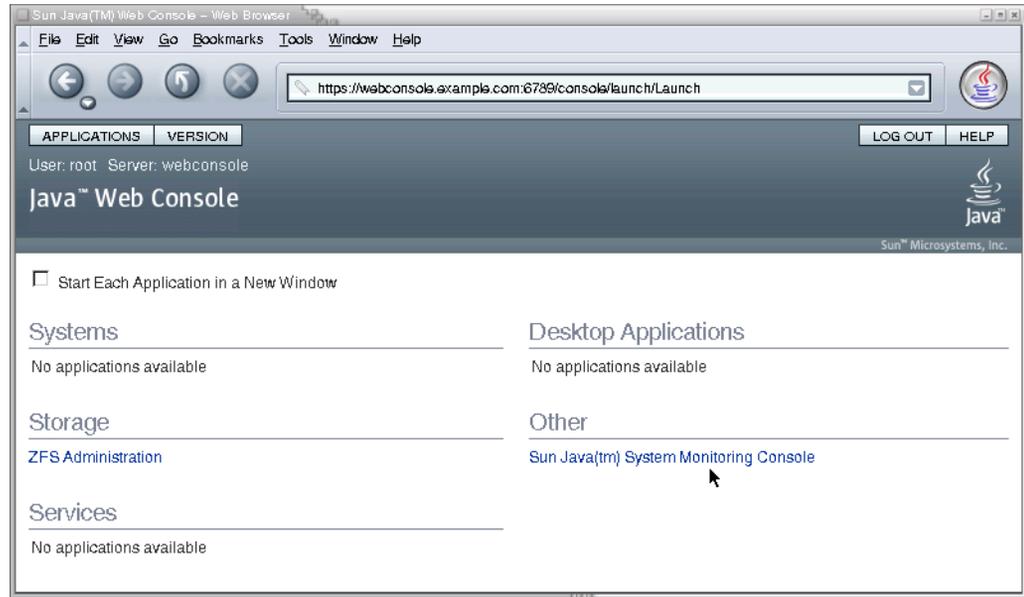
```
https://hôte-MC.domaine:6789
```

- 4 En fonction de la configuration de votre navigateur, un message peut s'afficher concernant un certificat non autorisé. Si tel est le cas, vous devez approuver le certificat pour accéder à Web Console.

- 5 Lorsque vous y êtes invité, connectez-vous à Web Console en tant que superutilisateur en utilisant le mot de passe root sur l'hôte Monitoring Console.

Une fois connecté, Web Console affiche la liste de tous les services qu'il propose.

- 6 Pour ouvrir la fenêtre principale du composant Monitoring Console, cliquez sur Sun Java System Monitoring Console sous le titre "Autre," comme illustré ci-dessous.

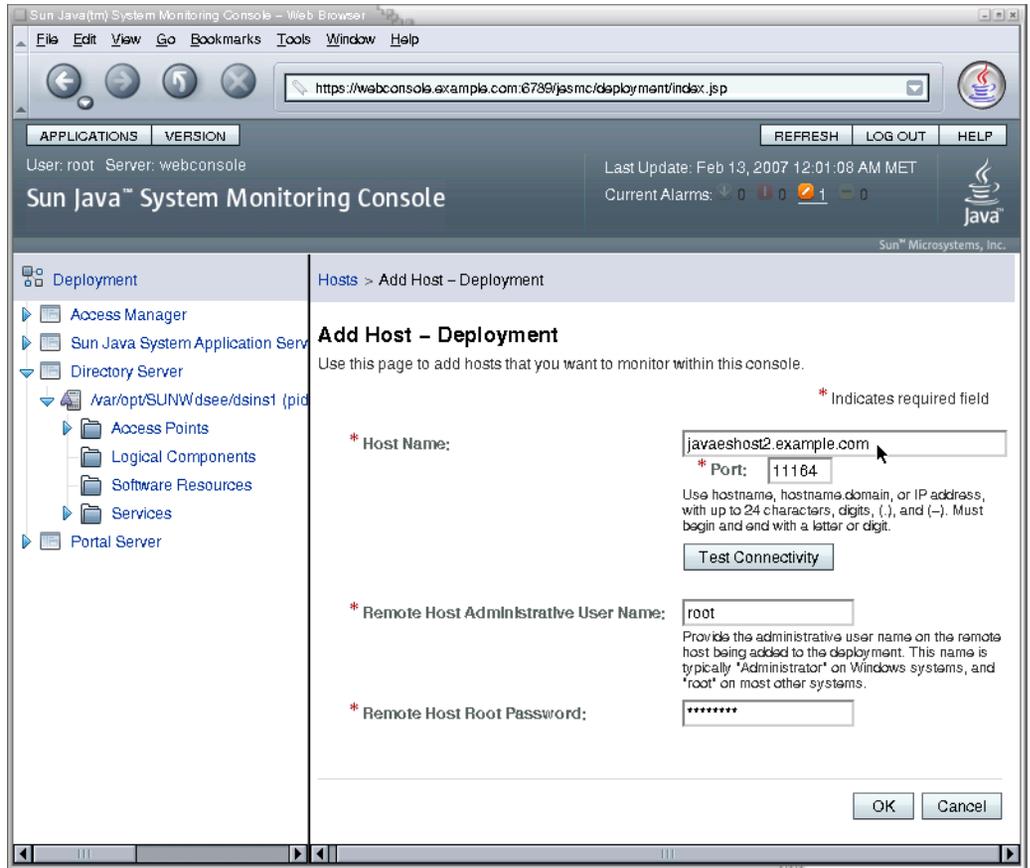


▼ Pour se connecter aux agents de noeud

Lors du premier démarrage du composant Monitoring Console, vous devez indiquer l'emplacement où résident les composants contrôlés. Après avoir spécifié l'emplacement de chaque agent de noeud dans votre déploiement de Java ES, la console affiche automatiquement tous les composants contenus dans chaque agent. Vous devrez également répéter cette procédure si vous ajoutez par la suite des composants de Java ES dans votre déploiement en les installant sur de nouveaux hôtes.

Lorsque vous ajoutez un agent de noeud, le composant Monitoring Console se connecte à celui-ci chaque fois que vous accédez à la console et ce, jusqu'à la suppression explicite de cet agent. Si un agent de noeud ajouté précédemment ne répond pas, suivez la procédure [“Pour redémarrer l'agent de noeud”](#) à la page 31.

- 1 **Synchronisez la date et l'heure entre l'hôte logique où le composant Monitoring Console a été installé et l'hôte contenant l'agent de noeud et les composants de Java ES à contrôler. Quel que soit le type de synchronisation utilisé (automatique ou manuelle), l'heure sur chaque hôte ne doit pas avancer de plus d'environ 10 minutes par rapport aux autres hôtes.**
- 2 **Si nécessaire, activez le niveau d'affichage Déploiement en cliquant sur le lien “Déploiement” en haut de la liste du volet gauche du composant Monitoring Console. Sélectionnez l'onglet Hôtes dans le volet droit et cliquez sur Ajouter.**
- 3 **Dans la boîte de dialogue Ajouter hôte qui s'affiche, entrez les informations requises, comme illustré ci-dessous :**



- Nom d'hôte : entrez le nom d'hôte complet d'un agent de noeud dans lequel vous avez configuré les composants contrôlés.
- Port : 11164, sauf si vous avez configuré différemment le composant Monitoring Framework sur l'hôte où réside l'agent de noeud.
- Mot de passe de la racine de l'hôte distant : tapez le mot de passe root du système où réside l'agent de noeud.

4 Cliquez sur Tester la connectivité.

Si les informations de connexion sont correctes et si l'agent de l'hôte est configuré et en cours d'exécution, la boîte de dialogue indique que la connexion est établie.

5 Cliquez sur OK pour fermer la boîte de dialogue Ajout hôte. Le nouveau nom d'hôte s'affiche dans la liste des hôtes. Tous les composants contrôlés contenus dans l'agent de noeud de cet hôte apparaissent alors aussi dans la colonne de gauche.

- 6 Répétez cette procédure pour chaque hôte de votre déploiement Java ES où vous avez installé des composants contrôlés.

Étapes suivantes Vous pouvez parcourir la liste des composants affichée dans la colonne de gauche pour afficher leur état opérationnel, les attributs contrôlés qu'ils exposent, ainsi que toute alarme qu'ils peuvent avoir déclenchée.

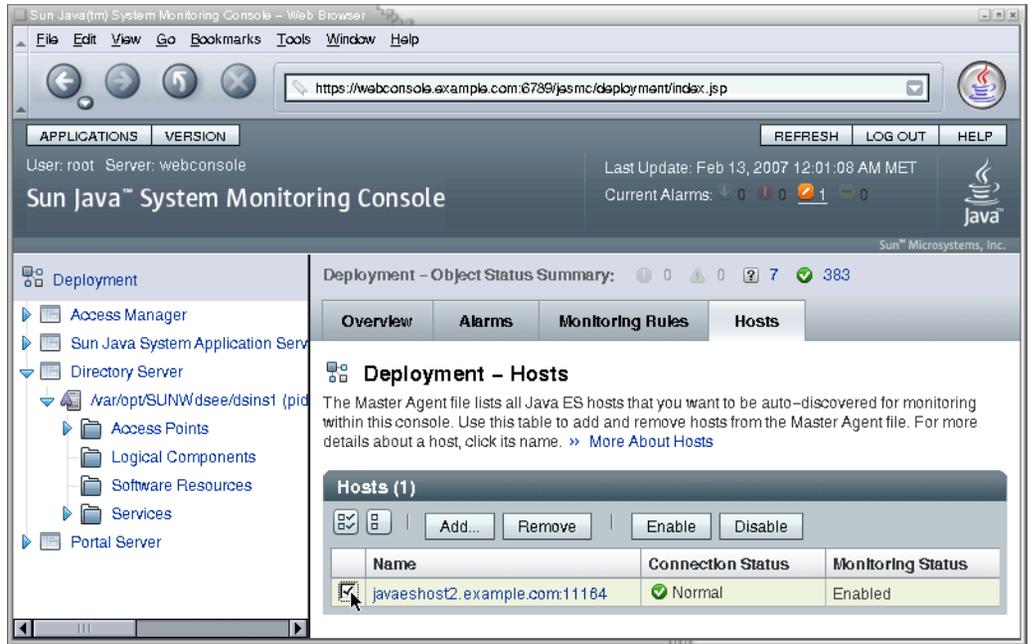
Utilisation du composant Monitoring Console

Les procédures présentées dans cette section expliquent comment interagir avec le composant Monitoring Console.

▼ Pour désactiver et réactiver le contrôle sélectif des composants

Les mécanismes de contrôle de Java ES sont des processus légers qui ont été conçus pour ne pas affecter les performances des systèmes de production. Toutefois, dans certains cas, il peut s'avérer préférable d'arrêter la collecte des valeurs de contrôle pour assurer l'absence d'impact de l'implémentation sur les performances. Le composant Monitoring Console intègre une fonction qui permet d'effectuer cette opération hôte par hôte, comme décrit dans la procédure suivante.

- 1 Si nécessaire, activez le niveau d'affichage Déploiement en cliquant sur le lien "Déploiement" en haut de la liste du volet gauche du composant Monitoring Console. Cliquez sur l'onglet Hôtes dans le volet droit.



La table présentée dans l'onglet Hôtes présente tous les hôtes sur lesquels sont installés des composants de Java ES contrôlés par le composant Monitoring Console.

- 2 Utilisez les cases à cocher affichées dans la colonne de gauche de la table pour sélectionner tous les hôtes pour lesquels vous souhaitez arrêter le contrôle. Cliquez sur Désactiver en haut de la table Hôtes.

Informations supplémentaires

Conséquences

Lorsque le contrôle d'un hôte est désactivé, tous les objets contrôlés de l'arborescence associée à cet hôte sont également désactivés. Les objets contrôlés dont l'état est Désactivé ne sont plus mis à jour, bien qu'ils puissent toujours afficher la dernière valeur de contrôle. Les règles de contrôle basées sur un objet désactivé sont suspendues. Pour réactiver un hôte, suivez la procédure ci-dessus en cliquant sur le bouton Activer en haut de la table Hôtes.

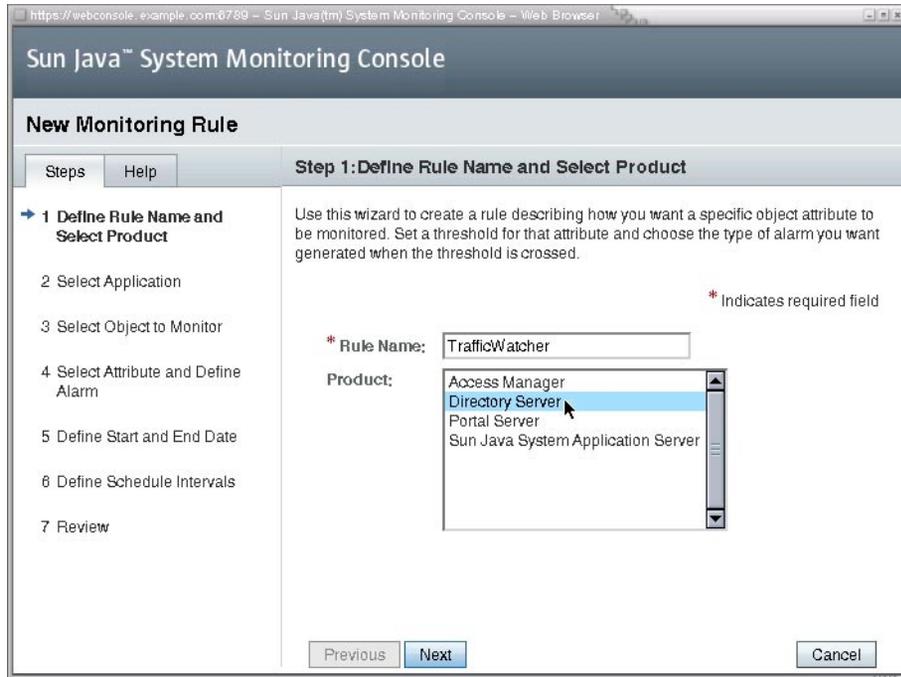
▼ Pour créer une nouvelle règle de contrôle

Une règle de contrôle, également appelée « tâche de contrôle », est un ensemble de conditions de valeurs contrôlées définies par l'utilisateur pour déclencher une alarme. L'assistant de création de règle de contrôle intégré au composant Monitoring Console vous aide à définir les conditions à contrôler.

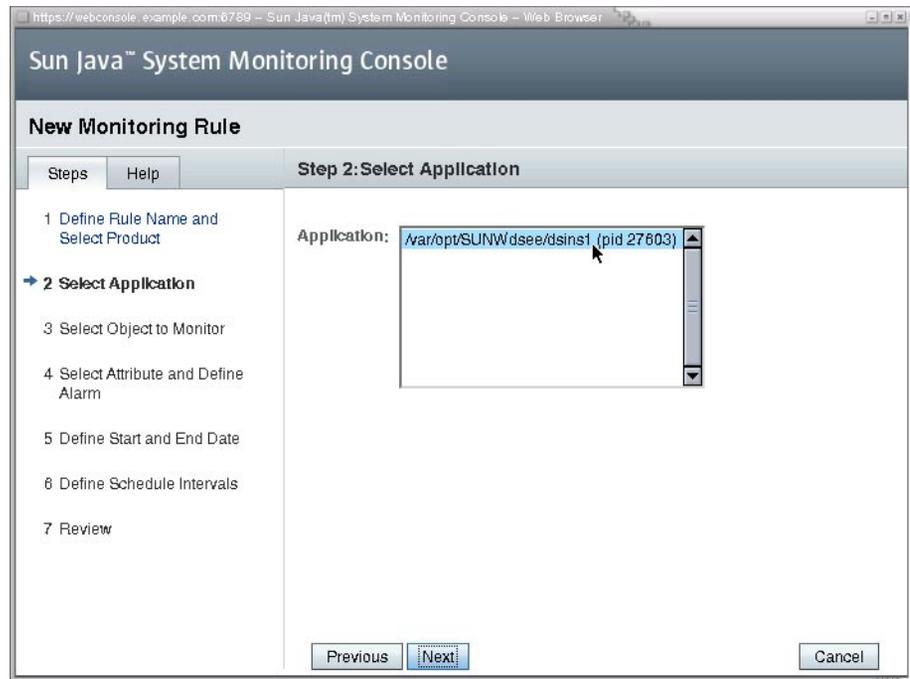
- 1 Si nécessaire, activez le niveau d'affichage Déploiement en cliquant sur le lien "Déploiement" en haut de la liste du volet gauche du composant Monitoring Console. Sélectionnez ensuite l'onglet Règles dans le volet droit, comme illustré ci-dessous, et cliquez sur Nouveau dans la table des règles de contrôle :



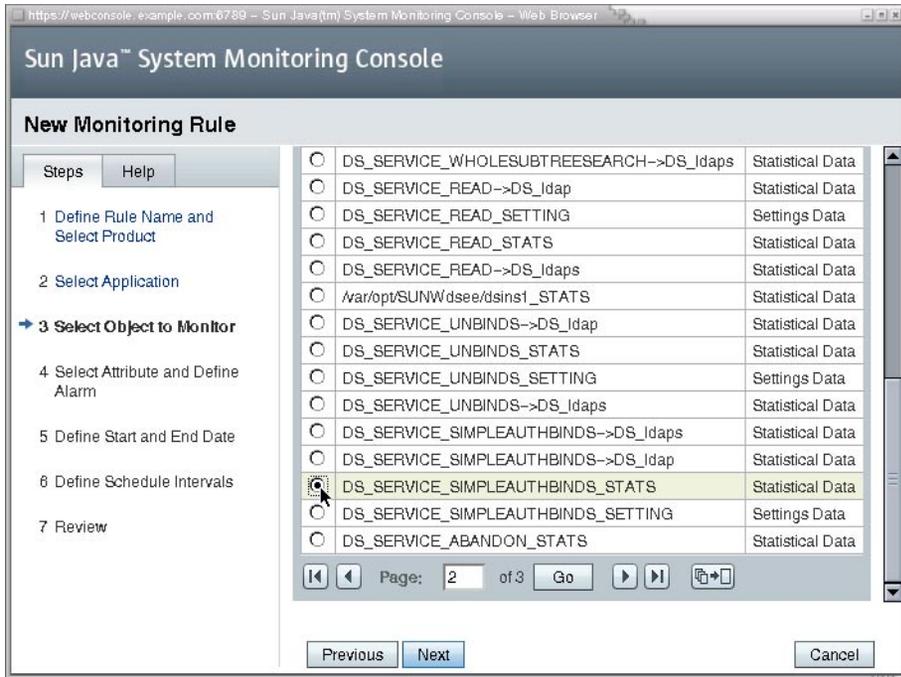
- 2 Donnez un nom à la nouvelle règle de contrôle et sélectionnez le type de serveur à contrôler.



- 3 Choisissez l'instance du composant à contrôler. Si deux instances du même composant installé existent sur des hôtes différents, il est possible que certaines instances affichées dans la table aient le même nom. Dans ce cas, il est possible que ces instances se présentent dans le même ordre que dans la liste du volet gauche, mais cela n'est pas systématique. Il est possible que vous deviez créer des règles de contrôle identiques pour chaque instance pour être sûr d'avoir défini la règle.



4 Sélectionnez l'objet qui contient l'attribut à contrôler :



- 5 Vous pouvez à présent spécifier l'attribut à contrôler, ainsi que les valeurs à utiliser pour déclencher une alarme.

The screenshot shows a web browser window titled "Sun Java™ System Monitoring Console". The main content area is titled "New Monitoring Rule" and "Step 4: Select Attribute and Define Alarm". On the left, a "Steps" sidebar lists seven steps, with step 4 being the current active step. The main area contains the following fields and instructions:

- Attribute:** A dropdown menu with "InRequests" selected.
- * Threshold:** A text input field containing "10000". Below it, the text reads: "Use seconds for time, KB for size."
- Direction:** A dropdown menu with "Rising" selected.
- * Offset:** A text input field containing "100". Below it, the text reads: "Enter a value greater than 0. Use the same measurement units as for Threshold."
- Alarm Severity:** A dropdown menu with "Minor" selected.

At the bottom of the form, there are three buttons: "Previous", "Next", and "Cancel". A red asterisk next to the "Threshold" and "Offset" labels indicates they are required fields.

- 6 Entrez les dates de début et de fin de la règle. Contrairement à la planification qui détermine la période d'activité ou d'inactivité d'une règle, les dates de début et de fin en définissent la période d'existence. Si la date début se trouve dans le passé, comme c'est toujours le cas avec la valeur par défaut, le contrôle associé à la règle débute immédiatement.

The screenshot shows a web browser window displaying the 'Sun Java™ System Monitoring Console'. The main heading is 'New Monitoring Rule'. Below this, there are two tabs: 'Steps' and 'Help'. The 'Steps' tab is active, showing a list of seven steps: 1. Define Rule Name and Select Product, 2. Select Application, 3. Select Object to Monitor, 4. Select Attribute and Define Alarm, 5. Define Start and End Date (highlighted with a blue arrow), 6. Define Schedule Intervals, and 7. Review. The main content area is titled 'Step 5: Define Start and End Date'. It contains several form fields: 'Start Date' (February 13, 2007), 'Start Time' (11:35), 'End Date' (February 14, 2009), and 'End Time' (00:00). A note below the end time states 'For 00:00, the rule is active through the End Date.' There is also a '* Granularity Period' field set to 300, with a note 'Time between data collections, in seconds'. A '* Indicates required field' note is present in the top right. At the bottom, there are 'Previous', 'Next', and 'Cancel' buttons.

- 7 Si vous le souhaitez, vous pouvez utiliser les boutons de contrôle pour créer un ou plusieurs intervalles pendant lesquels la règle de contrôle sera active. Il est également possible de sélectionner les jours de la semaine pour créer une planification hebdomadaire.

The screenshot shows a web browser window titled "Sun Java™ System Monitoring Console". The main heading is "New Monitoring Rule". Below it, there are two tabs: "Steps" and "Help". The "Steps" tab is active, showing a list of seven steps:

- 1 Define Rule Name and Select Product
- 2 Select Application
- 3 Select Object to Monitor
- 4 Select Attribute and Define Alarm
- 5 Define Start and End Date
- 6 Define Schedule Intervals
- 7 Review

The main content area is titled "Step 6: Define Schedule Intervals". It contains the following text: "Use this step to define one or more daily and weekly schedules. A daily schedule must be defined before a weekly schedule can be defined." Below this text is a section titled "Daily Schedule Intervals (1)". This section includes a table with two columns: "Start Time" and "End Time". The "Start Time" column has a checkbox and a dropdown menu showing "08" and "00". The "End Time" column has a dropdown menu showing "20" and "00". To the right of the table are "Add" and "Remove" buttons. Below the table, there is a "Days of Week" section with checkboxes for Mon, Tue, Wed, Thu, Fri, Sat, and Sun. At the bottom of the form are "Previous", "Next", and "Cancel" buttons.

- 8 Pour cette dernière étape de l'assistant de création de règle, vérifiez les informations entrées, puis cliquez sur Terminer pour créer la règle.



Une fois l'assistant de création de règle fermé, l'onglet Règles de contrôle réapparaît. La règle nouvellement créée s'affiche dans la table des règles.

Dépannage du composant Monitoring Console

Consultez également les problèmes connus décrits dans les *Notes de version de Sun Java Enterprise System 5 pour UNIX*.

Si un conflit existe entre l'agent maître et un agent de noeud, vérifiez les conditions suivantes :

- Si vous utilisez des zones Solaris, assurez-vous d'avoir installé le composant Monitoring Console dans une zone racine locale sparse.
- Assurez-vous d'avoir supprimé toute installation précédente du composant contrôlé sur l'hôte ou dans la zone concerné.
- Dans les deux cas, vous devez désinstaller le composant Monitoring Console et tout autre composant, résoudre le problème, puis installer de nouveau le composant Monitoring Console.

Si vous désinstallez et réinstallez le composant Monitoring Console sur le même hôte, celui-ci ne pourra pas être initialisé et il ne sera pas visible dans Web Console. Dans ce cas, exécutez la commande `masetup -i` sur l'hôte Monitoring Console pour initialiser l'agent maître. Suivez ensuite la procédure [“Pour démarrer le composant Monitoring Console”](#) à la page 58.

L'une des limitations inhérente aux règles de contrôle consiste à ne permettre leur désactivation que si elles sont activées. Pour désactiver une règle dont la planification la rend inactive au moment voulu, vous devez modifier sa planification de façon à la rendre temporairement active ou la supprimer totalement.

En raison des limitations spécifiques à la plate-forme Windows, les valeurs `handleCount` et `threadCount` présentées dans les statistiques de l'hôte affichent toujours 0 (zéro).

Références sur les objets CMM

Le modèle de contrôle commun Modèle de contrôle commun (CMM, Common Monitoring Model) est une extension du modèle d'information commune (CIM, Common Information Model) implémenté dans le langage de programmation Java. Le modèle CIM est intégré aux interfaces Java des packages `com.sun.cmm.cim.*`. L'implémentation du modèle CMM concerne les interfaces des packages `com.sun.cmm.*` destinés à l'extension des interfaces CIM. Les objets contrôlés sont représentés dans les agents de noeud par classes d'implémentation des interfaces CMM. Les tableaux suivants indiquent les attributs contrôlés pour chaque classe d'objets.

Présentation des objets CMM

Le modèle CMM est basé sur un ensemble limité d'interfaces de base qui déterminent les attributs qu'un objet contrôlé de ce type peut exposer. La liste ci-après répertorie les classes utilisées pour représenter les différents types d'objets contrôlés définis par le modèle CMM et fournit certains des principaux attributs associés :

<code>CMM_InstalledProduct</code>	Composant de Java ES, considéré dans son intégralité (par exemple, Java ES Directory Server).
<code>CMM_ApplicationSystem</code>	Instance installée et configurée d'un composant de Java ES. Cette instance peut être en cours d'exécution ou ne pas l'être. Les attributs standard de cet objet sont les coordonnées de l'administrateur, l'état opérationnel du système et l'heure de démarrage et d'arrêt de l'application.
<code>CMM_Service</code>	Fonction spécifique d'un composant (par exemple, le service d'authentification Java ES Directory Server). Pour cet objet, l'état opérationnel du système constitue l'un des attributs standard.

CMM_SoftwareResource	Représentation des entités logicielles de l'environnement, comme un cache, un pool de threads, etc. La taille du cache constitue l'un des attributs standard de ce type d'objet.
CMM_LogicalComponent	Entité utilisée par un service et visible pour l'utilisateur final, mais qui ne représente pas une ressource physique ni une fonction logicielle en tant que telle. Il peut s'agir, par exemple, d'un groupe de paramètres de configuration pour une instance de logiciel, indépendamment de l'instance proprement dite.
CMM_ServiceAccessURI	Point d'accès au niveau duquel un service est disponible pour utilisation. Les attributs standard pour ce type d'objet sont un numéro de port et un URI (Uniform Resource Identifier).
CMM_RemoteServiceAccessPoint	Informations relatives à l'accès et à l'adressage pour une connexion distante. Les attribut standard de ce type d'objet sont un URI et l'état opérationnel de la connexion (ouverte ou fermée).
CMM_Process	Instance unique d'un programme en cours d'exécution. Les attributs standard de ce type d'objet sont l'utilisation de la mémoire et de CPU.
CMM_UnitaryComputerSystem	Hôte unique utilisé pour le déploiement de Java ES (par exemple, un ordinateur de bureau ou un serveur). Les attributs standard de ce type d'objet sont le nombre de processeurs disponibles et la quantité de mémoire physique.
CMM_OperatingSystem	Logiciel ou microprogramme grâce auquel l'exploitation des ressources matérielles d'une machine hôte est possible. L'attribut standard de ce type d'objet est la quantité de mémoire virtuelle disponible sur le système.
CMM_JVM	Machine virtuelle Java (JVM) utilisée par un serveur Java ES. L'attribut standard de ce type d'objet est le numéro de version de la machine virtuelle Java.
CMM_DatabaseService	Tâche exécutée pour le compte d'une base de données (par exemple, la fourniture d'accès aux utilisateurs). L'attribut standard de ce type d'objet est le nombre maximum autorisé d'accès à la base de données.
CMM_CommonDatabase	Propriétés commune d'un type spécifique de base de données. L'attribut standard de ce type d'objet est la date

de la dernière sauvegarde.

Objets contrôlés exposés par chaque composant

Les sections de cette annexe répertorient les objets CMM qui ont été instrumentés dans chacun des composants du produit prenant en charge la fonction de contrôle. Dans le cas où seul un sous-ensemble des attributs d'un objet sont instrumentés, ceux-ci sont également répertoriés.

Instrumentation du Conteneur d'agents communs

Aucune documentation disponible actuellement.

Instrumentation du composant Access Manager

Aucune documentation disponible actuellement.

Instrumentation du composant Application Server

Aucune documentation disponible actuellement.

Instrumentation du composant Calendar Server

Aucune documentation disponible actuellement.

Instrumentation du composant Directory Server

Aucune documentation disponible actuellement.

Instrumentation du composant Instant Messaging

Aucune documentation disponible actuellement.

Instrumentation du composant Messaging Server

Aucune documentation disponible actuellement.

Instrumentation du composant Portal Server

Aucune documentation disponible actuellement.

Instrumentation du composant Web Server

Aucune documentation disponible actuellement.

Index

G

Glossaire, lien, 9

