Sun Desktop Manager 1.0 Installationshandbuch

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

Teilenr.: 819–6090–10

060406@14558

Inhalt

Vorwort

Dieses Dokument bietet eine Beschreibung der für die Bereitstellung von Sun™ Desktop Manager 1.0 erforderlichen Installations- und Konfigurationsschritte.

Überblick

Mit Sun Desktop Manager soll eine zentrale Konfiguration für Desktop-Rechner erreicht werden. Dank der Möglichkeit, Einstellungen für verschiedene Elemente einer Organisations- oder Domänenstruktur vorzunehmen, können Administratoren auf effiziente Weise Benutzer- oder Rechnergruppen verwalten.

Aufbau dieses Handbuchs

Kapitel 1 bietet einen kurzen Überblick über Sun Desktop Manager.

Kapitel 2 behandelt die serverseitige Installation von Sun Desktop Manager.

Kapitel 3 bietet Informationen zur Installation von Java Desktop System Configuration Agent.

Kapitel 4 bietet Installationsinformationen zu Java Web Console.

Anhang A enthält Informationen zu Konfigurationsparametern.

Anhang B behandelt die Verwendung von OpenLDAP und Active Directory mit Desktop Manager.

Anhang C bietet Informationen über organisatorische Zuordnungen.

Zusätzliche Dokumentation

- *Sun Desktop Manager 1.0 Administration Guide*
- *Sun Desktop Manager 1.0 Developer Guide*

Dokumentation, Support und Schulungen

Sun-Funktion	URL	Beschreibung
Dokumentation	`http://www.sun.com/documentation/`	PDF- und HTML-Dokumente herunterladen, gedruckte Dokumentation bestellen
Support und Schulung	`http://sunsolve.sun.com`	Technischen Support erhalten, Programmkorrekturen herunterladen, Informationen zu Sun-Kursen abrufen

◆ ◆ ◆ KAPITEL 1

Übersicht und Begriffe

Dieses Dokument bietet eine Beschreibung der Installations- und Konfigurationsschritte, die für die Bereitstellung von Sun™ Desktop Manager 1.0 erforderlich sind. Eine umfassendere Übersicht zu Sun Desktop Manager finden Sie im *Sun Desktop Manager 1.0 Administration Guide*.

Übersicht zu Sun Desktop Manager

Sun Desktop Manager bietet eine zentrale Konfiguration für Desktop-Rechner. Dank der Möglichkeit, Einstellungen für verschiedene Elemente einer Domänenstruktur vorzunehmen, können Administratoren auf effiziente Weise Benutzer- oder Rechnergruppen verwalten.

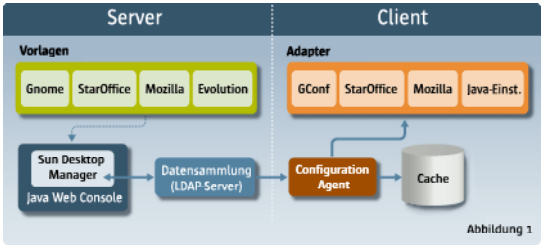

ABBILDUNG 1–1 Desktop Manager-Architektur

Folgendes sind die Hauptkomponenten von Desktop Manager:

- Konfigurationsdatensammlungen
- Verwaltungs-Tools
- Desktop Manager-Vorlagen
- Configuration Agent
- Konfigurationsadapter

Konfigurationsdaten werden zentral in Konfigurationsdatensammlungen gespeichert. Die Konfigurationsdaten werden mithilfe der Verwaltungs-Tools verwaltet

(Erstellen/Löschen/Ändern/Zuweisen/Zuweisung aufheben), welche aus einer webbasierten grafischen Benutzerschnittstelle für Desktop Manager und einer Befehlszeilen-Schnittstelle (CLI) bestehen. Die Vorlagen werden vom webbasierten Verwaltungs-Tool zur Darstellung der Konfigurationsdaten im Webbrowser verwendet.

Configuration Agent ruft im Namen der Benutzeranwendungen Konfigurationsdaten von der Konfigurationsdatensammlung ab. Der Agent legt die von der zentralen Konfigurationsdatensammlung abgerufenen Informationen im Cache ab.

Die Verwaltungs-Tools sind vom Agent vollkommen unabhängig. Das heißt, dass sie nur auf die Konfigurationsdatensammlung angewendet werden.

Die Benutzeranwendungen (die die Konfigurationsadapter verwenden) fragen die Konfigurationsdaten über Configuration Agent ab.

Das Produkt unterstützt direkt die Abfrage und Anwendung von Einstellungen für die folgenden Konfigurationssysteme:

- GConf: Gnome Configuration Framework
- StarOffice Registry
- Mozilla-Einstellungen
- Java-Einstellungen

◆ ◆ ◆ KAPITEL 2

2 Installation der Verwaltungsanwendung

Dieses Kapitel gibt Anweisungen zur Installation der serverseitigen Komponenten von Sun Desktop Manager.

Sun Desktop Manager

Desktop Manager bietet ein webbasiertes Administrations-Tool, das auf Java Web Console ausgeführt wird. Diese Benutzeroberfläche erlaubt es Administratoren, die Hierarchie einer Organisation zu durchlaufen und darin Richtlinien für Desktop-Anwendungen festzulegen. Es können Richtlinien für jedes Element in der Hierarchie definiert werden, beispielsweise für Organisationen, Rollen, Benutzer, Domänen und Rechner. In Desktop Manager werden die für die einzelnen Desktop-Anwendungen, wie Gnome, Mozilla, StarOffice und Evolution, spezifischen Einstellungen in verschiedenen Konfigurationsvorlagen präsentiert.

▼ Installation

Bevor Sie beginnen

Desktop Manager erfordert eine Installation von Java Web Console Version 2.2.5 oder höher. Stellen Sie sicher, dass auf Ihrem System eine gültige Version installiert ist. Um festzustellen, ob Sie eine gültige Version haben, melden Sie sich als Superuser (Root) an und führen Sie folgenden Befehl aus:

```
# smcwebserver status
```

Hinweis – Java Web Console 2.2.4 ist Teil des Betriebssystems Solaris™ 10. Desktop Manager erfordert jedoch Version 2.2.5 oder höher. Eine Kopie von Version 2.2.5 wird im Desktop Manager-Archiv im Verzeichnis server/console bereitgestellt. Führen Sie zur Installation den Befehl `./setup` in diesem Verzeichnis aus (falls erforderlich).

Ist Java Web Console nicht auf Ihrem System installiert, oder ist die installierte Version nicht für Desktop Manager geeignet, lesen Sie die Anweisungen in Kapitel 4, um zunächst Java Web Console zu installieren bzw. zu aktualisieren. Kehren Sie danach zu diesem Kapitel zurück und fahren Sie mit der Installation von Desktop Manager fort.

1 **Laden Sie das ZIP-Archiv von Desktop Manager herunter und extrahieren Sie dessen Inhalt in ein temporäres Verzeichnis.**

```
# unzip SunDesktopMgr-1.0.zip
```

2 **Melden Sie sich als Superuser (Root) an und führen Sie das Setup-Skript wie folgt aus:**

```
# cd SunDesktopMgr-1.0/<Platzform>/server/manager
# ./setup
```

3 **Prüfen Sie das Setup-Skript auf Fehler.**

Wurde die Installation erfolgreich durchgeführt, startet das Setup-Skript Java Web Console automatisch neu, und Sie können mit einem Webbrowser auf Desktop Manager zugreifen.

▼ Vorgang

1 **Geben Sie die folgende URL in Ihren Browser ein:**

```
https://<Hostname>.<Domänenname>:6789
```

2 **Geben Sie in den Anmeldebildschirm den Benutzernamen und das Passwort eines vorhandenen Unix-Benutzers ein.**

Java Web Console wird geöffnet.

3 **Klicken Sie auf der Startseite der Konsolenanwendung auf die Verknüpfung für Desktop Manager.**

- **Wenn Sie die Startseite der Konsolenanwendung überspringen und direkt zu Desktop Manager gehen möchten, geben Sie in Ihren Browser folgende URL ein:**

  ```
  https://<Host-Name>.<Domänenname>:6789/apoc
  ```

▼ Entfernen von Desktop Manager

◗ **Um Desktop Manager aus Java Web Console zu entfernen, müssen Sie in das für die Installation erstellte temporäre Verzeichnis wechseln, sich als Superuser anmelden und folgende Befehle ausführen:**

```
# cd server/manager
# ./setup -u
```

Migrationsprobleme

Desktop Manager ist kompatibel mit Vorgängerversionen von Java Desktop System Configuration Manager (Versionen 1.0 und 1.1). Es gibt jedoch einige Unterschiede, über die Sie sich im Klaren sein sollten.

In den früheren Versionen von Configuration Manager wurden alle Profildaten auf einem spezifischen LDAP-Server gespeichert. Dieser LDAP-Server wurde im Rahmen des Installationsvorgangs von Configuration Manager konfiguriert. Dazu gehörte auch die Konfiguration eines LDAP-Anmeldemoduls, das die Authentifizierung beim LDAP-Server enthielt.

Jetzt werden alle für Desktop Manager erforderlichen Konfigurationsschritte mithilfe eines Assistenten durchgeführt, und es müssen während der Installation keine Konfigurationen mehr vorgenommen werden. Desktop Manager bietet auch Unterstützung für viele Konfigurationsdatensammlungen. So können Sie Richtliniendaten verwalten, die auf verschiedenen LDAP-Servern, in dateibasierten Datensammlungen usw. gespeichert sind. Die Konfiguration eines spezifischen LDAP-Anmeldemoduls ist nicht mehr erforderlich.

In den LDAP-Schemata wurden von der einen Version zur anderen keine Änderungen vorgenommen. Wenn Sie bereits einen LDAP-Server für eine frühere Version von Configuration Manager konfiguriert haben, sind beim Umstieg auf Desktop Manager keine Änderungen erforderlich. Aus diesem Grund können Sie Desktop Manager nutzen, ohne den Client (Java Desktop System Configuration Manager1.1. Agent) oder die LDAP-Seite zu aktualisieren.

Hinweis – Vor der Installation von Desktop Manager sollten Sie zunächst alle früheren Installationen von Configuration Manager oder Desktop Manager von Ihrem System entfernen. Um frühere Installationen zu entfernen, müssen Sie (als Superuser) folgende Befehle ausführen:

```
# cd server/manager
# ./setup -u
```

Nach Installation von Desktop Manager können Sie eine Konfigurationsdatensammlung erstellen, die auf Ihren vorhandenen LDAP-Server zeigt:

▼ Erstellen von Konfigurationsdatensammlungen

1 Geben Sie die folgende URL in Ihren Browser ein:

```
https://<Host-Name>.<Domänenname>:6789
```

2 Geben Sie im Anmeldebildschirm den Benutzernamen und das Passwort eines vorhandenen Unix-Benutzers ein.

Java Web Console wird geöffnet.

3 Klicken Sie auf der Startseite der Konsolenanwendung auf die Verknüpfung für Sun Desktop Manager 1.0.

4 **Klicken Sie auf die Schaltfläche "Neu", um den Assistenten für neue Konfigurationsdatensammlungen zu starten.**

Der Assistent führt Sie durch die notwendigen Schritte zur Konfiguration einer LDAP-basierten Konfigurationsdatensammlung.

Achtung – Der Assistent bietet automatisch die Migration der vorhandenen Richtliniendaten zum neuen Format 2.0 an. Diese Migration ist optional und kann hauptsächlich zur Leistungsverbesserung der neueren Sun Desktop Manager 1.0-Agents verwendet werden. Sie sollten diese Migration nicht durchführen, wenn in Ihrer Umgebung noch Java Desktop System Configuration Manager 1.1-Agents unterstützt werden müssen.

Problembehebung für Desktop Manager

Installation nicht möglich

Symptom: Am Ende der Installation von Java Web Console wird eine Meldung angezeigt, dass das Programm nicht gestartet werden kann, da keine registrierten Anwendungen vorhanden sind.

Mögliche Ursachen: Es wurden keine Anwendungen installiert, auch nicht Desktop Manager. .

Lösung: Installieren Sie Desktop Manager und starten Sie dann Java Web Console.

Verbindung verweigert

Symptom: Sie versuchen, eine korrekte URL zu öffnen, beispielsweise `http://<`*Host-Name*`>.<`*Domänenname*`>:6789`, aber Sie erhalten eine Nachricht, dass die Verbindung verweigert wird.

Mögliche Ursachen: Java Web Console läuft nicht auf dem Server.

Lösung: Melden Sie sich zum Starten von Java Web Console als Superuser an und führen Sie folgende Befehle aus:

```
#smcwebserver status
#smcwebserver start
```

Anmeldung nicht möglich

Symptom: Auf der Anmeldeseite von Java Web Console wird die Kombination aus Benutzernamen und Passwort zurückgewiesen.

Mögliche Ursachen: Das entsprechende UNIX-Benutzerkonto ist nicht vorhanden.

Lösung: Prüfen Sie, ob ein entsprechender UNIX-Benutzername und ein entsprechendes Passwort auf Ihrem System konfiguriert ist. Erstellen Sie, falls erforderlich, ein lokales UNIX-Benutzerkonto für Ihre Tests.

Keine Verknüpfung zu Desktop Manager

Symptom: Auf der Seite mit der Java Web Console-Anwendungsliste wird die Verknüpfung für Sun Desktop Manager nicht angezeigt.

Mögliche Ursachen: Das Modul Desktop Manager ist nicht installiert.

Lösung: Um zu prüfen, ob Desktop Manager in Java Web Console installiert ist, müssen Sie sich als Superuser anmelden und den folgenden Befehl ausführen:

```
# smreg list -a
```

Enthält die Liste nicht die Anwendung `com.sun.apoc.manager_<`*Version*`>`, müssen Sie Desktop Manager erneut installieren.

Null-Zeiger-Ausnahme, Tomcat/Java-Fehler oder leere Seite

Symptom: Sie starten Desktop Manager, aber es werden nur eine leere Seite oder Fehlermeldungen angezeigt.

Mögliche Ursachen: Wenn in der Fehlermeldung der Fehler `NoClassDefFoundError: sun/tools/javac/Main` erwähnt wird, verwendet Java Web Console die falsche Java-Version.

Lösung: Die aktuelle Java-Umgebung für Java Web Console kann durch Ausführen des Befehls `# smreg list -p` und Prüfen der Eigenschaft `java.home` geprüft werden. Diese Eigenschaft muss einen gültigen Wert für Java-Home haben. Beim Java-Home muss es sich um ein JDK handeln. Ist dieser Wert falsch eingestellt, müssen Sie folgenden Befehl ausführen:

```
# smreg add -p java.home=<JAVA_HOME>
```

Hinweis – *<JAVA_HOME>* muss auf eine gültige Installation zeigen, beispielsweise eine Installation, bei der `javac` im Unterverzeichnis `bin` vorliegt.

Sie müssen dann Java Web Console mit folgendem Befehl neu starten:

```
# smcwebserver restart
```

Verbindung zu SSL-LDAP-Server verweigert

Symptom: Nach Angabe der LDAP-Serverdetails (einschließlich Aktivieren des Kontrollkästchens für die Verwendung von SSL) im Assistenten für die Erstellung von Datensammlungen wird eine Meldung eingeblendet, die besagt, dass die Verbindung mit dem Server nicht möglich ist.

Mögliche Ursachen: Die angegebene Port-Nummer ist nicht korrekt, der LDAP-Server ist nicht für Verbindungen mit SSL über diesen Port konfiguriert, oder die entsprechenden Zertifikate sind im Schlüsselspeicher von Java Web Console nicht vorhanden.

Lösung: Prüfen Sie zunächst, ob der LDAP-Server für den im Assistenten angegebenen Port für SSL-Verbindungen konfiguriert ist. Ist dies der Fall , stellen Sie sicher, dass entweder die Zertifizierungsautorität oder das LDAP-Serverzertifikat im Schlüsselspeicher von Java Web Console

vorhanden ist, der im Verzeichnis `/etc/opt/webconsole/keystore` liegt. Das Zertifikat kann mit dem Befehl `keytool -import -file <Zertifikatsdatei> -keystore /etc/opt/webconsole/keystore` hinzugefügt werden. Das Standardpasswort für diesen Schlüsselspeicher ist **`changeit`**. Java Web Console muss mit dem Befehl `smcwebserver restart` neu gestartet werden, damit die Änderung für Desktop Manager sichtbar wird.

Schreiben in Verzeichnis nicht möglich

Symptom: Beim Erstellen eines dateibasierten oder kombinierten Backend wird ein Fehler der Art "Schreiben in Verzeichnis nicht möglich!" angezeigt.

Mögliche Ursachen: Der Benutzer ohne Zugriff (noaccess) verfügt nicht über die korrekten Rechte.

Lösung: Weisen Sie diesem Benutzer Schreibrechte zu.

◆ ◆ ◆ KAPITEL 3

Client-Komponenten

Für den Zugriff auf die Konfigurationsdaten von Desktop Manager muss auf dem Desktop-Client Java Desktop System Configuration Agent installiert sein. Configuration Agent kommuniziert mit der entfernten Konfigurationsdatensammlung und den Adaptern und fügt Daten in spezifische Konfigurationssysteme ein. Derzeit werden die Konfigurationssysteme GConf, Java-Einstellungen, Mozilla-Einstellungen und StarOffice Registry unterstützt.

Eine Version von Configuration Agent wird mit dem Betriebssystem Solaris 10 bereitgestellt. Desktop Manager erfordert jedoch eine neuere Version dieses Tools. Diese neuere Version wird als Teil der Installation von Client-Komponenten für Desktop Manager und damit verbundenen Patches installiert.

So installieren Sie die Client-Komponenten für Desktop Manager:

1. Laden Sie das ZIP-Archiv von Desktop Manager herunter und extrahieren Sie dessen Inhalt in ein temporäres Verzeichnis.

   ```
   # unzip SunDesktopMgr-1.0.zip
   ```

2. Installieren Sie die empfohlenen Patches.

   Diese Patches werden im Verzeichnis `SunDesktopMgr-1.0/<Plattform>/client/Patches` bereitgestellt. Befolgen Sie die für jeden Patch bereitgestellten Anweisungen.

3. Melden Sie sich als Superuser (Root) an und führen Sie das Setup-Skript wie folgt aus:

   ```
   # cd SunDesktopMgr-1.0/<Plattform>/client
   # ./setup
   ```

Configuration Agent

Configuration Agent ist Bestandteil verschiedener Packages, die in folgender Tabelle aufgeführt sind:

Name des Solaris-Packages	Beschreibung
SUNWapbas	Gemeinsam genutzte Konfigurationsbibliotheken
SUNWapmsc	Diverse Configuration Agent-Dateien
SUNWapoc	Configuration Agent
SUNWapdc	Configuration Agent-Assistent

Mit der Installation dieser Packages werden die für diese API erforderlichen Dateien installiert. Sie können die Packages entweder manuell oder gemeinsam mit Java Desktop System installieren. Nach der Installation müssen Sie Configuration Agent auf Ihrem System konfigurieren und aktivieren.

Hinweis – Configuration Agent-Packages werden als Teil der Solaris-Installation mit Java Desktop System installiert. Jedoch ergänzt Desktop Manager diese Dateien während der Installation, um eine korrekte Funktionsweise sicherzustellen.

Für den Zugriff auf die entfernten Konfigurationsdaten benötigt Configuration Agent einige Bootstrap-Informationen wie den Rechnernamen und den Port des LDAP-Servers. Diese Informationen werden in einem Satz Eigenschaftendateien wie `policymgr.properties`, `apocd.properties` und `os.properties` geführt. Diese Dateien sind lokal im Verzeichnis `/etc/apoc` gespeichert. Sie können diese Eigenschaften manuell bearbeiten (siehe Anhang A) oder Sie können den Konfigurationsassistenten für Configuration Agent verwenden.

Der Konfigurationsassistent bietet eine grafische Benutzeroberfläche, die Sie durch die für Configuration Agent erforderlichen Einstellungen leitet. Für jede Seite im Assistenten steht Ihnen ein Hilfefenster zur Verfügung. Sie können den Assistenten mit dem Skript `/usr/bin/apoc-config` als Superuser (root) starten.

Hinweis – Der Assistent kann auch ohne die grafische Benutzeroberfläche gestartet werden. Führen Sie beispielsweise `/usr/bin/apoc-config -nodisplay` aus, um den Assistenten im Konsolenmodus zu starten.

Startinformationen

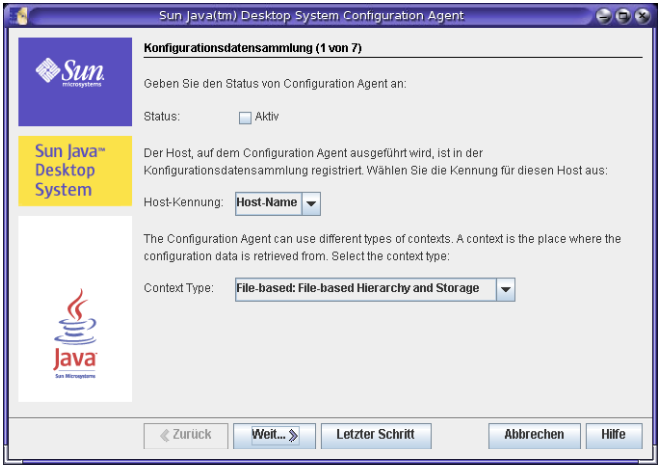

ABBILDUNG 3–1 Configuration Agent, Konfigurationsdatensammlung

---

Hinweis – Sofern zutreffend, sind die Schlüssel der entsprechenden Eigenschaftendatei in Klammern angegeben.

- Status: Status von Configuration Agent. Das Kontrollkästchen dient zum Aktivieren bzw. Deaktivieren von Configuration Agent. Auf die Konfigurationsdatensammlung kann nur zugegriffen werden, wenn Configuration Agent aktiviert ist. Die Aktivierung schließt automatisch die notwendige Registrierung bei der Dienstverwaltungsfunktion (smf(5)) von Solaris ein.
- Host-Kennung (HostIdentifierType): kann "Host-Name" oder "IP-Adresse" sein. Bei der Suche nach Host-spezifischen Richtliniendaten identifiziert Configuration Agent den aktuellen Host entweder über den Host-Namen oder die IP-Adresse. Wählen Sie entsprechend der Art und Weise, wie Ihr Host im ausgewählten Kontexttyp identifiziert wird, den korrekten Wert.
- Kontexttyp: Verwenden Sie diese Einstellung, um für Configuration Agent anzugeben, ob Ihre Organisationshierarchie und Konfigurationsdaten in LDAP oder in einer dateibasierten Speicherung oder einer Kombination aus beiden definiert sind.

Hinweis – Um Configuration Agent manuell zu aktivieren bzw. zu deaktivieren, melden Sie sich als **`root`** an und geben Sie den Befehl `/usr/lib/apoc/apocd enable` bzw. `/usr/lib/apoc/apocd disable` ein.

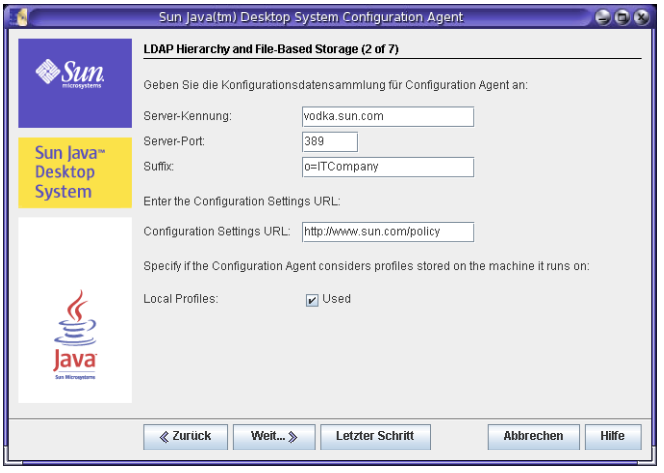

ABBILDUNG 3–2 Configuration Agent, LDAP-Hierarchie und dateibasierte Speicherung

---

Hinweis – Der Bildschirm in Abbildung 3–2 variiert je nach im vorhergehenden Bildschirm gewähltem Kontexttyp. Server-Kennung, Server-Port und Suffix sind erforderlich, wenn LDAP oder ein hybrider Kontexttyp gewählt wird. Eine URL für die Konfigurationseinstellungen ist erforderlich, wenn ein dateibasierter oder hybrider Kontexttyp gewählt wird.

- Server-Kennung: Rechnername des LDAP-Servers.
- Server-Port: Port-Nummer des LDAP-Servers.
- Suffix: Base-DN der LDAP-Datensammlung
- URL für Konfigurationseinstellungen: URL, die den Speicherort der dateibasierten Datensammlung angibt.

  Eine Liste von URLs kann dafür verwendet werden, Ersatzdatensammlungen anzugeben, falls die Verbindung mit der ersten fehlschlägt. Die Liste kann aus einer oder mehreren URLs bestehen, die durch Leerzeichen getrennt aufgeführt werden und die folgende Form haben: `file://<Dateipfad>` , `http://<host>:<Port>/<Dateipfad>` oder `https://<Host>:<Port>/<Dateipfad>`. Weitere Informationen finden Sie in Anhang A.

Hinweis – Configuration Agent versucht zunächst, mithilfe einer SSL-Verbindung auf den LDAP-Server zuzugreifen. Gelingt dies nicht, versucht Configuration Agent, eine einfache SSL-Verbindung herzustellen.

Für eine erfolgreiche SSL-Verbindung muss im Schlüsselspeicher der Java-Laufzeitumgebung das korrekte Zertifikat vorliegen. Dieser Schlüsselspeicher befindet sich bei einer Standard-JRE im Verzeichnis `<Installationsverzeichnis>/lib/security/cacerts` und bei einem Standard-JDK im Verzeichnis `<Installationsverzeichnis>/jre/lib/security/cacerts`. Diesem Speicher müssen entweder der Zertifikataussteller oder das LDAP-Serverzertifikat mithilfe des Befehls `keytool -import -file <Zertifikatdatei> -keystore <cacerts-Speicherort>` hinzugefügt werden. Das Standardpasswort für diesen Schlüsselspeicher ist **`changeit`**.

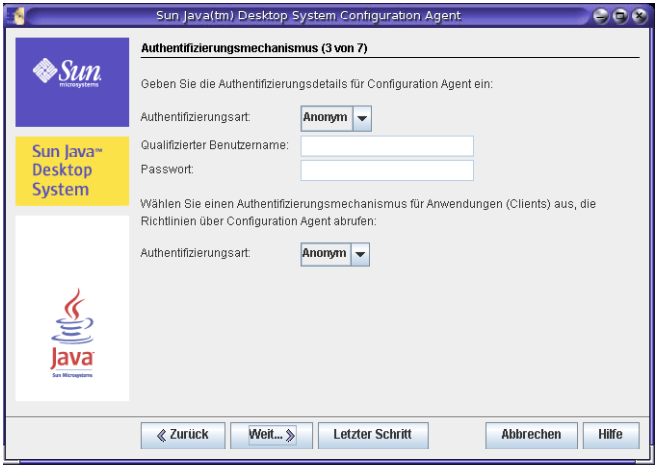

ABBILDUNG 3–3 Configuration Agent, Authentifizierungsmechanismus

- Authentifizierungsart für Configuration Agent: Entweder "Anonym" oder "Einfach". Wird "Anonym" ausgewählt, werden die Felder "Qualifizierter Benutzername" und "Passwort" automatisch deaktiviert.
- Qualifizierter Benutzername (AuthDn): vollständiger DN eines Benutzers mit Lese- und Suchberechtigung für die Datensammlung.
- Passwort (Password): Passwort eines registrierten LDAP-Benutzers

Hinweis – Wenn für das Verzeichnis der anonyme Zugriff aktiviert ist, können die Einstellungen "Qualifizierter Benutzername" und "Passwort" leer bleiben.

- Authentifizierungsart für Anwendungen (AuthType): "Anonym" oder "GSSAPI", je nach der Methode zur Benutzerauthentifizierung auf dem LDAP-Server

Hinweis – Weitere Informationen finden Sie im Abschnitt „Zugriff auf Daten/Benutzerauthentifizierung” auf Seite 25.

Port-Einstellungen

Configuration Agent verwendet zwei Ports:

- Agent-Port (DaemonPort): Port, den der Agent für die Kommunikation mit Client-Anwendungen nutzt (Standardeinstellung ist **38900**).
- Administrations-Port (DaemonAdminPort): Port, über den das Agent-Controller-Programm `apocd` mit dem Agent kommuniziert (Standardeinstellung ist **38901**).

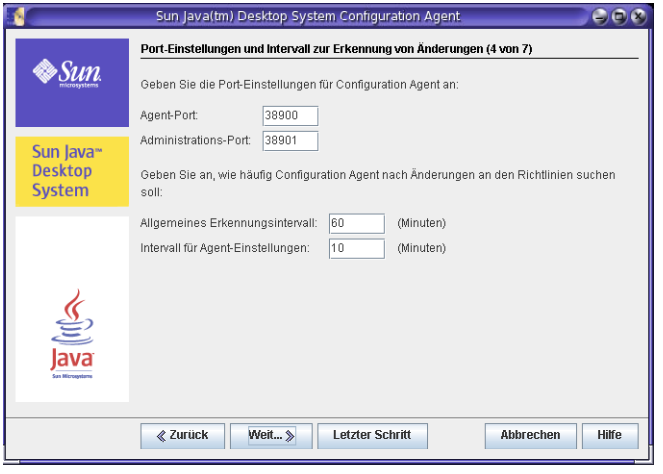

ABBILDUNG 3–4 Configuration Agent, Port-Einstellungen

Intervall zur Erkennung von Änderungen

Configuration Agent prüft die Konfigurationsdaten regelmäßig auf Änderungen. Hierbei gelten die folgenden Intervalle:

- Allgemeines Erkennungsintervall (ChangeDetectionInterval): Intervall in Minuten zwischen Zyklen zur Erkennung von Änderungen an Konfigurationsdaten der Desktop-Anwendung (des Clients).

Hinweis – Durch Angabe von **-1** wird die Änderungserkennung deaktiviert.

- Intervall für Agent-Einstellungen (DaemonChangeDetectionInterval): Intervall in Minuten zwischen Zyklen zur Erkennung von Änderungen an Agent-spezifischen Konfigurationseinstellungen.

Hinweis – Durch Angabe von **`-1`** wird die Änderungserkennung deaktiviert.

Mit dem allgemeinen Erkennungsintervall lässt sich die Weitergabe von entfernten Konfigurationsdatenänderungen an Client-seitige Anwendungen einstellen. Der für diese Einstellung festgelegte Wert bestimmt, wie viele Minuten maximal verstreichen, bevor entfernt vorgenommene Änderungen auf die Client-Anwendungen übertragen werden.

Je niedriger der Wert, desto höher die Configuration Agent- und LDAP-Server-Aktivität. Bedenken Sie dies bitte, wenn Sie einen Wert für diese Einstellungen wählen. Geschickt wäre es zum Beispiel, den Wert für die anfängliche Bereitstellungsphase auf eine Minute einzustellen, damit sich die Auswirkung der entfernten Konfiguration auf die Client-Anwendungen testen lässt, und die Einstellung nach Abschluss dieser Tests auf den Ausgangswert zurückzusetzen.

Betriebseinstellungen

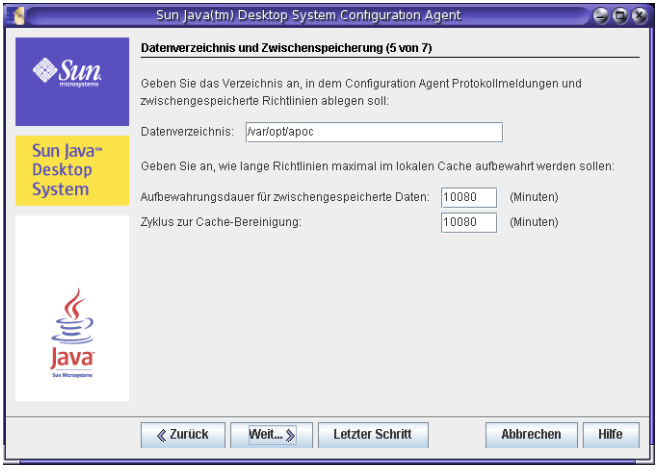

ABBILDUNG 3–5 Configuration Agent, Datenverzeichnis

Die folgenden Einstellungen können konfiguriert werden:

- Datenverzeichnis (DataDir): Verzeichnis, in dem Laufzeitdaten abgelegt werden. Das Standardverzeichnis ist **`/var/opt/apoc`**.
- Aufbewahrungsdauer für zwischengespeicherte Daten (TimeToLive): Aufbewahrungsdauer in Minuten für Nicht-Offline-Konfigurationsdaten in der lokalen Datenbank

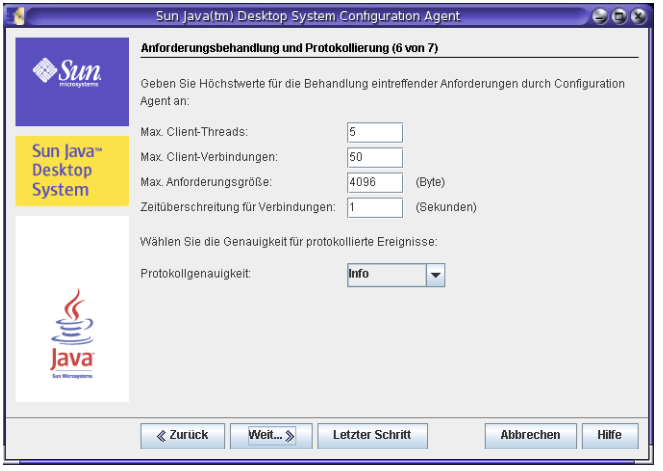

ABBILDUNG 3–6 Configuration Agent, Anforderungsbehandlung und Protokollierung

- Zyklus zur Cache-Bereinigung (GarbageCollectionInterval): Intervall in Minuten zwischen Zyklen zur Bereinigung (garbage collection) der lokalen Konfigurationsdatenbank
- Max. Client-Threads (MaxClientThreads): Höchstanzahl für Client-Anforderungen, die gleichzeitig abgearbeitet werden können.
- Max. Client-Verbindungen (MaxClientConnections): Höchstanzahl für Client-Verbindungen
- Max. Anforderungsgröße (MaxRequestSize): maximale Größe für Client-Anforderungen
- Zeitüberschreitung für Verbindungen (ConnectTimeout): zulässige Verzögerung der Reaktion des LDAP-Servers auf Verbindungsanforderungen. Das Standardintervall beträgt eine Sekunde.
- Protokollgenauigkeit (LogLevel): Genauigkeit der Agent-Protokolldateien. Die Stufen der Protokollgenauigkeit stimmen mit den Java Logger-Levels überein. Diese lauten, nach abnehmender Wichtigkeit geordnet:
  - *ERNST*
  - *WARNUNG*
  - *INFO*
  - *KONFIGURATION*
  - *DETAILS*
  - *MEHR DETAILS*
  - *MAX. DETAILS*

Hinweis – Die meisten Betriebseinstellungen außer "Datenverzeichnis" und "Zeitüberschreitung für Verbindungen" lassen sich auch zentral mithilfe entsprechender auf dem LDAP-Server gespeicherter Richtlinien verwalten. Wenn Sie dieses Leistungsmerkmal verwenden möchten, passen Sie diese Einstellungen nicht über den Assistenten an, sondern geben Sie die Betriebseinstellungen zentral mit den Configuration Agent-Richtlinien in Desktop Manager an.

Anwenden der Agent-Einstellungen

Betriebseinstellungen außer "Datenverzeichnis" und "Zeitüberschreitung für Verbindungen", die mithilfe von Desktop Manager auf dem LDAP-Server gespeichert wurden, werden beim nächsten Erkennungsintervall für Änderungen an der Agent-Konfiguration (siehe DaemonChangeDetectionInterval) automatisch wirksam.

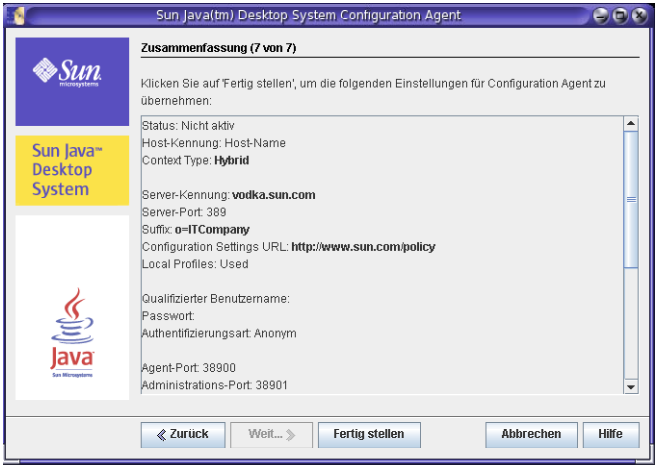

ABBILDUNG 3–7 Configuration Agent, Zusammenfassung

Für alle anderen lokal geänderten Einstellungen muss Configuration Agent neu geladen oder gestartet werden. Bei Verwendung des Konfigurationsassistenten erfolgt das erneute Laden oder der Neustart automatisch.

Hinweis – Um Configuration Agent manuell neu zu starten, vergewissern Sie sich, dass keine der dazugehörigen Client-Anwendungen ausgeführt wird. Melden Sie sich als Root an, und geben Sie den Befehl `/usr/lib/apoc/apocd restart` ein.

Zusätzliche Agent-Einstellungen

Hinweis – Die folgenden Einstellungen stehen im Konfigurationsassistenten nicht zur Verfügung.

- Anwendung lokaler Richtlinien (ApplyLocalPolicy): Diese Einstellung wird verwendet, um anzugeben, ob die auf dem lokalen Host verfügbaren Richtliniendaten Client-Anwendungen zur Verfügung gestellt werden sollten. Der Wert "true" bedeutet, dass die lokalen Richtliniendaten verfügbar gemacht werden sollten. Der Wert "false" bedeutet, dass die lokalen Richtliniendaten nicht verfügbar gemacht werden sollten. Weitere Informationen finden Sie im Abschnitt „Verwendung lokaler Richtlinien" auf Seite 24.

Verwendung lokaler Richtlinien

Sie können Configuration Agent so konfigurieren, dass Konfigurationseinstellungen von lokal bereitgestellten Richtlinien zusätzlich oder alternativ zu den global verfügbaren Richtlinien angewendet werden. Gehen Sie zum Bereitstellen solcher lokaler Richtlinien wie folgt vor:

▼ Bereitstellen lokaler Richtlinien

1. **Erstellen Sie mithilfe von Desktop Manager ein Profil mit den erforderlichen Richtlinieneinstellungen.**

2. **Exportieren Sie das Profil mithilfe von Desktop Manager in eine ZIP-Datei.**

3. **Erstellen Sie auf dem Client-Host das Verzeichnis `${DataDir}/Policies/profiles/PROFILE_REPOSITORY_default` , sofern es nicht bereits vorhanden ist.**

   `${DataDir}` entspricht dem Wert des Datenverzeichnisses von Configuration Agent, welcher standardmäßig `/var/opt/apoc` lautet.

4. **Kopieren Sie die zuvor exportierte ZIP-Datei in das Verzeichnis `${DataDir}/Policies/profiles/PROFILE_REPOSITORY_default` .**

5. **Vergewissern Sie sich, dass Configuration Agent so konfiguriert ist, dass verfügbare lokale Richtlinien angewendet werden (weitere Informationen im Abschnitt „ Zusätzliche Agent-Einstellungen" auf Seite 24).**

Hinweis – Wenn Sie die ApplyLocalPolicy-Einstellung von Configuration Agent ändern, müssen Sie Configuration Agent erneut laden, indem Sie sich als `root` anmelden und den Befehl `/usr/lib/apoc/apocd reload` eingeben.

Jede auf diese Weise bereitgestellte lokale Richtlinie wird den Clients während des nächsten Zyklus zur Erkennung von Änderungen von Configuration Agent zur Verfügung gestellt.

Automatischer Neustart von Configuration Agent

Tritt ein Fehler auf, wird Configuration Agent automatisch neugestartet. Die Dienstverwaltungsfunktion (smf(5)) ist für diese Entscheidung zuständig. Entscheidet die Dienstverwaltungsfunktion, dass ein Neustart nicht angebracht ist (beispielsweise, wenn bereits zu viele Fehler aufgetreten sind), wird Configuration Agent in den Wartungsmodus versetzt.

Wenn Configuration Agent nicht neugestartet wird, sollten Sie Configuration Agent temporär deaktivieren, indem Sie sich als root anmelden und dann den Befehl `/usr/lib/apoc/apocd disable` ausführen. Beheben Sie die Probleme, die dazu führen, dass der Agent nicht korrekt ausgeführt werden kann, und aktivieren Sie ihn erneut mit dem Befehl `/usr/lib/apoc/apocd enable`.

Zugriff auf Daten/Benutzerauthentifizierung

Configuration Agent ruft in Abhängigkeit von der Anmelde-ID des jeweiligen Desktop-Benutzers Informationen vom LDAP-Server ab. Durch die `User/UniqueIdAttribute`-Einstellung der organisatorischen Zuordnungsdatei wird die Anmelde-ID einem Benutzerelement auf dem LDAP-Server zugeordnet. Außerdem ruft Configuration Agent Informationen über den Rechner ab, wie zum Beispiel dessen Namen oder IP-Adresse. Diese Informationen werden durch die `Host/UniqueIdAttribute`-Einstellung der organisatorischen Zuordnungsdatei eines Rechnerelements auf dem LDAP-Server zugeordnet. Weitere Informationen zu Organisationszuordnungen finden Sie im Anhang C.

Es kann anonym oder mit der GSSAPI-Methode auf den LDAP-Server zugegriffen werden. Der anonyme Zugriff erfordert keinen Eingriff seitens des Desktops. Für die GSSAPI-Methode müssen auf dem Desktop Kerberos-Berechtigungsnachweise erworben werden. Damit der Kerberos-Berechtigungsnachweiserwerb in die Benutzeranmeldung integriert werden kann, muss das Modul `pam_krb5` auf dem Java Desktop System-Rechner installiert und konfiguriert sein.

Mithilfe von `gdm` lässt sich Kerberos in die Benutzeranmeldung integrieren. Verwenden Sie hierzu beispielsweise die folgende Datei `/etc/pam.d/gdm`:

```
#%PAM-1.0
auth   required    pam_unix2.so  nullok #set_secrpc
auth   optional  pam_krb5.so use_first_pass missing_keytab_ok ccache=SAFE putenv_direct
```

```
account required    pam_unix2.so
password required    pam_unix2.so  #strict=false
session required    pam_unix2.so  # trace or none
session required    pam_devperm.so
session optional    pam_console.so
```

Wenn Sie Kerberos auf diese Weise in die Benutzeranmeldung einbinden, sollten Sie die Kerberos-Unterstützung im Bildschirmschoner aktivieren. Beispielsweise durch Verwendung der folgenden Datei `/etc/pam.d/xscreensaver` :

```
auth required pamkrb5.so use_first_pass missing_keytab_ok
ccache=SAFE putenv_direct
```

Adapter

Die Anwendungsadapter sind Erweiterungen des Konfigurationssystems, die durch Desktop Manager unterstützt werden. Mithilfe der Adapter können die verschiedenen Anwendungen (je nach Konfigurationssystem) die zentralen Konfigurationsdaten berücksichtigen. Folgende Konfigurationssysteme werden unterstützt:

- GConf: Das Gnome-Konfigurationssystem wird vom Desktop und den meisten Gnome-Anwendungen, wie etwa Evolution, verwendet.
- StarOfficeRegistry: Dieses Konfigurationssystem wird von StarOffice und OpenOffice.org verwendet.
- Mozilla-Einstellungen: Dieses Konfigurationssystem wird von Mozilla verwendet.
- Java-Einstellungen: Dies ist eine Konfigurations-API, die für Java-Anwendungen bereitgestellt wird.

Es wird auch ein Desktop-Definitionsadapter bereitgestellt, welcher dem Benutzer-Desktop Desktop-Launcher, Menüelemente und Startprogramme hinzufügt.

GConf-Adapter

Der GConf-Adapter ist Teil des Pakets `SUNWapoc-adapter-gconf` für Solaris. Bei der Installation des Adapters aus dem entsprechenden packageAdapter wird der GConf-Datenquellenpfad in `/etc/gconf/2/path` aktualisiert, d. h. die Desktop Manager-Quellen werden hinzugefügt. Der Adapter stellt die folgenden beiden Datenquellen zur Verfügung:

- "apoc:readonly:": ermöglicht den Zugriff auf ungeschützte Einstellungen über die Richtlinien. Fügen Sie diese Datenquelle nach den Benutzereinstellungen und vor den lokalen Standardwerten ein.
- "apoc:readonly:mandatory@": ermöglicht den Zugriff auf geschützte Einstellungen über die Richtlinien. Fügen Sie diese Datenquelle nach den obligatorischen lokalen Einstellungen und vor den Benutzereinstellungen ein.

Konfiguration des GConf-Adapters

Der GConf-Adapter wird während seiner Installation konfiguriert, jedoch hängt sein Betrieb von der Gegenwart von zwei Datenquellen, die die obligatorischen zentralen Einstellungen und die Standardeinstellungen repräsentieren, in der GConf-Pfaddatei (`/etc/GConf/2/path`) ab. Diese Pfaddatei enthält die korrekten Informationen, damit GConf die zentralen Einstellungen wie erwartet nach der Installation des Systems berücksichtigt. Zugleich sollten Administratoren sicherstellen, dass die Datenquellen mit dem Präfix "apoc" noch in der Datei vorhanden sind, für den Fall, dass sie diesen Pfad für zusätzliche benutzerdefinierte Datenquellen ändern müssen. Sie sollten auch sicherstellen, dass sich die Datenquellen zwischen den lokalen obligatorischen Einstellungen und den Benutzereinstellungen für die Datenquelle befinden, welche die obligatorischen zentralen Einstellungen repräsentiert, und zwischen den Benutzereinstellungen und den lokalen Standardeinstellungen für die Datenquelle, welche die standardmäßigen zentralen Einstellungen repräsentiert.

Java-Einstellungen-Adapter

Der Java-Einstellungen-Adapter ist Teil des Pakets `SUNWapcj` für Solaris.

Konfiguration des Java-Einstellungen-Adapters

Der Java-Einstellungen-Adapter wird als Implementierung der Einstellungen-API bereitgestellt, die als Wrapper für eine andere vorhandene Implementierung (wie das mit der JRE gelieferte standardmäßige dateibasierte System) verwendet werden muss. Für die Aktivierung der zentralen Konfiguration in einer Java-Anwendung, die die Einstellungen-API verwendet, muss ein Startskript für diese Anwendung geschrieben werden. Dabei muss das Skript `/usr/lib/apoc/apocjlaunch` als Hilfsprogramm verwendet werden. Dieses Skript muss einige Umgebungsvariablen definieren und dann an seinem Ende das Skript `apocjlaunch` enthalten, welches die Java-Anwendung mit der notwendigen Umgebung startet. Folgende Umgebungsvariablen müssen eingestellt werden:

- `JAVA`: Enthält den Pfad zur ausführbaren Datei der Java-Laufzeit
- `APPLICATION`: Enthält den nachgestellten Teil des regulären Aufrufs der Java-Laufzeit für diese Anwendungen. Beispielsweise *Klassendatei [Argumente]* für den Start einer einzelnen Klasse oder `-jar` *JAR-Datei [Argumente]* für den Start eines JAR-Archivs.

Folgende optionalen zusätzlichen Umgebungsvariablen können eingestellt werden:

- `CLASSPATH`: Eine durch Kommas getrennte Liste von JAR- oder Klassendateien, die Teil eines Anwendungsklassenpfads sein müssen
- `DEFINES`: String, der die Anweisungen enthält, die Teil des Anwendungsstarts sein müssen
- `PREFFACTORY`: Klassenname der Factory in der zugrundeliegenden Implementierung der Einstellungen-API, die die Anwendung verwenden muss

Mozilla-Adapter

Der Mozilla-Adapter ist Teil des Pakets `SUNWmozapoc-adapter` auf Solaris.

Konfiguration des Mozilla-Adapters

Der Mozilla-Adapter wird während der Installation dieses Produkts eingerichtet und bedarf keiner zusätzlichen Konfiguration.

StarOffice-Adapter

Der StarOffice-Adapter ist in Standardinstallationen von StarOffice enthalten und ermöglicht den Zugriff auf Profilkonfigurationsdaten, ohne dass Sie spezielle Änderungen vornehmen müssen.

Konfiguration des StarOffice-Adapters

Der StarOffice-Adapter wird während der Installation dieses Produkts eingerichtet und bedarf keiner zusätzlichen Konfiguration.

Desktop-Definitionsadapter

Der Desktop-Definitionsadapter besteht aus folgenden Paketen:

Package-Name	Beschreibung
SUNWapleg	Konfiguration von Zugriffsbinärdateien
SUNWardsa	Desktop-Definitionsadapter
SUNWardsa-misc	Systemintegration für Adapter

Diese Pakete werden bei der Installation der Desktop Manager-Client-Komponenten installiert und bedürfen keiner weiteren Konfiguration.

Konfiguration des Desktop-Definitionsadapters

Der Desktop-Definitionsadapter wird vom Installationsvorgang so konfiguriert, dass er immer dann verwendet werden kann, wenn sich ein Benutzer anmeldet. Er bedarf keiner weiteren Konfiguration.

Entfernen von Adaptern

Der Mozilla- und der StarOffice-Adapter werden entfernt, wenn die zugehörigen Produkte entfernt werden. Der GConf-, Java-Einstellungen- und Desktop-Definitionsadapter können mithilfe der entsprechenden System-Tools für die Paketverwaltung entfernt werden, indem die im Abschnitt über die Installation erwähnten Pakete entfernt werden.

Nach dem Entfernen des Java-Einstellungen-Adapters, sollten die für das Starten von Java-Anwendungen geschriebenen Startskripte, die die Einstellungen-API verwenden, nicht mehr verwendet werden. Ein darin vorgenommener Java-Aufruf schlägt fehl, da einige der benötigten Klassen nicht mehr verfügbar sind.

Problemlösung für Adapter

Die meisten der Probleme, die dazu führen können, dass die zentralen Konfigurationsdaten in den entsprechenden Anwendungen nicht zu sehen sind, werden mit hoher Wahrscheinlichkeit von Configuration Agent verursacht, da diese Anwendung von allen Adaptern zum Abrufen von Daten verwendet wird.

Wenn eine zentrale Konfigurationsänderung keine Auswirkung auf eine bestimmte Einstellung (oder eine Gruppe von Einstellungen) zu haben scheint, ist eine mögliche Erklärung hierfür, dass der Benutzer für diese Einstellung in der Anwendung explizit einen Wert festgelegt hat (normalerweise über die Dialogfelder für Optionen oder Einstellungen im jeweiligen Produkt). In diesem Fall hat die Benutzereinstellung Vorrang vor dem mithilfe von Desktop Manager eingestellten Wert, es sei denn die zentralen Einstellungen sind als geschützt definiert, was bedeutet, dass dieser Wert durch den Administrator erzwungen wird und der Benutzer diesen Wert nicht ändern darf.

Configuration Agent – Problembehebung

Dieser Abschnitt geht auf einige Fragen ein, die Sie möglicherweise zur Funktionsweise von Configuration Agent haben, und gibt Tipps zur Behebung von Problemen bei der Arbeit mit Configuration Agent.

Fragen und Antworten

Was ist Configuration Agent und wie funktioniert diese Anwendung?

Configuration Agent ist eine Anwendung, die Richtlinien speichert und bereitstellt. Sie stellt sicher, dass Desktop-Client-Anwendungen zentral konfiguriert werden können, ohne dass die Leistung dieser Anwendungen auf den Hosts beträchtlich gemindert wird. Dies wird durch folgende Funktionen erreicht:

- Zwischenspeichern aller heruntergeladenen Richtlinien in einem lokalen Cache für die Verwendung durch Clients
- Gemeinsames Nutzen teurer Ressourcen, bei denen dies möglich und sinnvoll ist (beispielsweise Verbindungen mit einem LDAP-Server, auf dem sich die Richtlinie befindet)

Der typische Anwendungsfall, bei dem die Interaktion zwischen den Client-Anwendungen und Configuration Agent abläuft, ist sehr einfach und kann wie folgt beschrieben werden:

1. Ein Benutzer startet eine der relevanten Desktop-Client-Anwendungen (gconfd, Mozilla oder StarOffice).
2. Die Client-Anwendung stellt eine Verbindung mit Configuration Agent her.
3. Die Client-Anwendung fordert von Configuration Agent die erforderlichen Richtliniendaten an.
4. Configuration Agent durchsucht seinen Cache nach den angeforderten Richtliniendaten.

5. Werden die Richtliniendaten nicht im Cache gefunden, lädt Configuration Agent die erforderlichen Daten aus einer zuvor konfigurierten Datensammlung herunter und speichert sie im Cache.
6. Die Richtliniendaten werden an die entsprechende Client-Anwendung gesendet.
7. Configuration Agent überwacht die Richtliniensammlung auf Änderungen an den Richtliniendaten.
8. Wird eine Änderung erkannt, aktualisiert Configuration Agent seinen Cache-Speicher, sodass dieser auf dem neuesten Stand ist, und benachrichtigt die Client-Anwendung von der Änderung.

Wo erhalte ich Configuration Agent und wie installiere ich die Anwendung?

Configuration Agent ist standardmäßig in Solaris 10 enthalten und wird mit diesem installiert.

Ich habe die Installation von Solaris 10 abgeschlossen. Wie muss ich nun vorgehen?

Configuration Agent ist standardmäßig deaktiviert und nicht konfiguriert. Um Configuration Agent zu verwenden, müssen Sie eine Mindestkonfiguration durchführen und die Anwendung aktivieren. Nach Durchführung dieser Schritte wird automatisch die entsprechende Desktop-Client-Anwendung gestartet, sodass sie beim nächsten Starten automatisch die von Ihnen bereitgestellte Richtlinie verwenden.

Ich möchte Configuration Agent konfigurieren. Wie muss ich vorgehen?

Die korrekte Konfiguration von Configuration Agent erfolgt mit dem entsprechenden Assistenten (Configuration Agent Wizard). Diesen können Sie (als Root-Benutzer) mit dem Befehl `/usr/bin/apoc-config` starten. Configuration Agent Wizard führt Sie durch die Schritte, die zur korrekten Konfiguration von Configuration Agent erforderlich sind. In den meisten Fällen ist die einzige Angabe, die zum Abschluss des Assistenten erforderlich ist, der Speicherort der Richtliniensammlung.

Sie können Configuration Agent auch durch manuelles Bearbeiten der entsprechenden Konfigurationsdateien konfigurieren. Dies wird jedoch nicht empfohlen, da auf diese Weise eher eine Fehlkonfiguration vorgenommen wird. Außerdem prüft Configuration Agent Wizard, ob die vorgenommene Änderung an der Konfiguration einen Neustart oder ein erneutes Laden von Configuration Agent erfordert.

Ich möchte Configuration Agent aktivieren. Wie muss ich vorgehen?

Configuration Agent kann auf drei Arten aktiviert werden:

1. Setzen Sie den Status von Configuration Agent mithilfe von Configuration Agent Wizard (`/usr/bin/apoc-config`) auf "Aktiv".

2. Führen Sie als Root-Benutzer den folgenden Befehl aus, um das Configuration Agent-Steuerprogramm (`/usr/lib/apoc/apocd`) zu verwenden:

   ```
   /usr/lib/apoc/apocd enable
   ```

3. Führen Sie als Superuser den folgenden Befehl aus, um smf(5) zu verwenden:

   ```
   /usr/sbin/svcadm enable svc:/network/apocd/udp
   ```

Ich habe Configuration Agent konfiguriert und installiert. Wie kann ich feststellen, ob die Anwendung korrekt arbeitet?

Die einfachste Art, die korrekte Funktionsweise von Configuration Agent zu prüfen, ist das Erstellen einer Richtlinie mit Desktop Manager und das Zuweisen der Richtlinie zu einem Benutzer. Nun können Sie sich am Desktop-Rechner als dieser Benutzer anmelden und überprüfen, ob die von Ihnen vorgenommenen Einstellungen verwendet werden. Hierzu eignen sich Richtlinieneinstellungen, deren Änderungen in einer Desktop-Sitzung einfach erkannt werden können, beispielsweise der Hintergrund oder das Motiv.

Welche Bedeutung hat das Aktivieren von Configuration Agent?

Configuration Agent ist ein smf(5)-kompatibler Dienst und wird daher für smf(5) aktiviert. Nach der Aktivierung von Configuration Agent kann die Anwendung eingesetzt werden. Beim Aktivieren von Configuration Agent geschieht Folgendes:

- Configuration Agent wird gestartet.
- Alle Desktop-Client-Anwendungen, die nach der Aktivierung von Configuration Agent gestartet werden, können Richtliniendaten abrufen.
- Configuration Agent wird beim Neustart des Systems automatisch erneut gestartet.

Wie kann ich feststellen, ob Configuration Agent aktiviert ist?

Es gibt mehrere Möglichkeiten festzustellen, ob Configuration Agent aktiviert ist:

- Verwenden Sie das Steuerprogramm für Configuration Agent. Melden Sie sich als Superuser an und geben Sie den folgenden Befehl ein:

  ```
  /usr/lib/apoc/apocd is-enabled
  ```

  Ist Configuration Agent aktiviert, gibt das Steuerprogramm die folgende Meldung aus:

  ```
  Checking Configuration Agent enabled status ... Enabled
  ```

  Anderenfalls gibt es die folgende Meldung aus:

  ```
  Checking Configuration Agent enabled status ... Not enabled
  ```

- Führen Sie mit smf(5) den folgenden Befehl aus:

  ```
  /usr/bin/svcs svc:/network/apocd/udp:default
  ```

  Ist Configuration Agent aktiviert, gibt svcs die folgende Meldung aus:

```
STATE          STIME    FMRI
online          8:36:04 svc:/network/apocd/udp:default
```

Ist Configuration Agent deaktiviert, gibt svcs die folgende Meldung aus:

```
STATE          STIME    FMRI
disabled       15:58:34 svc:/network/apocd/udp:default
```

Befindet sich Configuration Agent im Wartungsmodus, gibt svcs die folgende Meldung aus:

```
STATE          STIME    FMRI
maintenance     8:38:42 svc:/network/apocd/udp:default
```

Wie kann ich feststellen, ob Configuration Agent gerade ausgeführt wird?

Sie haben mehrere Möglichkeiten festzustellen, ob Configuration Agent läuft:

- Melden Sie sich als Superuser an und führen Sie das Steuerprogramm für Configuration Agent aus:

  ```
  /usr/lib/apoc/apocd status
  ```

  Ist Configuration Agent aktiviert, gibt das Steuerprogramm die folgende Meldung aus:

  ```
  Checking Configuration Agent status ... Running
  ```

  Anderenfalls gibt das Steuerprogramm die folgende Meldung aus:

  ```
  Checking Configuration Agent status ... Not running
  ```

- Führen Sie folgenden Befehl aus:

  ```
  /usr/bin/svcs svc:/network/apocd/udp:default
  ```

  Wird Configuration Agent gerade ausgeführt, gibt svcs die folgende Meldung aus:

  ```
  STATE          STIME    FMRI
  online          8:36:04 svc:/network/apocd/udp:default
  ```

  Wird Configuration Agent nicht ausgeführt, gibt svcs die folgende Meldung aus:

  ```
  STATE          STIME    FMRI
  disabled       15:58:34 svc:/network/apocd/udp:default
  ```

  Befindet sich Configuration Agent im Wartungsmodus, gibt svcs die folgende Meldung aus:

  ```
  STATE          STIME    FMRI
  maintenance     8:38:42 svc:/network/apocd/udp:default
  ```

- Führen Sie folgenden Befehl aus:

  ```
  ps -ef | grep apoc
  ```

  Wird Configuration Agent ausgeführt, sollte die Ausgabe des Befehls `ps` den folgenden Java-Prozess enthalten:

```
 daemon 29295 29294  0 13:05:22?  0:03 java -Djava.library.path=/usr/lib/apoc
 -cp /usr/share/lib/apoc/apocd.jar:/usr/s
 daemon 29294   1   0 13:05:22?  0:00 sh -c java
-Djava.library.path=/usr/lib/apoc -cp /usr/share/lib/apoc/apocd.jar:
 root 29345 28134  0 13:08:59 pts/1   0:00 grep apoc
```

Wo sind die Protokolldateien?

Sie können zur Problemsuche für Configuration Agent die folgenden Protokolldateien zurate ziehen:

- smf(5)-Protokolldateien:
  - In der Datei `/var/svc/log/network-apocd-udp:default.log` werden alle Ereignisse aufgezeichnet, die sich auf das Starten und Anhalten von Configuration Agent-Instanzen beziehen. Diese Datei enthält außerdem die Meldungen, die das Configuration Agent-Steuerprogramm (`/usr/lib/apoc/apocd`) in seine Standardausgabe schreibt, und die Ausgabemeldungen von JVM oder Configuration Agent.
  - Die Protokolldatei `/var/svc/log/svc.startd.log` zeichnet smf(5)-Ereignisse höherer Ebene auf. Schlagen mehrere kurz aufeinander folgende Startversuche Configuration Agent fehl, entscheidet smf(5) möglicherweise, dass Configuration Agent nicht gestartet werden kann. In diesem Fall versetzt smf(5) Configuration Agent in den Wartungsmodus und schreibt einen entsprechenden Eintrag in das Protokoll.

  Diese beiden Protokolldateien sind in der Regel nützlich, wenn Sie Probleme beim Starten von Configuration Agent haben.
- Configuration Agent-Protokolle:

  Configuration Agent schreibt die Protokollmeldungen in Protokolldateien im Standardverzeichnis für Protokolle (`/var/opt/apoc/Logs`). Das "Datenverzeichnis" für Configuration Agent ist `/var/opt/apoc` . Sie können den Speicherort für dieses Verzeichnis mithilfe von Configuration Agent Wizard (`/usr/bin/apoc-config`) ändern. Wie detailliert die Protokollmeldungen sind, können Sie durch Ändern der "Protokollgenauigkeit" mithilfe von Configuration Agent Wizard anpassen. Wenn Sie glauben, dass Sie Configuration Agent nicht korrekt konfiguriert haben, oder falls Sie andere Probleme mit der Anwendung haben, können Sie die Protokollgenauigkeit mit Configuration Agent Wizard auf "Max. Details" stellen, bevor Sie die Protokolldateien der Anwendung auswerten. So stellen Sie sicher, dass Sie die größtmögliche Menge an Protokollinformationen erhalten.
- Systemprotokolle:

  Sie können außerdem die Protokolldateien `/var/adm/messages` und `/var/opt/SUNWut/log/messages` (SunRay) prüfen, um eine Diagnose von Problemen mit Configuration Agent durchzuführen.

Wie kann ich detailliertere Protokolle erhalten?

Siehe „Wo sind die Protokolldateien?” auf Seite 33

Was ist der Wartungsmodus?

smf(5) versetzt Configuration Agent in den Wartungsmodus, wenn es Probleme beim Starten oder erneuten Starten von Configuration Agent erkennt. Kann smf(5) Configuration Agent nicht starten, unternimmt es mehrere Versuche, bis der Start erfolgreich ist oder smf(5) entscheidet, dass Configuration Agent nicht gestartet werden kann. In letzterem Fall versetzt smf(5) Configuration Agent in den Wartungsmodus, um anzuzeigen, dass Sie die erkannten Probleme beheben müssen. Wurden die Probleme behoben, können Sie den smf(5)-Status von Configuration Agent aufheben, um zum Normalbetrieb zu wechseln.

Wie kann ich den Wartungsmodus verlassen und/oder den smf(5)-Status aufheben?

Melden Sie sich als Superuser an und führen Sie den Befehl `/usr/sbin/svcadm clear svc:/network/apocd/udp` aus.

Was geschieht, wenn Configuration Agent unerwartet beendet wird?

smf(5) erkennt, dass die Anwendung nicht mehr ausgeführt wird und versucht, sie erneut zu starten. Sollten mehrere aufeinander folgende Versuche fehlschlagen, versetzt smf(5) Configuration Agent in den Wartungsmodus. Laufende Desktop-Client-Anwendungen sind nicht betroffen, wenn Configuration Agent erfolgreich erneut gestartet wird. Diese Client-Anwendungen stellen automatisch erneut eine Verbindung zu Configuration Agent her, wenn diese Anwendung neu gestartet wird.

Muss ich Desktop-Client-Anwendungen neu starten, wenn ich Configuration Agent aktiviere bzw. starte?

Welche Maßnahmen Sie ergreifen, hängt davon ab, ob Configuration Agent zum Zeitpunkt des Starts der entsprechenden Desktop-Client-Anwendung aktiviert war und ausgeführt wurde. Wurde Configuration Agent aktiviert und wird die Anwendung ausgeführt, stellte die Client-Anwendung eine Verbindung mit Configuration Agent her und versucht, die Verbindung bei einer Trennung wiederherzustellen. Das heißt, dass die Client-Anwendungen bei jedem Starten, Aktivieren oder Deaktivieren von Configuration Agent versuchen, die Verbindung mit Configuration Agent wiederherzustellen, sobald diese Anwendung wieder ausgeführt wird. War Configuration Agent beim Starten der Client-Anwendung nicht aktiviert bzw. wurde nicht ausgeführt, verwendet die Client-Anwendung Configuration Agent nicht und versucht auch nicht, beim Start von Configuration Agent eine Verbindung herzustellen. Dies führt zu folgendem Verhalten:

- Desktop-Client-Anwendungen, die gestartet wurden, während Configuration Agent aktiviert war und ausgeführt wurde, müssen nicht erneut gestartet werden.
- Desktop-Client-Anwendungen, die gestartet wurden, während Configuration Agent nicht aktiviert war bzw. nicht ausgeführt wurde, müssen erneut gestartet werden.

Meine Desktop-Client-Anwendungen scheinen nicht die konfigurierten Richtlinien zu verwenden. Was kann ich tun?

Das gängigste Problem im Zusammenhang mit Configuration Agent ist, dass die Auswirkungen von konfigurierten Richtlinien für Desktop-Client-Anwendungen nicht gesehen werden können. Die häufigsten Ursachen hierfür sind eine fehlerhafte Konfiguration von Configuration Agent, eine fehlerhafte Konfiguration der Richtliniensammlung und eine nicht erreichbare Richtliniensammlung. Die folgenden Schritte unterstützen Sie beim Auffinden und Beheben von Problemen:

- Stellen Sie sicher, dass Configuration Agent konfiguriert ist.
- Stellen Sie sicher, dass Configuration Agent aktiviert ist und ausgeführt wird. Wenn Sie Configuration Agent starten müssen, müssen Sie auch die aktuell geöffneten Desktop-Client-Anwendungen erneut starten.
- Bestehen die Probleme weiterhin, erhöhen Sie vorübergehend die Detailstufe der Configuration Agent-Protokolle und starten Sie Configuration Agent nach Möglichkeit erneut, sodass Sie ein vollständiges und detailliertes Protokoll ab dem Startzeitpunkt von Configuration Agent haben.
- Wird Configuration Agent nicht korrekt gestartet, lesen Sie den Abschnitt „Probleme beim Starten von Configuration Agent” auf Seite 35.
- Wird Configuration Agent korrekt gestartet, aber die Desktop-Client-Anwendungen verwenden eine verfügbare Richtlinie nicht, lesen Sie den Abschnitt "Probleme beim Abrufen einer Richtlinie von Configuration Agent".
- Sollte Ihr Problem weiterhin bestehen, wenden Sie sich an den technischen Support.

Probleme beim Starten von Configuration Agent

Kann Configuration Agent nicht gestartet werden, obwohl Sie sicher sind, dass Sie Configuration Agent konfiguriert und aktiviert haben, müssen Sie die Protokolldateien prüfen. In den folgenden Abschnitten werden die häufigsten Fehler erläutert, die dieses Problem verursachen können.

Ungültiges oder nicht verfügbares Agent-Datenverzeichnis

Das Configuration Agent-Datenverzeichnis wird von Configuration Agent erstellt und zum Speichern von Protokolldateien, Richtlinien-Caches usw. verwendet. Der standardmäßige Pfad für dieses Verzeichnis ist `/var/opt/apoc`.

Configuration Agent gibt die folgende Meldung in den smf(5)-Protokollen aus, wenn als Pfad für das Datenverzeichnis ein nicht verfügbarer Speicherort (`/dev/null/cant/write/here`) bestimmt wurde. Um dieses Problem zu beheben, können Sie mithilfe von Configuration Agent Wizard (`/usr/bin/apoc-config`) einen verfügbaren Speicherort als Datenverzeichnis festlegen.

```
[ Nov 17 14:35:38 Executing start method ("/usr/lib/apoc/apocd svcStart") ]
Starting Configuration Agent ... Warning: Cannot create Log directory
   '/dev/null/cant/write/here/Logs'
Warning:Failed to create log file handler
Nov 17, 2005 2:35:39 PM com.sun.apoc.daemon.misc.APOCLogger config
```

```
CONFIG: Daemon configuration:
 MaxRequestSize = 4096
 DaemonAdminPort = 38901
 ThreadTimeToLive = 5
 DaemonChangeDetectionInterval = 10
 IdleThreadDetectionInterval = 15
 PROVIDER_URL =
 DataDir = /dev/null/cant/write/here
 ApplyLocalPolicy = true
 ChangeDetectionInterval = 60
 MaxClientConnections = 50
 GarbageCollectionInterval = 10080
 InitialChangeDetectionDelay = 10
 TimeToLive = 10080
 ConnectionReadTimeout = 5000
 DaemonPort = 38900
 LogLevel = FINEST
 MaxClientThreads = 5


Nov 17, 2005 2:35:39 PM Daemon main
FINER: THROW
com.sun.apoc.daemon.misc.APOCException
    at com.sun.apoc.daemon.apocd.Daemon.initAuthDir(Unknown Source)
    at com.sun.apoc.daemon.apocd.Daemon.init(Unknown Source)
    at com.sun.apoc.daemon.apocd.Daemon.<init>(Unknown Source)
    at com.sun.apoc.daemon.apocd.Daemon.main(Unknown Source)
[ Nov 17 14:36:08 Method or service exit timed out.  Killing contract 980 ]
[ Nov 17 14:36:08 Method "start" failed due to signal KILL ]
```

Verwenden eines bereits belegten Anforderungs-Ports

Configuration Agent kommuniziert über TCP/IP-Socketverbindungen mit Desktop-Client-Anwendungen. Diese Verbindungen werden standardmäßig über Port 38900 hergestellt.

Die folgende Fehlermeldung wird ausgegeben, wenn Configuration Agent für den Port 1234 konfiguriert wird und dieser bereits von einem anderen Dienst verwendet wird. Die Fehlermeldung wird in den Configuration Agent-Protokollen aufgezeichnet. Um dieses Problem zu beheben, können Sie die Einstellung "Agent-Port" mithilfe von Configuration Agent Wizard (`/usr/bin/apoc-config`) auf einen nicht verwendeten Port einstellen.

```
Nov 17, 2005 2:50:59 PM com.sun.apoc.daemon.misc.APOCLogger config
CONFIG: Daemon configuration:
 MaxRequestSize = 4096
 DaemonAdminPort = 38901
 ThreadTimeToLive = 5
 DaemonChangeDetectionInterval = 10
```

```
 IdleThreadDetectionInterval = 15
 PROVIDER_URL =
 DataDir = /var/opt/apoc
 ApplyLocalPolicy = true
 ChangeDetectionInterval = 60
 MaxClientConnections = 50
 GarbageCollectionInterval = 10080
 InitialChangeDetectionDelay = 10
 TimeToLive = 10080
 ConnectionReadTimeout = 5000
 DaemonPort = 1234
 LogLevel = FINEST
 MaxClientThreads = 5


Nov 17, 2005 2:50:59 PM com.sun.apoc.daemon.misc.APOCLogger info
INFO: Daemon starting
Nov 17, 2005 2:50:59 PM com.sun.apoc.daemon.misc.APOCLogger fine
FINE: Garbage collection scheduled ( interval = 10080 minutes )
Nov 17, 2005 2:50:59 PM Daemon main
FINER: THROW
com.sun.apoc.daemon.misc.APOCException: java.net.BindException: Address already in use
    at com.sun.apoc.daemon.transport.ChannelManager.<init>(Unknown Source)
    at com.sun.apoc.daemon.apocd.Daemon.run(Unknown Source)
    at com.sun.apoc.daemon.apocd.Daemon.main(Unknown Source)
Caused by: java.net.BindException: Address already in use
    at sun.nio.ch.Net.bind(Native Method)
    at sun.nio.ch.ServerSocketChannelImpl.bind(ServerSocketChannelImpl.java:119)
    at sun.nio.ch.ServerSocketAdaptor.bind(ServerSocketAdaptor.java:59)
    at sun.nio.ch.ServerSocketAdaptor.bind(ServerSocketAdaptor.java:52)
```

Verwenden eines bereits belegten Administrations-Ports

Configuration Agent kommuniziert über TCP/IP-Socketverbindungen mit dem Configuration Agent-Steuerprogramm (`/usr/lib/apoc/apocd`). Diese Verbindungen werden standardmäßig über Port 38901 hergestellt.

Die folgende Fehlermeldung wird in den Configuration Agent-Protokollen aufgezeichnet, wenn Configuration Agent für den Port 1234 konfiguriert wird und dieser bereits von einem anderen Dienst verwendet wird. Um dieses Problem zu beheben, können Sie die Einstellung "Administrations-Port" mithilfe von Configuration Agent Wizard (`/usr/bin/apoc-config`) auf einen nicht verwendeten Port einstellen.

```
ONFIG: Daemon configuration:
 MaxRequestSize = 4096
 DaemonAdminPort = 1234
 ThreadTimeToLive = 5
 DaemonChangeDetectionInterval = 10
```

```
IdleThreadDetectionInterval = 15
PROVIDER_URL =
DataDir = /var/opt/apoc
ApplyLocalPolicy = true
ChangeDetectionInterval = 60
MaxClientConnections = 50
GarbageCollectionInterval = 10080
InitialChangeDetectionDelay = 10
TimeToLive = 10080
ConnectionReadTimeout = 5000
DaemonPort = 38900
LogLevel = FINEST
MaxClientThreads = 5


Nov 17, 2005 2:55:11 PM com.sun.apoc.daemon.misc.APOCLogger info
INFO: Daemon starting
Nov 17, 2005 2:55:11 PM com.sun.apoc.daemon.misc.APOCLogger fine
FINE: Garbage collection scheduled ( interval = 10080 minutes )
Nov 17, 2005 2:55:11 PM com.sun.apoc.daemon.misc.APOCLogger fine
FINE: Client manager started
Nov 17, 2005 2:55:11 PM com.sun.apoc.daemon.misc.APOCLogger fine
FINE: Channel manager started
Nov 17, 2005 2:55:11 PM Daemon main
FINER: THROW
com.sun.apoc.daemon.misc.APOCException: java.net.BindException: Address already in use
    at com.sun.apoc.daemon.admin.AdminManager.initChannel(Unknown Source)
    at com.sun.apoc.daemon.admin.AdminManager.<init>(Unknown Source)
    at com.sun.apoc.daemon.apocd.Daemon.run(Unknown Source)
    at com.sun.apoc.daemon.apocd.Daemon.main(Unknown Source)
Caused by: java.net.BindException: Address already in use
    at sun.nio.ch.Net.bind(Native Method)
    at sun.nio.ch.ServerSocketChannelImpl.bind(ServerSocketChannelImpl.java:119)
    at sun.nio.ch.ServerSocketAdaptor.bind(ServerSocketAdaptor.java:59)
    at sun.nio.ch.ServerSocketAdaptor.bind(ServerSocketAdaptor.java:52)
    ... 4 more
```

Probleme beim Abrufen einer Richtlinie von Configuration Agent

Fehlende oder ungültige Angabe von Konfigurationsdatensammlung

Configuration Agent muss eine Verbindung zu einer gültigen Konfigurationsdatensammlung herstellen, um die Richtlinieninformationen herunterzuladen und im Cache zu speichern. Ist die Angabe der Konfigurationsdatensammlung in der Konfiguration von Configuration Agent nicht korrekt (beispielsweise durch Verwendung eines ungültigen Formats oder Unterlassen der Angabe), werden beim Start der Desktop-Client-Anwendungen in den Configuration Agent-Protokollen

Fehler wie die folgenden aufgezeichnet. Um dieses Problem zu beheben, können Sie mithilfe von Configuration Agent Wizard (`/usr/bin/apoc-config`) die zu verwendende Konfigurationsdatensammlung angeben.

```
FINER: New client added
Nov 18, 2005 1:59:22 PM com.sun.apoc.daemon.misc.APOCLogger finer
FINER: CreateSession transaction started
Nov 18, 2005 1:59:22 PM com.sun.apoc.daemon.misc.APOCLogger finer
FINER: Creating new client session
Nov 18, 2005 1:59:22 PM com.sun.apoc.daemon.misc.APOCLogger finest
FINEST: Authenticating user geoffh
Nov 18, 2005 1:59:22 PM com.sun.apoc.daemon.misc.APOCLogger finest
FINEST: Authentication successful
Nov 18, 2005 1:59:23 PM PolicyBackend openPolicyBackend
FINER: THROW
com.sun.apoc.daemon.misc.APOCException: com.sun.apoc.daemon.misc.APOCException:
com.sun.apoc.spi.environment.InvalidParameterException: The parameter organisation
PROVIDER_URL#protocol (null) is not valid, the value must be comprised in
{ldaps,ldap,https,http,file}.
    at com.sun.apoc.daemon.apocd.PolicyBackend.<init>(Unknown Source)
    at com.sun.apoc.daemon.apocd.HostPolicyBackend.<init>(Unknown Source)
    at com.sun.apoc.daemon.apocd.PolicyBackendFactory.openPolicyBackend(Unknown Source)
    at com.sun.apoc.daemon.apocd.Cache$DataSource.openPolicyBackend(Unknown Source)
    at com.sun.apoc.daemon.apocd.Cache$DataSource.open(Unknown Source)
    at com.sun.apoc.daemon.apocd.Cache.createDataSources(Unknown Source)
    at com.sun.apoc.daemon.apocd.Cache.<init>(Unknown Source)
    at com.sun.apoc.daemon.apocd.CacheFactory.createNewCache(Unknown Source)
    at com.sun.apoc.daemon.apocd.CacheFactory.openCache(Unknown Source)
    at com.sun.apoc.daemon.apocd.Session.<init>(Unknown Source)
    at com.sun.apoc.daemon.transaction.CreateSessionTransaction.executeTransaction
(Unknown Source)
    at com.sun.apoc.daemon.transaction.Transaction.execute(Unknown Source)
    at com.sun.apoc.daemon.apocd.ClientEventHandler.handleEvent(Unknown Source)
    at com.sun.apoc.daemon.apocd.EventWorkerThread.run(Unknown Source)
Caused by: com.sun.apoc.daemon.misc.APOCException:
 com.sun.apoc.spi.environment.InvalidParameterException:
 The parameter organisation PROVIDER_URL#protocol (null) is not valid,
 the value must be comprised in {ldaps,ldap,https,http,file}.
    at com.sun.apoc.daemon.apocd.PolicyBackendFactory.openPolicyMgr(Unknown Source)
    ... 14 more
Caused by: com.sun.apoc.spi.environment.InvalidParameterException: The parameter
organisation PROVIDER_URL#protocol (null) is not valid, the value must be comprised in
{ldaps,ldap,https,http,file}.
    at com.sun.apoc.spi.PolicyMgrFactoryImpl.createPolicyMgr(Unknown Source)
    ... 15 more
Nov 18, 2005 1:59:23 PM PolicyBackend openPolicyBackend
```

Fehler beim Herstellen der Verbindung mit der Richtliniensammlung

Configuration Agent muss eine Verbindung zu einer gültigen Konfigurationsdatensammlung herstellen, um die Richtlinieninformationen herunterzuladen und im Cache zu speichern. Kann keine Verbindung hergestellt werden, werden beim Start von Desktop-Client-Anwendungen Fehler ähnlich der folgenden in den Configuration Agent-Protokollen aufgezeichnet. Im folgenden Fall existiert der Host "sobuild" nicht, es kann keine Verbindung zum Host hergestellt werden oder der Host kann nicht über Port 389 auf einen LDAP-Server zugreifen. Sie können dieses Problem beheben, indem Sie mithilfe von Agent Configuration Wizard (`/usr/bin/apoc-config`) überprüfen, ob die Richtliniensammlung korrekt angegeben ist, und - sollte dies der Fall sein - sicherstellen, dass der Zugriff auf die Richtliniensammlung möglich ist. Beispielsweise müssen Sie bei einer LDAP-Datensammlung sicherstellen, dass ein LDAP-Server ausgeführt wird, der Rechner, auf dem der LDAP-Server läuft, im Netzwerk verfügbar ist und dass der von Ihnen angegebene Port mit dem übereinstimmt, den der LDAP-Server verwendet.

Wenn Sie über eine SSL-Verbindung auf einen LDAP-Server zugreifen möchten, müssen Sie sicherstellen, dass im Schlüsselspeicher das Zertifikat für die Java-Laufzeitumgebung enthalten ist, auf die Configuration Agent zurückgreift. Lesen Sie den Abschnitt „Configuration Agent” auf Seite 16, um detaillierte Informationen zu `apoc-config` zu erhalten.

```
FINER: New client added
Nov 18, 2005 2:17:43 PM com.sun.apoc.daemon.misc.APOCLogger finer
FINER: CreateSession transaction started
Nov 18, 2005 2:17:43 PM com.sun.apoc.daemon.misc.APOCLogger finer
FINER: Creating new client session
Nov 18, 2005 2:17:43 PM com.sun.apoc.daemon.misc.APOCLogger finest
FINEST: Authenticating user geoffh
Nov 18, 2005 2:17:43 PM com.sun.apoc.daemon.misc.APOCLogger finest
FINEST: Authentication successful
Nov 18, 2005 2:17:43 PM PolicyBackend openPolicyBackend
FINER: THROW
com.sun.apoc.daemon.misc.APOCException: com.sun.apoc.daemon.misc.APOCException:
com.sun.apoc.spi.OpenConnectionException: An error occured while connecting to
ldap://sobuild:389.
    at com.sun.apoc.daemon.apocd.PolicyBackend.<init>(Unknown Source)
    at com.sun.apoc.daemon.apocd.HostPolicyBackend.<init>(Unknown Source)
    at com.sun.apoc.daemon.apocd.PolicyBackendFactory.openPolicyBackend(Unknown Source)
    at com.sun.apoc.daemon.apocd.Cache$DataSource.openPolicyBackend(Unknown Source)
    at com.sun.apoc.daemon.apocd.Cache$DataSource.open(Unknown Source)
    at com.sun.apoc.daemon.apocd.Cache.createDataSources(Unknown Source)
    at com.sun.apoc.daemon.apocd.Cache.<init>(Unknown Source)
    at com.sun.apoc.daemon.apocd.CacheFactory.createNewCache(Unknown Source)
    at com.sun.apoc.daemon.apocd.CacheFactory.openCache(Unknown Source)
    at com.sun.apoc.daemon.apocd.Session.<init>(Unknown Source)
    at com.sun.apoc.daemon.transaction.CreateSessionTransaction.executeTransaction
(Unknown Source)
    at com.sun.apoc.daemon.transaction.Transaction.execute(Unknown Source)
    at com.sun.apoc.daemon.apocd.ClientEventHandler.handleEvent(Unknown Source)
```

```
    at com.sun.apoc.daemon.apocd.EventWorkerThread.run(Unknown Source)
Caused by: com.sun.apoc.daemon.misc.APOCException:
com.sun.apoc.spi.OpenConnectionException: An error occured while
connecting to ldap://sobuild:389. at
com.sun.apoc.daemon.apocd.PolicyBackendFactory.openPolicyMgr(Unknown Source)
    ... 14 more
Caused by: com.sun.apoc.spi.OpenConnectionException: An error occured while
connecting to ldap://noSuchHost:389.
    at com.sun.apoc.spi.ldap.LdapClientContext.prepareConnection(Unknown Source)
    at com.sun.apoc.spi.ldap.LdapClientContext.connect(Unknown Source)
    at com.sun.apoc.spi.ldap.LdapConnectionHandler.openAuthorizedContext(Unknown Source)
    at com.sun.apoc.spi.ldap.LdapConnectionHandler.connect(Unknown Source)
    at com.sun.apoc.spi.ldap.entities.LdapOrganizationProvider.open(Unknown Source)
    at com.sun.apoc.spi.PolicyMgrFactoryImpl.createPolicyMgr(Unknown Source)
    ... 15 more
Caused by: netscape.ldap.LDAPException: failed to connect to server sobuild:389 (91);
Cannot connect to the LDAP server
    at netscape.ldap.LDAPConnSetupMgr.connectServer(LDAPConnSetupMgr.java:422)
    at netscape.ldap.LDAPConnSetupMgr.openSerial(LDAPConnSetupMgr.java:350)
    at netscape.ldap.LDAPConnSetupMgr.connect(LDAPConnSetupMgr.java:244)
    at netscape.ldap.LDAPConnSetupMgr.access$0(LDAPConnSetupMgr.java:241)
    at netscape.ldap.LDAPConnSetupMgr$1.run(LDAPConnSetupMgr.java:179)
    at java.lang.Thread.run(Thread.java:595)
Nov 18, 2005 2:17:44 PM PolicyBackend openPolicyBackend
```

Verbindungen mit nicht konfigurierten Richtliniensammlungen

Damit Configuration Agent die Richtliniendaten in einer Richtliniensammlung finden kann, muss die Richtliniensammlung ordnungsgemäß konfiguriert sein. Wenn Sie eine Richtliniensammlung angeben, die überhaupt nicht oder falsch konfiguriert ist, werden beim Start von Desktop-Client-Anwendungen Fehlermeldungen ähnlich der folgenden in den Configuration Agent-Protokollen aufgezeichnet. Informationen zur Behebung dieses Problems finden Sie im entsprechenden Abschnitt.

```
FINER: New client added
Nov 18, 2005 2:36:55 PM com.sun.apoc.daemon.misc.APOCLogger finer
FINER: CreateSession transaction started
Nov 18, 2005 2:36:55 PM com.sun.apoc.daemon.misc.APOCLogger finer
FINER: Creating new client session
Nov 18, 2005 2:36:55 PM com.sun.apoc.daemon.misc.APOCLogger finest
FINEST: Authenticating user geoffh
Nov 18, 2005 2:36:55 PM com.sun.apoc.daemon.misc.APOCLogger finest
FINEST: Authentication successful
Nov 18, 2005 2:36:55 PM PolicyBackend openPolicyBackend
FINER: THROW
com.sun.apoc.daemon.misc.APOCException: com.sun.apoc.daemon.misc.APOCException:
com.sun.apoc.spi.environment.RemoteEnvironmentException: Error on reading the
configuration data on LDAP server ldap://sobuild:389.
```

```
    at com.sun.apoc.daemon.apocd.PolicyBackend.<init>(Unknown Source)
    at com.sun.apoc.daemon.apocd.HostPolicyBackend.<init>(Unknown Source)
    at com.sun.apoc.daemon.apocd.PolicyBackendFactory.openPolicyBackend(Unknown Source)
    at com.sun.apoc.daemon.apocd.Cache$DataSource.openPolicyBackend(Unknown Source)
    at com.sun.apoc.daemon.apocd.Cache$DataSource.open(Unknown Source)
    at com.sun.apoc.daemon.apocd.Cache.createDataSources(Unknown Source)
    at com.sun.apoc.daemon.apocd.Cache.<init>(Unknown Source)
    at com.sun.apoc.daemon.apocd.CacheFactory.createNewCache(Unknown Source)
    at com.sun.apoc.daemon.apocd.CacheFactory.openCache(Unknown Source)
    at com.sun.apoc.daemon.apocd.Session.<init>(Unknown Source)
    at com.sun.apoc.daemon.transaction.CreateSessionTransaction.executeTransaction
(Unknown Source)
    at com.sun.apoc.daemon.transaction.Transaction.execute(Unknown Source)
    at com.sun.apoc.daemon.apocd.ClientEventHandler.handleEvent(Unknown Source)
    at com.sun.apoc.daemon.apocd.EventWorkerThread.run(Unknown Source)
```

Die Configuration Agent-Protokolle enthalten eine Meldung, die "Max. Client-Verbindungen" betreffen. Was bedeutet dies?

Jede Desktop-Client-Anwendung (gconfd, Mozilla, StarOffice), die von Configuration Agent aktiviert ist, stellt eine Verbindung zu Configuration Agent her, wenn sie ausgeführt wird. Die maximale Anzahl dieser Verbindungen wird in der Konfiguration von Configuration Agent festgelegt. Standardmäßig ist diese Zahl auf 50 Verbindungen beschränkt. Auf einem Rechner mit mehreren Benutzern müssen Sie diesen Wert möglicherweise erhöhen, indem Sie mit Configuration Agent Wizard (`/usr/bin/apoc-config`) die Einstellung "Max. Client-Verbindungen" ändern. Erreicht Configuration Agent die maximale Anzahl von Verbindungen, werden in den Configuration Agent-Protokollen Fehlermeldungen ähnlich der folgenden aufgezeichnet:

```
Nov 18, 2005 3:20:55 PM com.sun.apoc.daemon.misc.APOCLogger warning
WARNING: The maximum number of client connections ( 50 ) has been reached.
No new client connections can be established at this time.
```

Ich habe einige Richtlinien mithilfe von Desktop Manager geändert. Diese Änderungen werden auf den Client-Rechnern jedoch nicht angezeigt.

Bei der Entwicklung von Configuration Agent wurde unter anderem davon ausgegangen, dass mit Desktop Manager erstellte Richtliniendaten relativ statisch sind, das heißt, dass sie sich nicht häufig ändern. Hieraus ergibt sich ein Ansatz, der dazu führt, dass Configuration Agent periodisch die Richtliniensammlung auf durchgeführte Änderungen prüft. Standardmäßig prüft Configuration Agent die Datensammlung stündlich für alle Desktop-Anwendungen. Wenn Sie mit Desktop Manager eine Änderung durchführen, müssen Sie daher bis zu eine Stunde warten, bevor laufende Desktop-Anwendungen Kenntnis von der Änderung erhalten. Bei Bedarf können Sie den Wert von "Allgemeines Erkennungsintervall" mithilfe von Agent Configuration Wizard (`/usr/bin/apoc-config`) erhöhen, um die Datensammlung häufiger zu prüfen. Alternativ können Sie in Configuration Agent die Aktualisierung der Richtliniendaten für alle verbundenen

Anwendungen erzwingen, indem Sie sich als Superuser anmelden und den Befehl `/usr/lib/apoc/apocd change-detect` ausführen.

◆ ◆ ◆ KAPITEL 4

Java Web Console

Java Web Console ist darauf ausgerichtet, eine gemeinsame, webbasierte Systemverwaltungslösung für Sun Microsystems zu schaffen. Sie dient als der eine, zentrale Einstiegspunkt, von dem aus Benutzer auf Systemverwaltungsanwendungen mit einer einheitlichen Benutzeroberfläche zugreifen können.

Die Konsole basiert auf einem Webmodell. Dafür sprechen zahlreiche gute Gründe. Der Hauptgrund ist, dass Systemadministratoren ihre Verwaltungsanwendungen über einen Webbrowser erreichen können sollen.

Java Web Console bietet:

- Eine gemeinsame Authentifizierung und Autorisierung
- Eine gemeinsame Anmeldung
- Einen einzelnen Einstiegspunkt für den Zugriff auf sämtliche Verwaltungsanwendungen über denselben HTTPS-Port
- Ein einheitliches Aussehen

Diese Konsole bietet den Vorteil, dass Sie sich als Administrator nur einmal anmelden müssen, um beliebige über die Konsole zugängliche Anwendungen verwenden können.

Installation

Systemanforderungen

Java Web Console unterstützt mehrere Client- und Server-Betriebssysteme sowie verschiedene Browser.

Client

- Netscape™ 6.2x, und 7.x unter Solaris 10
- Netscape 6.2x und 7.x unter Windows 98, 98 SE, ME, 2000 und XP
- Internet Explorer 5.5x und 6.x unter Windows 98, 98 SE, ME, 2000 und XP
- Mozilla 1.4x unter Solaris
- Firefox 1.0 unter Solaris

Server

- Solaris 10
- Red Hat Application Server 2.1, 3.0
- SuSE Linux 8.0 oder höher
- J2SE Version 1.4.1_03 oder höher

  Wenn J2SE 1.4.1 oder eine niedrigere Version auf dem Server erkannt wird, fordert Sie das Einrichtungsprogramm dazu auf, die Installation mit der J2SE-Version der Java Desktop System Management Tools-CD zu aktualisieren.
- Tomcat: 4.0.3 oder höher

  Tomcat ist auf der Java Desktop System Management Tools-CD enthalten.

Installation von Java Web Console

Java Web Console 2.2.4 ist Teil des Betriebssystems Solaris 10. Desktop Manager erfordert jedoch Version 2.2.5. Eine Kopie von Version 2.2.5 wird im Archiv von Desktop Manager im Verzeichnis `server/console` bereitgestellt. Es kann installiert werden, indem der Befehl `./setup` in diesem Verzeichnis ausgeführt wird.

Wenn Sie Java Web Console 3.0 installiert haben, müssen Sie, wie oben erwähnt, die Version 3.0 deinstallieren und dann Java Web Console 2.2.5 vom Verzeichnis `server/console` installieren.

Ausführen der Konsole

Sie brauchen normalerweise nur den Server Java Web Console anzuhalten und neu zu starten, wenn Sie eine neue Anwendung registrieren möchten.

Achtung – Stellen Sie, bevor Sie Java Web Console zum ersten Mal starten, sicher, dass die Installation von Desktop Manager abgeschlossen ist. Java Web Console kann *nicht* erfolgreich ausgeführt werden, bevor Sie nicht mindestens eine Anwendung erfolgreich in der Konsole bereitgestellt haben.

- Geben Sie zum Starten von Java Web Console den Befehl `smcwebserver start` ein.
- Geben Sie zum Anhalten von Java Web Console den Befehl `smcwebserver stop` ein.

- Geben Sie zum erneuten Starten von Java Web Console den Befehl `smcwebserver restart` ein.
- Geben Sie für den Zugriff auf Java Web Console folgende URL in Ihren Browser ein: `https://`*<Host-Name>*`.`*<Domänenname>*`:6789`

Java Web Console unterstützt eine Unix-basierte Authentifizierung und eine rollenbasierte Zugriffskontrolle (RBAC). Es können jedoch auch andere Authentifizierungsmechanismen wie beispielsweise die LDAP-Authentifizierung konfiguriert werden.

Hinweis – Die Standardzeitüberschreitung für Sitzungen beträgt 15 Minuten. Der Zeitüberschreitungswert kann mit dem Befehl `smreg` modifiziert werden. Möchten Sie die Zeitüberschreitung beispielsweise auf 5 Minuten einstellen, geben Sie `smreg add -p -c session.timeout.value=5` ein.

Weitere Informationen zu den Befehlen für Java Web Console finden Sie in der Online-Dokumentation (Man Pages) zu smcwebserver und smreg.

Entfernen von Java Web Console

Achtung – Wenn Sie unter Solaris arbeiten, können Sie Java Web Console nicht entfernen, da es Teil des Betriebssystems ist.

Problembehebung für Java Web Console

Installation von Java Web Console nicht möglich

Symptom: Am Ende der Installation wird eine Meldung angezeigt, dass Java Web Console nicht gestartet werden kann, da keine registrierten Anwendungen vorhanden sind.

Mögliche Ursachen: Sobald das Modul Desktop Manager installiert ist, startet es Java Web Console.

Verbindung verweigert

Symptom: Sie versuchen, die korrekte URL zu öffnen, beispielsweise `https://`*<Ihr.Server>*`:6789`, aber die Verbindung wird verweigert.

Mögliche Ursachen: Java Web Console läuft nicht auf dem Server.

Anmeldung nicht möglich

Hinweis – Das LDAP-Anmeldemodul ist standardmäßig nicht installiert. Deshalb werden Anmeldeinformationen nicht mit den im LDAP-Server gespeicherten Informationen verglichen, und es sind normale Systemanmeldeinformationen erforderlich. Der Abschnitt zur Problembehebung ist nur dann anwendbar, wenn Sie das LDAP-Anmeldemodul manuell installiert haben.

Symptom: Sie erreichen die Anmeldeseite der Webkonsole, aber die Kombination aus Benutzernamen und Passwort wird zurückgewiesen.

Mögliche Ursachen:

- Der LDAP-Server läuft nicht.
- Das LDAP-Authentifizierungsmodul für die Webkonsole ist nicht korrekt konfiguriert.
- Der Benutzer ist auf dem LDAP-Server nicht vorhanden.
- Der Benutzer hat auf dem LDAP-Server ein anderes Passwort.

Keine Verknüpfung zu Desktop Manager

Symptom: Sie melden sich erfolgreich bei der Webkonsole an, aber die Seite mit der Anwendungsliste enthält Desktop Manager nicht.

Mögliche Ursachen:

- Das Modul Desktop Manager ist nicht installiert.

Null-Zeiger-Ausnahme, Tomcat/Java-Fehler oder leere Seite

Symptom: Sie können Desktop Manager öffnen, aber es wird nichts Sinnvolles angezeigt, nur eine leere Seite oder Fehlermeldungen.

Mögliche Ursachen: Wenn in der Fehlermeldung der Fehler `NoClassDefFoundError: sun/tools/javac/Main` erwähnt wird, verwendet Java Web Console die falsche Java-Installation.

Andere Probleme

Wenn der Webserver nicht ordnungsgemäß läuft, können Protokolldateien weiterhelfen. Sie befinden sich im Verzeichnis `/var/log/webconsole/`. Sie können die Detailstufe der Protokollinformationen erhöhen, indem Sie den folgenden `smreg`-Befehl verwenden:

```
smreg add -p debug.trace.level=3
smreg add -p debug.trace.options=tmp
```

Die ursprünglichen Einstellungen können mit folgendem Befehl wiederhergestellt werden:

```
smreg add -p debug.trace.level=0
smreg add -p debug.trace.options=m
```

Ein vollständiges Abbild der Konfigurationsdatenbank kann wie folgt abgerufen werden:

```
smreg list
```

Es ist möglich, dass der Webserver, auf dem Desktop Manager ausgeführt wird, nicht korrekt heruntergefahren wird, und die entsprechenden Ports in Verwendung bleiben. Dies führt dazu, dass der neue Webserver überhaupt nicht startet. Gibt der Befehl `smcwebserver start/restart` eine Fehlermeldung aus bzw. ist Desktop Manager immer noch erreichbar, selbst nach Ausführen des Befehls `smcwebserver stop`, oder verhält sich der neu gestartete Server immer noch wie die zuvor geöffnete Instanz, müssen Sie prüfen, ob Port 6789 noch in Verwendung ist (`netstat -a | grep 6789`) oder der Webserver noch läuft (`ps -ef | grep java`). Ist eines von beidem der Fall, muss der entsprechende Prozess beendet werden, sodass Port 6789 nicht mehr in Verwendung ist.

◆ ◆ ◆ ANHANG A

Konfigurationsparameter

Diese Parameter können für die folgenden Komponenten von Desktop Manager definiert werden:

- Desktop Manager: in den Dateien, die die Konfigurationsdatensammlungen (im Verzeichnis `/etc/opt/SUNWapcmg/`) definieren.
- Configuration Agent: in der Datei `/etc/apoc/policymgr.properties` .
- Desktop Manager-CLI: in der Datei `$HOME/pgtool.properties` , mit der Einschränkung, dass die CLI nur reine LDAP-Datensammlungen unterstützt.

Die Parameter können mit Präfixen versehen werden, die angeben, auf welchen Datensammlungsanbieter sie angewendet werden. Für jeden Anbieter wird der Parameter mit Präfix zuerst berücksichtigt. Wurde solch ein Parameter nicht definiert, wird der Parameter ohne Präfix verwendet.

TABELLE A–1 Präfixe

Präfixwert	Datensammlungsanbieter
ORGANIZATION_	Organisationsbaum
DOMAIN_	Domänenbaum
PROFILE_	Profile
ASSIGNMENT_	Zuweisungen
LDAP_META_CONF_	Zuordnung von Daten im Fall von LDAP-Datensammlungen

TABELLE A–2 Parameter

Name	Beschreibung	Mögliche Werte	Standardwert

PROVIDER_URL	URL, die die Verbindung zur Datensammlung angibt. Eine Liste von URLs kann dafür verwendet werden, Ersatzdatenbanken anzugeben, falls die Verbindung mit der ersten fehlschlägt.	Liste mit mindestens einer oder mehreren durch Leerzeichen getrennten URLs, die eine der folgenden Formen annimmt:
		`ldap://<Host>:<Port>/<baseDN>`
		`ldaps://<Host>:<Port>/<baseDN>`
		`file://<Dateipfad>`
		`http://<Host>:<Port>/<Dateipfad>`
		`https://<Host>:<Port>/<Dateipfad>`
			Keine Angabe, obligatorischer Parameter

SECURITY_PRINCIPAL	Benutzername für die Verbindung zur Datensammlung.	Benutzername eines Benutzers, der Zugriffsrechte zum Lesen und Suchen in der Datensammlung hat, oder keine Angabe (anonyme Verbindungen).	Keine Angabe, anonyme Verbindung
SECURITY_CREDENTIALS	Passwort für den in SECURITY_PRINCIPAL. definierten Benutzer	Verschlüsseltes oder Klartext-Passwort.	Kein
SECURITY_CREDENTIALS _ENCODING	Gibt an, ob das in SECURITY_PRINCIPAL definierte Passwort verschlüsselt ist. Warnung: Die Verschlüsselung des Passworts ist nur eine Maske für das Passwort. Sie ist in keiner Weise als sichere Verschlüsselung zu betrachten.	"Verschlüsseln", wenn das Passwort verschlüsselt ist (erfolgt automatisch durch Assistenten bei der Generierung der Konfigurationsdaten). "Kein", wenn das Passwort im Klartext angezeigt wird; verwenden Sie diesen Wert, wenn Sie das Passwort bearbeiten möchten.	"Kein"
MAX_SEARCH_RESULT	Maximum der von der Suche in den Datensammlungen ausgegebenen Ergebnisse. Hinweis: Das Präfixschema gilt nicht für diesen Parameter.	Positive Zahl, 0 bedeutet unbeschränkt.	100

Die folgenden Parameter gelten nur für LDAP-Datensammlungen.

TABELLE A–3 LDAP-spezifische Parameter

Name	Beschreibung	Mögliche Werte	Standardwert
AuthDn	Voll qualifizierter DN eines Benutzers, der für den erstmaligen Zugriff auf die LDAP-Datensammlung verwendet werden muss, um den in SECURITY_PRINCIPAL definierten Benutzer abzurufen.	Benutzername eines Benutzers der Zugriffsrechte für das Lesen und Suchen in der Datensammlung hat, oder kein Wert (anonyme Verbindungen).	Keine Angabe, anonymer Zugriff
Passwort	Passwort für AuthDN.	Verschlüsseltes oder Klartext-Passwort.	Kein
Password_ENCODING	Gibt an, ob das in "Passwort" definierte Passwort verschlüsselt ist. Warnung: Die Passwortverschlüsselung ist nur eine Maske für das Passwort. Sie ist in keiner Weise als sichere Verschlüsselung zu betrachten.	"Verschlüsseln", wenn das Passwort verschlüsselt ist (erfolgt automatisch durch Assistenten bei der Generierung der Konfigurationsdaten). "Kein", wenn das Passwort im Klartext angezeigt wird; verwenden Sie diesen Wert, wenn Sie das Passwort bearbeiten möchten.	"Kein"
Verbindungs-Zeitüberschreitung	Zeitüberschreitung bei der Herstellung der Verbindung in Sekunden.	Positive Zahl, 0 für unbegrenzte Zeit.	1

BEISPIEL A–1 Beispiel für ein kombiniertes Backend

Beispiel für ein kombiniertes Backend, bei dem die Informationen über die Hosts und Benutzer von einer vorhandenen LDAP-Datensammlung abgerufen werden, während die Profile und ihre Zuweisungen im Dateisystem gespeichert werden.

```
#Organization, Domain, MetaConf
PROVIDER_URL = ldap://server1.sun.com:389/o=apoc ldap://server2.sun.com:389/o=apoc
SECURITY_PRINCIPAL = jmonroe
SECURITY_CREDENTIALS = JmonroE
SECURITY_CREDENTIALS_ENCODING = none
AuthDn = cn=reader,ou=special users,o=apoc
Password = lakjflajf
Password_ENCODING = scramble
```

BEISPIEL A–1 Beispiel für ein kombiniertes Backend *(Fortsetzung)*

```
ConnectTimeout = 5

#Profile
PROFILE_PROVIDER_URL = file:///path/to/repository

#Assignment
ASSIGNMENT_PROVIDER_URL = file:///path/to/repository
```

◆ ◆ ◆ ANHANG B

Verwenden von OpenLDAP und Active Directory mit Desktop Manager

Verwenden eines OpenLDAP-Servers mit Desktop Manager

Wenn Sie einen OpenLDAP-Server als Datensammlung für die Desktop Manager-Daten einsetzen möchten, muss das Schema des Servers auf die Objektklassen und Attribute ausgeweitet werden, die zum Speichern von Konfigurationsdaten verwendet werden. Eine benutzerdefinierte Schemadatei mit dem Namen `apoc.schema` befindet sich im Verzeichnis `/usr/share/webconsole/apoc/deploy`.

Diese Datei muss in das Unterverzeichnis `schema` des OpenLDAP-Konfigurationsverzeichnisses (`/etc/openldap`) kopiert und in das OpenLDAP-Schema importiert werden. Hierzu fügen Sie an das Ende der Schema-Include-Sequenz in der Datei `slapd.conf`, die sich in demselben Verzeichnis befindet, die Zeile `include /etc/openldap/schema/apoc.schema` ein. Weitere Informationen zur Erweiterung des Schemas eines OpenLDAP-Servers entnehmen Sie bitte der Dokumentation des jeweiligen Servers.

Nach Erweiterung des Schemas des OpenLDAP-Servers kann die Konfiguration mithilfe des Assistenten für das Hinzufügen von Datensammlungen in Desktop Manager abgeschlossen werden.

Hinweis – Desktop Manager Agent versucht, eine anonyme Verbindung zum OpenLDAP-Server herzustellen und gibt dazu zwar den DN des Benutzers, für den Daten angefordert werden, aber kein Passwort an. Eine derartige anonyme Authentifizierung kann bei einigen Versionen von OpenLDAP-Servern unter Umständen standardmäßig deaktiviert sein. In diesem Fall muss die Zeile `allow bind_anon_cred` in die gemeinsamen Serverparameter eingefügt werden, die in der Datei `slapd.conf` im OpenLDAP-Konfigurationsverzeichnis (`/etc/openldap`) definiert sind. Weitere Informationen zu diesem Parameter entnehmen Sie bitte der Dokumentation des jeweiligen Servers.

Verwenden eines Active Directory-Servers mit Desktop Manager

Wenn Sie einen Active Directory-Server als Datensammlung für die Desktop Manager-Daten einsetzen möchten, muss das Schema des Servers auf die Objektklassen und Attribute ausgeweitet werden, die zum Speichern von Konfigurationsdaten verwendet werden. Eine Schema-Erweiterungsdatei mit dem Namen `apoc-ad.ldf` befindet sich im Verzeichnis `/usr/share/webconsole/apoc/deploy`.

Die Datei `apoc-ad.ldf` muss wie folgt in das Active Directory-Schema importiert werden:

1. Aktivieren Sie die Schema-Erweiterungen. Näheres hierzu entnehmen Sie bitte der Active Directory-Dokumentation.
2. Geben Sie Folgendes in die Befehlszeile ein: **`ldifde -i -c "DC=Sun,DC=COM" <`***`Basis-DN`***`> -f apoc-ad-registry.ldf`** .

Hinweis – Ersetzen Sie dabei <*Basis-DN*> durch den Basis-DN für Active Directory.

Nach Erweiterung des Active Directory-Server-Schemas kann die Konfiguration mithilfe des Assistenten zum Hinzufügen einer Konfigurationsdatensammlung in Desktop Manager abgeschlossen werden.

Bei Aufforderung zur Eingabe der LDAP-Anmeldeinformationen im Assistenten zum Hinzufügen von Konfigurationsdatensammlungen müssen Sie einen vollständigen DN und ein Passwort eines Benutzers eingeben, der Lesezugriff auf den Baum hat. Dabei kann es sich um einen Benutzer ohne weitere Berechtigungen für Active Directory handeln. Genaueres zur Einrichtung eines solchen Benutzers entnehmen Sie bitte der Active Directory-Dokumentation. Außerdem müssen Sie dem System, auf dem Desktop Manager ausgeführt wird, den Domänennamen von Active Directory mitteilen. Hierzu können Sie in die Datei `/etc/hosts` dieses Systems eine Zeile mit der Zuordnung zwischen der IP-Adresse des Active Directory-Servers und dessen Domänennamen einfügen.

Zum Abrufen der Konfigurationsdaten von einem Desktop-Host muss der Domänenname von Active Directory auch diesem Host mitgeteilt werden. Die Authentifizierung von Desktop-Benutzern kann entweder anonym oder per GSSAPI erfolgen.

- Für die Authentifizierung über anonyme Verbindungen muss der Active Directory-Server so konfiguriert sein, dass alle Benutzer leseberechtigt sind. Näheres hierzu entnehmen Sie bitte der Active Directory-Dokumentation.
- Für die Authentifizierung über GSSAPI muss sich der Benutzer bei Active Directory authentifiziert haben, und die Benutzer-Anmeldedaten müssen auf dem System verfügbar sein. Dies kann durch Konfiguration der Kerberos-Authentifizierung auf Ihrem System erreicht werden. Dadurch werden diese Anmeldedaten beim Anmelden generiert. Weitere Informationen zur Vorgehensweise finden Sie im Administrationshandbuch Ihres Systems.

◆ ◆ ◆ ANHANG C

Organisatorische Zuordnung

Organisatorische Zuordnung

Zur Definition der Zuordnung zwischen LDAP-Einträgen und Desktop Manager-Elementen muss die Datei `Organization` bearbeitet werden. Dabei sind für die verschiedenen Schlüssel Werte anzugeben, die der Struktur der LDAP-Datensammlung entsprechen.

Benutzerelemente sind durch eine für alle Elemente geltende Objektklasse sowie ein Attribut gekennzeichnet, dessen Wert im Bereich der gesamten Datensammlung einmalig sein muss. Sie können ein Namensanzeigeformat liefern, das sich auf die Anzeige der Benutzernamen in der Verwaltungsanwendung auswirkt, und haben die Möglichkeit, einen Behältereintrag zu definieren, für den Fall, dass für die Benutzereinträge in der Organisation solche Einträge verwendet werden. Sehen Sie hier die Schlüsselnamen und ihre Standardwerte:

```
# Objektklasse für alle Benutzereinträge
User/ObjectClass=inetorgperson
# Attribut, dessen Wert in Benutzereinträgen im Bereich der Datensammlung
einmalig sein muss
User/UniqueIdAttribute=uid
# Optionaler Behälter in Organisationseinträgen der Benutzereinträge;
# entfernen Sie diese Zeile, sofern sie nicht erforderlich ist
User/Container=ou=People
# Namensanzeigeformat innerhalb der Verwaltungsanwendung
User/DisplayNameFormat=sn, givenname
```

Rollenelemente sind durch eine Liste möglicher Objektklassen und die entsprechenden Namensattribute gekennzeichnet. Diese Listen haben das Format `<Objekt1>,<Objekt2>,...,<ObjektN>` und müssen bündig gemacht werden. Das heißt, dass die Listen dieselbe Elementanzahl aufweisen müssen und die n-te Objektklasse mit dem n-ten Namensattribut verbunden sein muss. Zwei Schlüssel bestimmen sowohl das Verhältnis zwischen Rollen und Benutzern als auch zwischen Rollen und Rechnern. Mit dem Schlüssel *VirtualMemberAttribute* ist ein Attribut anzugeben, dessen Werte von einem Benutzer- oder Rechnereintrag abgefragt werden können. Außerdem muss der Schlüssel die vollständigen DNs der Rollen enthalten, zu welchen der Eintrag gehört. Mit dem Schlüssel *MemberAttribute* ist ein Attribut

aus einem Benutzer- oder Rechnereintrag für den Suchfilter anzugeben. Außerdem muss der Schlüssel die vollständigen DNs der Rollen enthalten, zu welchen der Benutzer oder Rechner gehört. Während der Schlüssel *VirtualMemberAttribute* ein virtuelles Attribut vom Typ Class Of Service sein kann, muss für den Schlüssel *MemberAttribut* ein tatsächliches, in einem Filter verwendbares Attribut angegeben werden. Sehen Sie hier die Schlüsselnamen und ihre Standardwerte:

```
# Liste von Objektklassen für Rollen
Role/ObjectClass=nsRoleDefinition
# Sich deckende Liste mit entsprechenden Namensattributen
Role/NamingAttribute=cn
# Tatsächliches Attribut (in einem Filter verwendbar), das die DNs der
# Rollen eines Benutzers/Rechners enthält
Role/MemberAttribute=nsRoleDN
# Attribut, durch dessen Abfrage für einen Benutzer oder Rechner die DNs
# der zugehörigen Rollen geliefert werden
Role/VirtualMemberAttribute=nsRole
```

Organisationselemente werden ähnlich wie Rollen durch zwei bündige Listen von Objektklassen und den dazugehörigen Namensattributen definiert. Sehen Sie hier die Schlüsselnamen und ihre Standardwerte:

```
# Liste von Objektklassen für Organisationen
Organization/ObjectClass=organization
# Sich deckende Liste mit entsprechenden Namensattributen
Organization/NamingAttribute=o
```

Domänenelemente werden ähnlich wie Organisationselemente definiert. Sehen Sie hier die Schlüsselnamen und ihre Standardwerte:

```
# Liste von Objektklassen für Domänen
Domain/ObjectClass=ipNetwork
# Sich deckende Liste mit entsprechenden Namensattributen
Domain/NamingAttribute=cn
```

Rechnerelemente werden ähnlich wie Benutzerelemente definiert. Sehen Sie hier die Schlüsselnamen und ihre Standardwerte:

```
# Objektklasse für alle Rechnereinträge
Host/ObjectClass=ipHost
# Attribut, dessen Wert in Rechnereinträgen im Bereich der Datensammlung
 einmalig sein muss
Host/UniqueIdAttribute=cn
# Optionaler Behälter in Domäneneinträgen der Rechnereinträge;
# entfernen Sie diese Zeile, sofern sie nicht erforderlich ist.
Host/Container=ou=Hosts
```