IBM Cúram Social Program Management
Version 6.0.5

Cúram Deployment Guide for WebSphere Application Server

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen in „Bemerkungen" auf Seite 47 gelesen werden.

Überarbeitung: März 2014

Diese Ausgabe bezieht sich auf IBM Cúram Social Program Management v6.0.5 und alle nachfolgenden Releases, sofern nicht anderweitig in neuen Ausgaben angegeben.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellen

Auf IBM WebSphere Application Server implementieren

Um ein Build der IBM Cúram Social Program Management-Anwendung für die Bereitstellung auf IBM WebSphere Application Server zu erstellen, sind Web-Client-Server- und Anwendungs-EAR-Dateien erforderlich. Darüber hinaus sind eine Reihe von Konfigurationseinstellungen notwendig.

Einführung

Implementierungshandbuch

In diesem Handbuch werden die Schritte beschrieben, die notwendig sind, um eine IBM® Cúram Social Program Management-Anwendung zur Implementierung auf der Basisversion von IBM WebSphere Application Server aufzubauen[1]. Außerdem sind in diesem Handbuch die Details zu der Unterstützung aufgeführt, die für das Konfigurieren und Implementieren auf WebSphere Application Server bereitgestellt wird, sowie, wo nötig, die erforderlichen manuellen Schritte.

Es wird vorausgesetzt, dass der Leser Kenntnisse der IBM Cúram Social Program Management-Entwicklungsumgebung besitzt. Er sollte also wissen, wie man eine Serveranwendung und einen Web-Client entwickelt und aufbaut. Außerdem wird vorausgesetzt, dass WebSphere Application Server bereits installiert ist. Details zu dieser Installation finden Sie im Dokument *Cúram Third Party Tools Installation Guide* (Cúram-Handbuch für die Installation von Tools von Fremdanbietern)[2].

EAR-Dateien erstellen

Einführung

Der wichtigste Schritt vor der Implementierung einer IBM Cúram Social Program Management-Anwendung ist ihre Paketierung in EAR(Enterprise Archive)-Dateien. Die Serveranwendung, bestehend aus Web-Client und Server, und die Web-Service-Anwendung werden in separate EAR-Dateien paketiert, wobei die Serverentwicklungsumgebung (Server Development Environment, SDEJ) Build-Ziele bereitstellt, die diese Task ausführen.

Vor der Ausführung der im folgenden Abschnitt beschriebenen Ziele stellen Sie sicher, dass die folgende Umgebungsvariable gesetzt ist:

- WAS_HOME

  Sie verweist auf das Basisverzeichnis der Basisinstallation von WebSphere Application Server, z.B. `d:\WebSphere\AppServer` oder `/opt/WebSphere/AppServer`.

1. Informationen zur Verwendung der Anwendung mit der Network Deployment-Edition von WebSphere Application Server finden Sie unter „WebSphere Application Server manuell konfigurieren” auf Seite 18.
2. Ziehen Sie das Installationshandbuch, das für Ihre Plattform relevant ist, zu Rate, also Microsoft Windows oder **UNIX**.

Die Unternehmensanwendung

Anwendungs-EAR-Datei erstellen

Folgendes Ziel muss vom Stammverzeichnis des Serverprojektes ausgeführt werden, um die Anwendungs-EAR-Datei für WebSphere Application Server zu erstellen:

```
build websphereEAR
```

Dieses Ziel erstellt eine fertig zu installierende EAR-Datei, `<SERVER_MODEL_NAME>.ear`, die im Verzeichnis `<SERVER_DIR>/build/ear/WAS` abgelegt wird[3].

Vor der Ausführung dieses Zieles muss eine vollständig aufgebaute Anwendung verfügbar sein. Details zum Aufbau einer IBM Cúram Social Program Management-Anwendung finden Sie im Dokument *Cúram Server Developer's Guide* (Entwicklerhandbuch zu Cúram Server).

Anmerkung: Eine EAR-Datei kann nicht für eine H2-Datenbank erstellt werden.[4]

Innere Abläufe

Das Ziel **websphereEAR** sammelt eine Anzahl zuvor generierter Java-Dateien und Implementierungsdeskriptoren und paketiert sie in eine EAR-Datei.

Die Java-Dateien und Implementierungsdeskriptoren werden während des Aufbauprozesses anhand des Vorhandenseins von Geschäftsprozessobjekt(*BPO*)-Klassen, d.h. den Methoden von *Fassaden*- oder WebService-Klassen, erstellt, und können von fernen Clients aufgerufen werden.

Standardmäßig werden alle Fernaufrufe an den Server von der Session-Bean `curam.util.invoke.EJBMethod` bearbeitet, nicht von einer Session-Bean über eine öffentlich zugängliche Schnittstelle. Diese Bean bietet Unterstützung für IBM Cúram Social Program Management-Funktionen wie Autorisierung, Prüfung oder Traceerstellung. Bei Bedarf ist es auch möglich, eine Fassadenschnittstelle zu erstellen[5].

Inhalt der Anwendungs-EAR-Datei

Die erstellte EAR-Datei weist folgende Struktur und folgenden Inhalt auf:

- Verzeichnis META-INF
  - `application.xml`

    Diese Datei wird automatisch generiert und führt die Zuordnung der EJB-Module zu den JAR-Dateien auf, die in der Anwendung enthalten sind.
  - `ibm-application-bnd.xmi`

    Eine generierte Erweiterungsdatei, die für WebSphere Application Server bestimmt ist.
  - `ibm-application-ext.xmi`

    Eine generierte Erweiterungsdatei, die für WebSphere Application Server bestimmt ist.
  - `was.policy`

3. SERVER_MODEL_NAME und SERVER_DIR sind Umgebungsvariablen, die den Namen des Modells jeweils im Projekt und im Stammverzeichnis des Projekts angeben.

4. Weitere Informationen zur H2-Datenbank finden Sie im Dokument *Cúram Third-Party Tools Installation Guide for Windows* (Cúram-Installationshandbuch für Tools von anderen Anbietern für Windows).

5. Der optionale Build-Parameter `-Denablefacade=true` aktiviert die Erstellung eines Fassadencodes.

Eine WebSphere Application Server-Sicherheitsrichtliniendatei, die der Anwendung die Java-Berechtigung `java.security.AllPermission` ermöglicht.

– `MANIFEST.MF`

Die Manifestdatei, in der die Inhalte der EAR-Datei im Detail enthalten sind.

- **Kern-JAR-Dateien**

  Die Kern-JAR-Dateien beinhalten[6]:

  – `antlr.jar`
  – `appinf.jar`
  – `appinf_internal.jar`
  – `coreinf.jar`
  – `rules.jar`
  – `jde_commons.jar`
  – `log4j.jar`
  – `commons-pool.jar`
  – `commons-codec.jar`
  – `commons-discovery.jar`
  – `jdom.jar`
  – `axis.jar`
  – `castor.jar`
  – `jaxrpc.jar`
  – `saaj.jar`
  – `java_cup.zip`
  – `InfrastructureModule.jar`
  – `InvalidationModule.jar`
  – `DBtoJMS.war`
  – `ClientModule.war`

- **Fassaden-JAR-Dateien**

  Nur vorhanden, wenn die Fassadengenerierung aktiviert ist. Alle in der Anwendung definierten Fassaden sind in einer einzigen JAR-Datei paketiert, `FacadeModule.jar`. Diese JAR-Datei enthält die Bean-Implementierungsklassen für die EJB-Module, die die Fassaden darstellen. Die JAR-Datei enthält im Verzeichnis `META-INF` die folgenden Dateien:

  – `ejb-jar.xml`

  Diese Datei wird automatisch generiert und enthält die Definition jedes EJB-Moduls, das in der JAR-Datei enthalten ist. Alle öffentlich zugänglichen Methoden sowie die Details zu den für die EJB-Module verfügbaren Ressourcen sind hier aufgeführt.

  – `ibm-ejb-jar-bnd.xmi`

  Eine generierte Erweiterungsdatei, die für WebSphere Application Server bestimmt ist.

  – `ibm-ejb-jar-ext.xmi`

  Eine generierte Erweiterungsdatei, die für WebSphere Application Server bestimmt ist.

  – `Manifest.mf`

  Die Manifestdatei mit Zusatzinformationen zum Klassenpfad für die EJB.

6. Die Versionsnummern sind für die JAR-Dateien nicht im Detail aufgeführt.

- **Andere JAR-Dateien**

  Die restlichen JAR-Dateien enthalten den generierten und von Hand erstellten Code aus der Anwendung. Dazu gehören die Dateien `application.jar`, `codetable.jar`, `events.jar`, `struct.jar`, `messages.jar`, `implementation.jar` und `properties.jar`. Die Datei `properties.jar` enthält die Datei `Bootstrap.properties`. Dies ist die Datei mit den maschinenspezifischen Konfigurationseigenschaften für den ersten Verbindungsaufbau zur Datenbank.

Die Web-Service-Anwendung

Für die automatische Generierung von Web-Services, die durch Web Service Definition Language (WSDL) definiert sind, steht ein Stützelement zur Verfügung. Anwendungsentwickler können so das Potenzial des IBM Cúram Social Program Management-Modells mit der Zugänglichkeit der Web-Services kombinieren und auf diese Weise im Sinne des Wortes wiederverwendbare Komponenten erstellen.

Web-Services-EAR-Datei erstellen

Folgendes Ziel muss vom Stammverzeichnis des Serverprojektes ausgeführt werden, um die EAR-Datei für Web-Services zu erstellen:

```
build websphereWebServices
```

Optionale Überschreibungen sind:

- `prp.webipaddress`, die IP-Adresse, für die der Server, auf dem sich die Web-Services befinden, empfangsbereit ist. Die Standardadresse ist `http://localhost:2809`.
- `prp.contextproviderurl`, die URL des JNDI-Kontext-Providers. Dies ist die Adresse des Servers, auf dem sich die IBM Cúram Social Program Management-Komponenten befinden, die durch die Web-Services zugänglich gemacht werden. Die Standardadresse ist `iiop://localhost:2809`.
- `prp.contextfactoryname`, der JNDI-Kontext-Factoryname. Der Standardwert hierfür, der selten geändert werden muss, ist `com.ibm.websphere.naming.WsnInitialContextFactory`.

Dieses Ziel erstellt eine fertig zu installierende EAR-Datei, `<SERVER_MODEL_NAME>WebServices.ear`, die im Verzeichnis `<SERVER_DIR>/build/ear/WAS` abgelegt wird.

Vor Ausführen dieses Zieles muss eine vollständig aufgebaute IBM Cúram Social Program Management-Anwendung verfügbar sein, die für die Implementierung bereitsteht.

Innere Abläufe

Das Ziel **websphereWebServices** sammelt eine Anzahl zuvor generierter Java-Dateien und Implementierungsdeskriptoren und paketiert sie in eine EAR-Datei.

Die Java-Dateien und Implementierungsdeskriptoren werden während des Aufbauprozesses (siehe das Entwicklerhandbuch *Cúram Server Developer's Guide*) anhand der *Web-Service-Komponenenten* erstellt, die im Modell definiert worden sind. Geschäftsprozessobjekt(BPO)-Klassen sollten Serverkomponenten mit einem Web-Service-Stereotyp für diese Erstellung zugeordnet werden[7]. Jede Serverkomponente mit einem Web-Service-Stereotyp wird behandelt, als hätte sie auch ein Enterprise

7. Im Dokument *Cúram Server Modelling Guide* finden Sie Details zum Zuweisen von BPOs an die Serverkomponenten.

JavaBeans-Stereotyp. Das hängt damit zusammen, dass es sich bei Web-Service-Schnittstellen um Wrapper für öffentlich zugängliche BPOs handelt.

Inhalt der Web-Services-EAR-Datei

Die erstellte EAR-Datei weist folgende Struktur und folgenden Inhalt auf:

- Verzeichnis META-INF
  - `application.xml`

    Diese Datei enthält das Kernmodul für die Web-Service-Anwendung, die Datei `webservices.war`.
  - `ibm-application-bnd.xmi`

    Eine generierte Erweiterungsdatei, die für WAS (WebSphere Application Server) bestimmt ist.
  - `ibm-application-ext.xmi`

    Eine generierte Erweiterungsdatei, die für WAS (WebSphere Application Server) bestimmt ist.
  - `was.policy`

    Eine WAS-Sicherheitsrichtliniendatei, die der Anwendung die Java-Berechtigung `java.security.AllPermission` ermöglicht.
  - `MANIFEST.MF`

    Die Manifestdatei, in der die Inhalte der EAR-Datei im Detail enthalten sind.
- **Web-Service-WAR-Datei**

  Diese Datei enthält im Verzeichnis `WEB-INF/lib` unterstützende JAR-Dateien, darunter:
  - `coreinf.jar`

    Diese JAR-Datei enthält die Konvertierungsverfahren, die für die Unterstützung der Serialisierung der komplexen Typen, die in der Schnittstelle verwendet werden, notwendig sind.
  - `axis.jar`

    Diese JAR-Datei enthält die Axis-Web-Service-Engine.
  - `appwebservices.jar`

    Diese JAR-Datei enthält die Wrapperklassen, die es den Axis-Web-Services ermöglichen, eine Verbindung zu der/den Session-Bean(s) der IBM Cúram Social Program Management-Serveranwendung herzustellen, sowie die Klassen für die komplexen Typen, die in der Schnittstelle zu den Web-Services verwendet werden.
  - `server-config.wsdd`

    Diese `wsdd`-Datei ist im Verzeichnis `WEB-INF` abgelegt und enthält die Web-Service-Enginekonfiguration, die die IBM Cúram Social Program Management-GPOs den Web-Services zuordnet.

Web-Service-WSDL (Description Language)

Nach seiner Implementierung exponiert ein IBM Cúram Social Program Management-Web-Service seine eigene WSDL.

Gibt es beispielsweise einen Service unter der URL

`http://localhost:9082/CuramWS/services/MyTestService`,

so findet sich die WSDL-Beschreibung unter der URL

`http://localhost:9082/CuramWS/services/MyTestService?wsdl`.

Die URL

```
http://localhost:9082/CuramWS/services
```

gibt eine Webseite zurück, in der alle implementierten Web-Services sowie ein Link zu ihren WSDL-Dateien aufgeführt sind.

Das allgemeine URL-Format für die obigen Speicherpositionen lautet

```
http://<web-server>:<port-number>/<ServerModelName>WS/services/<BPO-name>.
```

Mehrfache EAR-Dateien

Das Erstellen einer Anwendungs-EAR-Datei erfordert auch eine optionale Datei, die es ermöglicht, die Clientkomponenten in unterschiedliche WAR- und EAR-Dateien zu splitten, und außerdem eine bessere Kontrolle über einige der EAR-Konfigurations- und eingeschlossenen Module gestattet. Diese Datei hat den Namen `deployment_packaging.xml` und muss in Ihrem `SERVER_DIR/project/config`-Verzeichnis abgelegt werden.

Die Datei `deployment_packaging.xml` hat das folgende Format:

```
<deployment-config>
  <ear name="Curam"
      requireServer="true">
    <components>custom,sample,SamplePublicAccess,core</components>
    <context-root>/Curam</context-root>
  </ear>
  <ear name="CuramExternal">
    <components>SamplePublicAccessExternal</components>
    <context-root>/CuramExternal</context-root>
    <custom-web-xml>${client.dir}/custom_web_xml</custom-web-xml>
  </ear>
</deployment-config>
```

Abbildung 1. deployment_packaging.xml sample

Jede Datei kann mehrfach vorhandene `ear`-Elemente aufweisen, was dazu führt, dass im Verzeichnis `SERVER_DIR/build/ear/WAS` eine EAR-Datei erstellt wird. Für jedes Element gibt es folgende Optionen:

- `name`

  Diese Option steuert die Benennung der EAR-Datei, die während des Vorgangs erstellt wird.

- `requireServer`

  Dieses optionale Attribut bestimmt, ob das Servermodul in die EAR-Datei eingeschlossen wird. Gültige Einträge sind `true` oder `false` (wahr/falsch). Der Standardwert ist `false`. Bei der Implementierung mehrfacher EAR-Dateien auf demselben Anwendungsserver darf dieses Attribut nur für eine der EAR-Dateien auf `true` gesetzt sein, da pro Cluster nur ein IBM Cúram Social Program Management-Servermodul implementiert werden darf. Ist `requireServer` für mehrere EAR-Dateien auf `true` gesetzt, müssen die übrigen EAR-Dateien in einem anderen Cluster implementiert werden, um Konflikte zu vermeiden.

- `components`

  Diese Option steuert, welche der Clientkomponenten in der EAR-Datei abgelegt werden. Außerdem steuert sie die Komponentenfolge für die Neuerstellung des Clients, die noch erfolgen muss. Das Kernverzeichnis stellt gewöhnlich keinen Teil der Komponentenfolge dar, aber in diesem Fall ist seine Aufnahme wichtig

zum Kennzeichnen, ob es in einer bestimmten WAR-Datei eingeschlossen werden muss. Einträge sollten hier in der typischen Komponentenfolge erfolgen, wie sie im Dokument *Cúram Server Developer's Guide* (Entwicklerhandbuch für Cúram Server) festgelegt ist, und durch Kommas getrennt sein.

- `context-root`

  Diese Option bildet den Stammkontext des WAR-Moduls im Implementierungsdeskriptor `application.xml`. Einträge, die hier gemacht werden, sollten mit einem Schrägstrich beginnen.

- `custom-web-xml`

  Dieses optionale Element bestimmt, ob eine angepasste `web.xml`-Datei die Standardversion in der WAR-Datei überschreiben soll. Einträge sollten hier ein Apache Ant-Pfad zu dem Verzeichnis sein, das die Datei `web.xml` enthält.

  Es besteht die Möglichkeit, als Teil dieses Pfads Verweise auf Umgebungsvariablen zu verwenden. So kann zum Beispiel `${client.dir}` zum Verweisen auf das Web-Client-Verzeichnis oder `${SERVER_DIR}` zum Verweisen auf das Serververzeichnis verwendet werden.

- `requireSearchServer`

  Weitere Informationen hierzu finden Sie unter *Cúram-Server für generische Suche*.

Für jeden Web-Client, also jede WAR-Datei, ist eine separate Clientkomponente erforderlich, in der die Anpassungen enthalten sind. Im Falle mehrfacher Web-Clients schließt Ihre Umgebungsvariable CLIENT_COMPONENT_ORDER alle Ihre angepassten Komponenten mit ein; es sind jedoch separate <ear>-Elemente erforderlich, für jede angepasste Webkomponente (und bei Bedarf auch für andere Komponenten) jeweils eines.

Wie für das Standardziel, so gilt auch hier, dass eine vollständig aufgebaute IBM Cúram Social Program Management-Anwendung verfügbar sein muss. Details zum Aufbau einer Anwendung finden Sie im Dokument *Cúram Server Developer's Guide* (Entwicklerhandbuch zu Cúram Server).

Alternative Ziele

Das Ziel **`websphereEAR`** erstellt eine `.ear`-Datei für die Cúram Social Program Management-Anwendung, die sowohl den Web-Client als auch die Anwendung enthält. Es wird ein Stützelement zur Verfügung gestellt, um eine Anwendungs-`.ear`-Datei zu erstellen, die nur die Webanwendung oder nur die Serveranwendung enthält.

Diese Ziele können notwendig sein, wenn die Web-Client- und die Serveranwendung auf unterschiedlichen Servern installiert werden müssen. Um beispielsweise sicheren Zugriff für externe Benutzer auf die Cúram-Anwendung zu unterstützen, kann eine neue Web-Client-Anwendung entwickelt werden. Diese Webanwendung kann einzeln implementiert werden und eine vorhandene Serveranwendung nutzen.[8]

Für die Erstellung einer `ear`-Datei, die nur die Web-Client-Anwendung enthält, ist der folgende Befehl zu verwenden:

```
build websphereEAR -Dclient.only=true
```

8. Weitere Informationen zur Sicherheit für externen Zugriff finden Sie im Entwicklerhandbuch *Cúram Server Developers Guide*.

Für die Erstellung einer ear-Datei, die nur die Serveranwendung enthält, ist der folgende Befehl zu verwenden:

```
build websphereEAR -Dserver.only=true
```

Anwendungsserver konfigurieren

Einführung

In diesem Kapitel wird davon ausgegangen, dass WebSphere bereits installiert ist. Für Details zur Installation ziehen Sie das Dokument *Cúram Third Party Tools Installation Guide*(Cúram-Handbuch für die Installation von Tools von Fremdanbietern)[9]zu Rate.

Die Konfiguration von WebSphere Application Server ist für alle Plattformen ähnlich und die Serverentwicklungsumgebung für Java (SDEJ) stellt als Unterstützung für die Konfiguration und Verwaltung der Installation eine Anzahl von Ant-Zielen zur Verfügung. Bei Interesse sind unter „WebSphere Application Server manuell konfigurieren" auf Seite 18 Einzelheiten zu den manuellen Schritten aufgeführt, die von den Konfigurationsscripts durchgeführt werden.

Bei dem vom SDEJ bereitgestellten Konfigurationsziel handelt es sich um eine einfache Standardkonfiguration, die für eine Produktionsumgebung möglicherweise unpassend ist.

Anmerkung: Das Ziel **configure** überschreibt das von WebSphere Application Server erstellte *Standard*profil, sofern nicht '-Dkeep.profile=true' an das Ziel übermittelt wird.

Konfiguration von WebSphere Application Server

Zur Konfiguration von WebSphere Application Server gehört das Einrichten eines Profils, einer Datenquelle und einer Anzahl von Servern sowie das Konfigurieren der JMS- und Sicherheitseinstellungen. Alle diese Aufgaben können durch Ausführen des Ziels **configure** durchgeführt werden, welches vom SDEJ zur Verfügung gestellt wird.

Sofern es beim Aufrufen des Ziels nicht gezielt überschrieben wird, nimmt das vom Ziel **configure** erstellte Profil die folgenden Standardwerte an:

- `profile.name=AppSvr01`
- `cell.name=${node.name}Cell`

Der Befehl **build configure** sollte vom Verzeichnis `<SERVER_DIR>` aus ausgeführt werden, um die automatische Konfiguration aufzurufen. Für dieses Ziel ist es erforderlich, dass die Dateien `AppServer.properties` und `Bootstrap.properties` im Verzeichnis `<SERVER_DIR>/project/properties`[10]vorhanden sind. Im Dokument *Cúram Server Developer's Guide* (Cúram Server-Entwicklerhandbuch) finden Sie weitere Informationen zur Einrichtung einer `Bootstrap.properties`-Datei. Unter „Konfiguration von WebSphere Application Server" finden Sie Beispielinhalte für die Datei `AppServer.properties`.

9. Verwenden Sie das Installationshandbuch, das für Ihre Plattform relevant ist, also Windows oder **UNIX**.

10. Es ist möglich, diese Standardposition für die Eigenschaftendatei zu überschreiben, indem man bei der Ausführung des Ziels **configure** -Dprop.file.location=<new location> angibt.

```
## EIGENSCHAFTEN DES ANWENDUNGSSERVERS

# Property to indicate WebSphere is installed.
as.vendor=IBM

# The username and encrypted password for admin server.
security.username=<z.B. websphere>
security.password=<encrypted password>

# Name des WebSphere-Knotens
node.name=MyNode

# Name des Servers, auf dem sich die Anwendung befindet.
curam.server.name=CuramServer
curam.server.port=2809

################################################
## DIE FOLGENDEN EIGENSCHAFTEN GELTEN NUR FÜR WebSphere ##
################################################
# The alias that should be used for the database authorization
curam.db.auth.alias=databaseAlias

# HTTP Port for the server on which the client
# will be accessed
curam.client.httpport=9044

# HTTP Port for the server on which the Web services
# will be accessed
curam.webservices.httpport=9082

# Property to set JVM initial and maximum heap size.
curam.server.jvm.heap.size=1024
```

Abbildung 2. Muster für Anwendungsservereigenschaften

Standardmäßig baut das Ziel **configure** eine Datenquelle mit einem universellen Typ-4-Treiber (XA) auf. Sie können jedoch auch eine Datenquelle mit universellem Typ-2-Treiber (XA) konfigurieren, indem Sie die Eigenschaft 'curam.db.type2.required' in der Datei `AppServer.properties` einrichten.

Außerdem setzt das Ziel **configure** die JVM-Initialen und die maximale Größe des Heapspeichers standardmäßig auf "1024" MB. Man kann diese Standardwerte jedoch überschreiben, indem man die Eigenschaft 'curam.server.jvm.heap.size' in der Datei `AppServer.properties` einrichtet.

Anmerkung:

1. Die Einstellung des Java-Heapspeichers, wie in dem Beispiel unter „Konfiguration von WebSphere Application Server" auf Seite 8 beschrieben und von den Konfigurationsscripts gesetzt, dient nur Anschauungszwecken. Je nach Größe Ihrer angepassten Anwendung, Implementierungsstrategie usw. können diese Einstellungen zu hoch oder zu niedrig sein. Der optimale Wert wird durch das Überwachen der Speicherleistung Ihres Servers bestimmt.
2. Beim Abrufen großer CLOBs und BLOBs (3MB+) aus der Datenbank hat man Speicherprobleme bei den in WebSphere Application Server eingeschlossenen Datenbanktreibern festgestellt. Solche Probleme können umgangen werden, indem man auf dem implementierten Server den JVM-Parameter der maximalen Heapspeichergröße angemessen erhöht.

3. Das Ziel **configure** kann nicht ausgeführt werden, solange die H2-Datenbank im Gebrauch ist.[11]

Sicherheitskonfiguration

Die Standardsicherheitskonfiguration von IBM Cúram Social Program Management innerhalb von WebSphere Application Server bezieht die standarddateibasierte Benutzerregistry und ein JAAS-Anmeldemodul mit ein. Weitere Details hierzu finden Sie im Abschnitt *Default Configuration for IBM WebSphere Application Server* (Standardkonfiguration für IBM WebSphere Application Server) des Dokuments *Cúram Security Handbook* (Cúram-Sicherheitshandbuch).

Es gibt eine Anzahl an alternativen Sicherheitskonfigurationen, die mit WebSphere Application Server verwendet werden können. Die Konfigurationen stehen dafür zur Verfügung, die Verwendung von alternativen Authentifizierungsmechanismen wie einem LDAP-Verzeichnisserver oder einer Single Sign-on-Lösung unterstützen.

Um eine andere Konfiguration zu nutzen, sollten die in den folgenden Abschnitten detailliert beschriebenen Eigenschaften in der Datei `AppServer.properties` gesetzt sein, bevor das Ziel `configure` ausgeführt wird. Alternative Authentifizierungsmechanismen sollten manuell konfiguriert werden, nachdem das Ziel `configure` mit den entsprechend gesetzten Eigenschaften ausgeführt worden ist. Um das Anmeldemodul für eine Authentifizierung nur anhand der Identität zu konfigurieren, muss die Eigenschaft 'curam.security.check.identity.only' auf `true` (wahr) gesetzt sein. Damit wird sichergestellt, dass der konfigurierte alternative Authentifizierungsmechanismus angewendet wird.

Weitere Details finden Sie im Abschnitt *Identity Only Authentication* (Authentifizierung nur anhand der Identität) des Dokuments *Cúram Security Handbook* (Cúram-Sicherheitshandbuch).

Besondere Konfigurationsschritte bei der Verwendung von 'Authentifizierung nur anhand der Identität' und LDAP

Bei der Verwendung von 'Authentifizierung nur anhand der Identität' in Kombination mit WebSphere Application Server und LDAP kann es sein, dass zusätzliche manuelle Konfigurationsschritte ausgeführt werden müssen, unabhängig davon, ob die Konfiguration über die WebSphere Application Server-Administrationskonsole oder das Ziel `configure` vorgenommen wird. Bei einer solchen Kombination stellen Sie möglicherweise fest, dass WebSphere Application Server nicht erfolgreich startet. Das liegt daran, dass das Hinzufügen eines von WebSphere Application Server generierten Benutzernamens zur Ausschlusslisten-Eigenschaft 'exclude_usernames' des Anmeldemoduls notwendig ist, wie unter „Anmeldemodul hinzufügen" auf Seite 28 beschrieben. Wenn der Start von WebSphere Application Server fehlzuschlagen droht, erscheint zuvor eine SECJ0270E-Fehlernachricht in der Datei `SystemOut.log`.

Folgende Schritte sind für die Behebung dieses Problems erforderlich:

1. Es muss der Benutzername identifiziert werden, der das Fehlschlagen des Starts von WebSphere Application Server verursacht. Konfigurieren Sie den Trace des Anmeldemoduls wie unter „Authentifizierungsprozess protokollieren" auf Seite 12 (bezogen auf das Ziel `configure`) oder „Anmeldemodul hinzufügen" auf Seite 28 (bezogen auf das Konfigurieren über die Administrationskonsole) beschrieben und starten Sie WebSphere Application Server erneut. Wenn der Trace

11. Weitere Informationen zur H2-Datenbank finden Sie im Dokument *Cúram Third-Party Tools Installation Guide for Windows* (Cúram-Installationshandbuch für Tools von anderen Anbietern für Windows).

des Anmeldemoduls aktiv ist, ermitteln die Trace-Daten vor Erscheinen der SECJ0270E-Fehlermeldung in der Datei `SystemOut.log` den Benutzernamen, der das Fehlschlagen verursacht, und geben einen Eintrag ähnlich dem folgenden aus:

```
SystemOut     O Username: server:MyNodeCell_MyNode_CuramServer
```

Wobei "MyNode" der Knotenname, "MyNodeCell" der Zellenname und "CuramServer" der Name des WebSphere-Servers ist. Auf die Trace-Daten des Anmeldemoduls folgt der Fehler, der folgendermaßen aussieht:

```
SECJ0270E: Failed to get actual credentials.
   Die Ausnahmebedingung ist 'javax.security.auth.login.LoginException':
   Kontext: MyNodeCell/nodes/MyNode/servers/CuramServer,
   Name: curamejb/LoginHome:
   Erste Komponente im Namen 'curamejb/LoginHome' nicht gefunden.
```

2. Geben Sie den Benutzernamen, der das Fehlschlagen des Starts verursacht, in der Eigenschaft 'exclude_usernames' des Anmeldemoduls in der WebSphere Application Server-Konfiguration an. Da das Starten von WebSphere Application Server fehlschlägt, können Sie diese Änderung nicht über die Administrationskonsole vornehmen und müssen die Konfigurationsdatei von WebSphere Application Server direkt bearbeiten. Bearbeiten Sie im Konfigurationsdateisystem von WebSphere Application Server die Datei `config\cells\MyNodeCell\security.xml`, in der die Eigenschaft 'exclude_usernames' dreimal vorhanden ist (für jeden Alias einmal), z.B.:

```
<options xmi:id="Property_1301940482165"
   name="exclude_usernames"
   value="websphere,db2admin"
   required="false"/>
```

Alle drei Vorkommen müssen modifiziert werden, so dass sie den neu ermittelten Benutzernamen aus dem obigen Traceeintrag einschließen, z.B.:

```
<options xmi:id="Property_1301940482165"
   name="exclude_usernames"
   value="websphere,db2admin,server:MyNodeCell_MyNode_CuramServer"
   required="false"/>
```

Beachten Sie, dass in den Vorkommen der Eigenschaft 'exclude_usernames' das ID-Attribut je nach Ihrer Systemkonfiguration unterschiedlich sein kann und das Kommatrennzeichen des Beispielwertattributs den Standardwert 'curam.security.usernames.delimiter' darstellt, der in Ihrem Fall anders ausfallen kann.

3. Starten Sie WebSphere Application Server erneut.

WebSphere Application Server-Benutzerregistry

Standardmäßig wird die konfigurierte WebSphere Application Server-Benutzerregistry nicht als Teil der Authentifizierung abgefragt. Das geschieht nur, wenn das Anmeldemodul für eine Authentifizierung nur anhand der Identität konfiguriert ist. Durch Einstellen der Eigenschaft 'curam.security.user.registry.enabled' ist es möglich, dieses Standardverhalten zu überschreiben. Ist diese Eigenschaft auf `true` (wahr) gesetzt, wird die WebSphere Application Server-Benutzerregistry während des Authentifizierungsprozesses abgefragt, unabhängig davon, ob die "Authentifizierung nur anhand der Identität" aktiviert ist oder nicht. Ist die Eigenschaft auf `false` (falsch) gesetzt, wird die WebSphere Application Server-Benutzerregistry nicht abgefragt. Beispiel: Ist 'curam.security.check.identity.only' auf `true` und 'curam.security.user.registry.enabled' auf `false` gesetzt, so werden weder die IBM

Cúram Social Program Management-Authentifizierungsverifizierungen noch die WebSphere Application Server-Benutzerregistry als Teil des Authentifizierungsprozesses verwendet.

Die Authentifizierung von Typen externer Benutzer (d.h. nicht-interner Benutzer) anhand der WebSphere Application Server-Benutzerregistry kann auch über die Verwendung der Eigenschaften 'curam.security.user.registry.enabled.types' und/oder curam.security.user.registry.disabled.types' gesteuert werden. Diese Eigenschaften geben eine durch Kommas begrenzte Liste von externen Benutzertypen an, die über die WebSphere Application Server-Benutzerregistry authentifiziert bzw. nicht authentifiziert werden:

- Benutzertypen, die in der Liste 'curam.security.user.registry.enabled.types' angegeben sind, werden anhand der WebSphere Application Server-Benutzerregistry (z.B. LDAP) und Ihrer `ExternalAccessSecurity`-Implementierung verarbeitet.
- Benutzertypen, die in der Liste 'curam.security.user.registry.disabled.types' angegeben sind, werden nicht anhand der WebSphere Application Server-Benutzerregistry verarbeitet. Stattdessen wird die Verarbeitung Ihrer `ExternalAccessSecurity`-Implementierung für sie zur entscheidenden Instanz für die Authentifizierung.

Die Rangfolge für die Verarbeitung dieser drei Eigenschaften und der WebSphere Application Server-Benutzerregistry bzw. externen (z.B. LDAP-) Registry ist folgende:

- Standardmäßig ist die WebSphere Application Server-Benutzerregistry nicht aktiviert; stattdessen wird die Authentifizierung der Anwendung verwendet.
- Ist die Eigenschaft 'curam.security.user.registry.enabled' auf `true` gesetzt, wird die Authentifizierung sowohl über die WebSphere Application Server- bzw. die externe (z.B. LDAP-) Benutzerregistry als auch über die Anwendungssicherheit (für interne Benutzer) bzw. Ihre `ExternalAccessSecurity`-Implementierung (für externe Benutzer) erforderlich.
- Ein externer Benutzer eines Typs, der in der Liste 'curam.security.user.registry.enabled.types' angegeben ist, muss über die WebSphere Application Server- bzw. die externe Benutzerregistry sowie Ihre `ExternalAccessSecurity`-Implementierung authentifiziert werden.
- Ein externer Benutzer eines Typs, der in der Liste 'curam.security.user.registry.disabled.types' angegeben ist, wird nicht über die WebSphere Application Server- bzw. externe Benutzerregistry authentifiziert, sondern Ihre `ExternalAccessSecurity`-Implementierung wird für ihn zur entscheidenden Instanz.

Siehe „JAAS-Anmeldemodul für das System einrichten" auf Seite 28 für weitere Informationen zum Einstellen der resultierenden Eigenschaften in der `CuramLoginModule`-Konfiguration.

Authentifizierungsprozess protokollieren

Die optionale Eigenschaft 'curam.security.login.trace' ermöglicht dem Anmeldemodul das Protokollieren. Auf `true` (wahr) gesetzt, bewirkt sie, dass während des Authentifizierungsprozesses Informationen aus der Traceerstellung zur Datei `SystemOut.log` von WebSphere Application Server hinzugefügt werden.

Alternativen Begrenzer zum Ausschließen von Benutzernamen erstellen

Die optionale Eigenschaft 'curam.security.usernames.delimiter' ermöglicht es, einen alternativen Begrenzer für die Benutzernamenliste in der Eigenschaft 'exclude_usernames' einzurichten. Die Eigenschaft kann auf ein Zeichen festgelegt werden, das Benutzernamen mit eingebetteten Kommas wie beim LDAP zulässt.

Caching-Verhalten von WebSphere Application Server

WebSphere Application Server speichert Benutzerdaten und Identifikationsdaten in einem Sicherheitscache und das Anmeldemodul wird nicht aufgerufen, solange ein Benutzereintrag in diesem Cache gültig ist. Die standardmäßig eingestellte Inaktivierungszeit beträgt für diesen Sicherheitscache zehn Minuten. Weitere Informationen zu diesem Thema finden Sie im Abschnitt *Caching-Verhalten von WebSphere* des Dokuments *Cúram Security Handbook* (Cúram-Sicherheitshandbuch).

Angepasste Sicherheitseigenschaften

- `com.ibm.ws.security.webChallengeIfCustomSubjectNotFound`

  Diese Eigenschaft bestimmt das Verhalten einer LTPA-Token2-Anmeldung mit Single Sign-on.

  Wenn diese Eigenschaft den Wert `true` (wahr) hat, das Token einen angepassten Cache-Schlüssel enthält und das angepasste Subjekt nicht gefunden wird, wird das Token für eine direkte Anmeldung verwendet, da die angepassten Informationen erneut erfasst werden müssen. Es erscheint eine Aufforderung an den Benutzer zur erneuten Anmeldung. Ist der Wert dieser Eigenschaft auf `false` (falsch) gesetzt und das angepasste Subjekt wird nicht gefunden, so wird das LTPA-Token2 für die Anmeldung und die Erfassung aller Registry-Attribute verwendet. Das Token ist jedoch unter umständen nicht in der Lage, bestimmte Attribute abzurufen, die von Downstream-Anwendungen erwartet werden.

  Standardmäßig setzt das Konfigurationsscript die WebSphere Application Server-Eigenschaft 'com.ibm.ws.security.webChallengeIfCustomSubjectNotFound' auf `false`, um sicherzugehen, dass Websitzungen nahtlos zwischen den zwei Servern eines entsprechend übertragen werden können (z.B. in einem Übernahmeszenario), ohne dass der Benutzer zur Eingabe von Sicherheitsberechtigungsnachweisen aufgefordert wird. Diese Einstellung ermöglicht, dass das von WebSphere Application Server verwendete Sicherheitstoken ordnungsgemäß und ohne Benutzereingabe ausgewertet wird.

  Ist ein solches Verhalten nicht erforderlich, so ist es möglich, den Wert dieser Eigenschaft in 'true' zu ändern. Weitere Informationen zum Einstellen der *angepassten Sicherheitseigenschaften* finden Sie unter „JAAS-Anmeldemodul für das System einrichten" auf Seite 28. Ist der Wert der Eigenschaft auf `true` gesetzt und eine Websitzung wechselt von einem Server zu einem anderen Server innerhalb des Clusters, z.B. aufgrund eines Ausfalls des ursprünglichen Servers, so wird der Benutzer zum Eingeben von Sicherheitsinformationen aufgefordert, bevor er fortfahren kann.

Sicherheitsmaßnahmen zur Abschottung des Systems

Wenn sich ein Benutzer bei der Anwendung anmeldet, gibt er Benutzernamen und Kennwort an. Diese werden an den Server geschickt, der nach erfolgreicher Authentifizierung mit einem eindeutigen Token antwortet. In diesem Fall ist es das 'LTPA-Token'. Es wird für alle nachfolgenden Anforderungen zur Benutzererkennung verwendet und stellt dann privilegierten Inhalt bereit. Man sollte annehmen, dass dieses Token beim Abmelden des Benutzers ungültig wird. Dies ist jedoch nicht der Fall. Es besteht keine Möglichkeit, das LTPA-Token ungültig zu machen, was von IBM bestätigt wurde. **Die Empfehlung von Seiten der IBM ist, die zwei folgenden 'Sicherheitsmaßnahmen zur Abschottung des Systems' vorzunehmen:**

1. Einstellen der Sicherheitsoption 'Erfordert SSL'
2. Einstellen einer angepassten Eigenschaft, mit der LTPA-Cookies auf SSL beschränkt werden.

Diese Änderungen werden mithilfe der Standardkonfigurationsscripts vorgenommen. Die erforderlichen Schritte dazu sind unter „Verwaltungssicherheit konfigurieren” auf Seite 26 beschrieben.

Weitere Informationen finden Sie in:

- http://www.ibm.com/developerworks/websphere/techjournal/1004_botzum/1004_botzum.html?ca=drs#step19
- http://www.ibm.com/developerworks/websphere/techjournal/1004_botzum/1004_botzum.html?ca=drs#step29

Cúram-Kryptografie

Der Begriff der Cúram-Kryptografie bezieht sich auf die Funktionalität für die Verwaltung von Kennwörtern und wird im Dokument *Cúram Security Handbook* ausführlich erläutert, das Sie insbesondere unter Berücksichtigung der folgenden Punkte in Betracht ziehen sollten:

- Bei Produktionsumgebungen wird dringend empfohlen, die Standardeinstellungen zu ändern.
- Bei Entwicklungs- und Testumgebungen müssen Sie abwägen, in welchen Bereichen die Standardwerte ausreichend Schutz in Ihrer Umgebung sicherstellen.
- Für Benutzer, die ein Upgrade von einer Vorgängerversion von IBM Cúram Social Program Management ausführen, funktionieren die vorhandenen Kennwörter nicht ohne Vorbereitungs- oder Anpassungsaufwand. Sie können, wenn Sie bereit sind, ein geringeres Maß an Sicherheit in Kauf zu nehmen, auf eigenes Risiko die entsprechenden Schritte ausführen, um das vorhandene System und die Benutzerkennwörter unverändert zu belassen, doch dies wird nicht empfohlen. Weitere Informationen zu Upgrades enthält das Handbuch *Cúram Upgrade Guide*.

Zeitzonenkonfiguration

Bei der Verwendung mehrerer Servermaschinen müssen alle ihre Taktgeber synchronisiert und auf dieselbe Zeitzone eingestellt sein, so dass die "natürliche" Anordnung von Daten und Uhrzeiten in der Datenbank die Reihenfolge der Ereignisse in der realen Welt genau widerspiegelt. Beispiel: Wenn in Datenbanksatz *A* ein Erstellungsdatum oder eine Erstellungszeit früher angegeben ist als in Datensatz *B*, kann man mit Sicherheit sagen, dass *A* vor *B* erstellt wurde, unabhängig davon, welcher Server den einen oder anderen Datensatz erstellt hat.

Die Zeitzone des Servers bzw. der Server darf während der Laufzeit einer Anwendung niemals geändert werden. Der Grund hierfür liegt darin, dass die zum Zeitpunkt der Speicherung der Daten in der Datenbank vorausgesetzte Zeitzone die Zeitzone des aktuellen Servers ist. Wenn also die Zeitzone des Servers geändert wird, dann weichen alle vor der Änderung der Zeitzone eingegebenen Daten um die Stundendifferenz zwischen der neuen und der alten Zeitzone ab.

WebSphere Server starten und stoppen

Es wird eine Anzahl von Ant-Zielen bereitgestellt, die das Starten und Stoppen von WebSphere Servern unterstützen. Diese Ziele sollten vom Verzeichnis `<SERVER_DIR>` aus ausgeführt werden. Was das Ziel **`configure`** betrifft, benötigen sie die Datei `AppServer.properties`, um ordnungsgemäß eingerichtet zu werden (siehe auch unter „Konfiguration von WebSphere Application Server” auf Seite 8). Ebenso erfordern sie die Angabe einer Anzahl von zusätzlichen Parametern, die unten detailliert aufgeführt sind.

WebSphere Server starten

Das Ziel zum Starten eines WebSphere Servers lautet **startserver** und erfordert die folgenden Optionen:

- `-Dserver.name`

  Der Name des zu startenden Servers.

Wichtig: Vor dem ersten Starten des Anwendungsservers muss das Ziel **database** ausgeführt worden sein, gefolgt von dem Ziel **prepare.application.data**. Werden sie in anderer Reihenfolge ausgeführt, führt das mit einiger Wahrscheinlichkeit zu Transaktionszeitlimits bei der ersten Anmeldung und einem Fehlschlagen der Initialisierung und des Zugriffs auf die Anwendung. Bei jeder erneuten Ausführung des Ziels **database** (z.B. in einer Entwicklungsumgebung) muss auch das Ziel **prepare.application.data** mit ausgeführt werden.

```
build startserver -Dserver.name=CuramServer
```

Abbildung 3. Anwendungsbeispiel

WebSphere Server stoppen

Das Ziel zum Stoppen eines WebSphere Servers lautet **stopserver** und erfordert die folgenden Optionen:

- `-Dserver.name`

  Der Name des zu stoppenden Servers.

```
build stopserver -Dserver.name=CuramServer
```

Abbildung 4. Anwendungsbeispiel

WebSphere Server erneut starten

Das Ziel für den Neustart eines WebSphere-Servers ist **restartserver**. Die Optionen sind dieselben wie für das Ziel **startserver**. Unter „WebSphere Server starten" finden Sie ein Anwendungsbeispiel.

Anmerkung: Wenn der Server beim Neustartversuch nicht bereits gestartet ist, bewirkt der Sperrabschnitt des Ziels kein Fehlschlagen des Neustart-Ziels.

Implementierung

Einführung

Der letzte Schritt nach dem Paketieren der IBM Cúram Social Program Management-Anwendung und der Web-Service-Anwendung in EAR-Dateien ist ihre Implementierung auf dem Anwendungsserver.

Vor dem Implementieren ist es wichtig zu beachten, dass die in WebSphere Application Server bereitgestellten Konfigurationsscripts eine einfache Konfiguration unterstützen, die auf den Einzelserver in den Express- oder Basis-Editionen von WebSphere Application Server ausgerichtet ist.

Implementierung

Das SDEJ stellt Ziele zum Installieren und Deinstallieren von Anwendungen auf einem WebSphere-Server bereit. Wie die Ziele **startserver** / **stopserver**, so erfordern auch die Ziele **installapp** / **uninstallapp**, dass die Datei `AppServer.properties` ordnungsgemäß konfiguriert ist (siehe auch „Konfiguration

von WebSphere Application Server" auf Seite 8). Diese Ziele erfordern auch die Angabe einer Anzahl von Optionen, die unten aufgeführt sind.

Stellen Sie sicher, dass vor der Installation der Anwendung der Server gestartet worden ist. Da das Installationsziel die Anwendung automatisch startet, muss der Server nach der Installation nicht erneut gestartet werden.

Anwendung installieren

Das Ant-Ziel zum Installieren einer Anwendung (in Form einer EAR-DAtei) ist **installapp**. Es erfordert die folgenden Optionen:

- `-Dserver.name`

  Der Name des Servers, auf dem die Anwendung installiert werden soll.
- `-Dear.file`

  Der vollständig qualifizierte Name der zu installierenden EAR-Datei.
- `-Dapplication.name`

  Der Name der Anwendung.

```
build installapp -Dserver.name=CuramServer
 -Dear.file=d:/ear/Curam.ear
 -Dapplication.name=Curam
```

Abbildung 5. Anwendungsbeispiel

Anmerkung: Die EAR-Datei, die das Servermodul enthält, muss implementiert sein, bevor andere (nur-Client-) EAR-Dateien installiert werden.

Um zusätzliche Argumente an das WebSphere-Tool 'wsadmin' zu übergeben, steht die optionale Ant-Eigenschaft namens `wsadmin.extra.args` zur Verfügung. Mit dem folgenden Befehl werden zum Beispiel neue Größenangaben für den Java™-Heapspeicher festgelegt und die Option zum Anhängen von Tracing mit wsadmin übergeben:

```
-Dwsadmin.extra.args="-javaoption -Xms1024m -javaoption -Xmx1024m -appendtrace true"
```

Diese Eigenschaft sollte nicht verwendet werden, um Argumente anzugeben, die bereits über die Ant-Scripts von Cúram übergeben wurden. Welche Argumente bei der Ausführung von Ant übergeben werden, können Sie feststellen, wenn Sie mit der Option `-v` die ausführliche Anzeige anfordern.

SYSTEM-Benutzernamen ändern

Es wird dringend empfohlen, den Benutzernamen für den JMS-Aufruf im Zuge der Implementierung der Anwendung zu ändern. Um diesen Benutzernamen ändern zu können, sollten vor der Implementierung die folgenden Eigenschaften in der Datei `AppServer.properties` gesetzt sein:

- curam.security.credentials.async.username

  Der Benutzername, unter dem die JMS-Aufrufe ausgeführt werden sollten.
- curam.security.credentials.async.password

  Das verschlüsselte Kennwort, das dem Benutzernamen zugehörig ist. Das Kennwort sollte mithilfe des Ant-Ziels **encrypt** verschlüsselt werden. Im Dokument *Cúram Server Developers Guide* finden Sie hierzu weitere Informationen.

Mithilfe der WebSphere Application Server-Administrationskonsole besteht auch die Möglichkeit, den Benutzernamen nach der Implementierung der Anwendung zu ändern. Navigieren Sie dafür zu **Anwendungen** > **Anwendungstypen** > **Web-**

Sphere-Unternehmensanwendungen und wählen Sie die Anwendung IBM Cúram Social Program Management aus. Wählen Sie den Link **RunAs-Rollen für Benutzer** aus. Aktivieren Sie die Rolle `everyone` (jeder), geben Sie einen neuen Benutzernamen und ein neues Kennwort ein (wobei zu beachten ist, dass das Kennwort hier in unverschlüsseltem Format eingegeben werden sollte) und klicken Sie auf die Schaltfläche **Anwenden**. Speichern Sie die Änderungen wie unter „Masterkonfiguration speichern" auf Seite 26 im Detail beschrieben.

Beachten Sie, dass nach dem Ändern des Benutzernamens der neue Benutzername in der Benutzerdatenbanktabelle vorhanden sein und dieser Benutzer die Rolle 'SUPERROLE' besitzen muss.

Der SYSTEM-Benutzer ist der Benutzer, unter dem die JMS-Nachrichten ausgeführt werden.

Anwendung deinstallieren

Das Ant-Ziel zum Deinstallieren einer Anwendung ist **`uninstall`**. Es erfordert die folgenden Optionen:

- `-Dserver.name`

  Der Name des Servers, auf dem die Anwendung installiert ist.
- `-Dapplication.name`

  Der Name der Anwendung, die deinstalliert werden soll (wie bei der Installation konfiguriert).

```
build uninstallapp -Dserver.name=CuramServer
-Dapplication.name=Curam
```

Abbildung 6. Anwendungsbeispiel

JSPs vorkompilieren

Es gibt ein zusätzliches Ziel, **`precompilejsp`**, das während der Implementierung verfügbar ist. Es ermöglicht die Vorkompilierung der JSPs einer Client-EAR-Datei *vor* der Installation der EAR-Datei. Das Vorkompilieren der JSPs vor der Installation verschnellert die Anzeige einer bestimmten Seite im Web-Browser beim ersten Zugriff.

Die Optionen für das Ziel **`precompilejsp`** sind:

- `-Dear.file`

  Der vollständig qualifizierte Name der vorzukompilierenden EAR-Datei.

```
build precompilejsp -Dear.file=d:/Curam.ear
```

Abbildung 7. Anwendungsbeispiel

Anmerkung: Beim Ausführen des Ziels **`precompilejsp`** für WebSphere Application Server kann es vorkommen, dass eine Hauptspeicher-Ausnahmebedingung auftritt oder einige JSPs im Hintergrund ignoriert und nicht vorkompiliert werden. Zur Problemumgehung sollte das Script `JspBatchCompiler.bat` im Verzeichnis `%WAS_HOME%\bin` dahingehend modifiziert werden, dass die maximale Hauptspeicherkapazität erhöht wird. Ändern Sie die Speicherbelegung von `-Xmx256m` auf mindestens `-Xmx1024m`.

Implementierung testen

Nachdem die Anwendung auf einer konfigurierten WebSphere Application Server-Installation installiert ist[12], besteht der nächste Schritt darin, die Anwendung zu starten und zu testen.

Stellen Sie sicher, dass der entsprechende Server gestartet ist[13], und öffnen Sie in einem Web-Browser die folgende Seite:

```
https://<some.machine.com>:<port>/<context-root>
```

Wobei

<*some.machine.com*> den Hostnamen oder die IP-Adresse angibt, wo Ihr WebSphere Application Server ausgeführt wird, <*port*> den Server-Port, auf dem die Clientanwendung implementiert ist, und <*context-root*> den Stammkontext des WAR-Moduls (weitere Details unter „Mehrfache EAR-Dateien" auf Seite 6).

Bevor die Seite geöffnet werden kann, wird der Browser zur Anmeldeseite geleitet. Melden Sie sich mit einem gültigen Cúram-Benutzernamen und Kennwort an. Der Browser wird auf die angeforderte Seite umgeleitet.

Anmerkung: Die Verwendung des EAR-Dateinamens `Curam.ear` für die Option `-Dear.file` und des Anwendungsservernamens `Curam` für die Option `-Dapplication.name` in den Beispielen dieses Kapitels dient Anschauungszwecken. Diese Werte können je nach Ihrer angepassten Anwendung und Implementierungsstrategie unterschiedlich sein.

IBM WebSphere Application Server mit der USGCB verwenden

Bei der "United States Government Configuration Baseline" (Konfigurationsbaseline der Regierung der Vereinigten Staaten, USGCB) handelt es sich um eine Initiative der US-Regierung zur Bereitstellung von Anleitungen für Behörden mit dem Ziel der Verbesserung von Konfigurationseinstellungen, vor allem im Bereich der Sicherheit. Bei der Ausführung der Anwendung IBM Cúram Social Program Management unter Verwendung von IBM WebSphere Application Server V7 (siehe dazu das Handbuch *IBM Cúram Social Program Management v6 Supported Prerequisites* zu unterstützten Versionen von IBM WebSphere Application Server V7) mit USGCB-Einstellungen kann es vorkommen, dass Grafiken fehlen. Das Auftreten eines solchen Fehlers zeigt an, dass IBM WebSphere Application Server keine PNG-Dateien erkennt. Zur Behebung dieses Problems muss IBM WebSphere Application Server aktualisiert werden, so dass er den PNG-MIME-Typ unterstützt. Nähere Einzelheiten hierzu finden Sie in der Dokumentation des *WebSphere Application Server Information Center*.

Weitere Informationen zur USGCB finden Sie auf der folgenden Website: http://usgcb.nist.gov/

WebSphere Application Server manuell konfigurieren

Einführung

In den Abschnitten dieses Kapitels werden die manuellen Schritte beschrieben, die für das Konfigurieren und Implementieren auf der Basis- oder Express-Edition von WebSphere Application Server erforderlich sind. Für eine Implementierung in einer

12. Die Installation einer Web-Service-Anwendung kann ebenfalls erforderlich sein.

13. Nach dem Implementieren einer Anwendung muss der Server nicht erneut gestartet werden.

Network Deployment-Installation von WebSphere Application Server müssen diese Schritte entsprechend geändert werden. Unter „WebSphere Network Deployment" auf Seite 42 finden Sie weitere Informationen zu diesem Bereich.

WebSphere Application Server manuell konfigurieren

Die IBM WebSphere Application Server-Installation kann bei Bedarf manuell konfiguriert werden, jedoch ist dies nicht zu empfehlen. Nur zu Informationszwecken werden in diesem Abschnitt die manuellen Schritte beschrieben, die für das Konfigurieren von WebSphere Application Server erforderlich sind.

Es ist zu beachten, dass alle Einstellungen, die unter dem Abschnitt **Ressourcen** der WebSphere Application Server-Administrationskonsole eingegeben werden, auf mehreren Ebenen konfiguriert werden können, die den JNDI(Java Naming and Directory Interface)-Bereich steuern. Das kann eine Zelle, ein Knoten oder ein Server sein. Nach der Auswahl einer **Ressource** wird dieser Bereich oben im Hauptbrowserfenster angezeigt, wodurch es möglich ist, die verschiedenen Ressourcen im aktuellen Bereich anzuzeigen. Der Bereich und damit die Speicherposition aller eingestellten Ressourcen sollte auf der geplanten Verwendung beruhen. Wenn in einem Cluster gearbeitet wird, ist es möglicherweise nicht notwendig, für jeden Server dieselben Einstellungen einzugeben, und der Bereich kann auf Zelle oder Knoten gesetzt werden.

Administrationskonsole

Die meisten Konfigurationen für WebSphere Application Server werden mithilfe der Administrationskonsole erstellt. Um die Administrationskonsole auszuführen, muss erst der Standardserver, z.B. `server1`, gestartet werden, da die Administrationskonsole als Webanwendung auf diesem Server installiert ist.

Zum Starten von `server1` sollte `startServer.bat` verwendet werden, das sich im Verzeichnis `profiles/AppSvr01/bin` der WebSphere Application Server-Installation befindet:

```
<WEBSPHERE INSTALL DIR>/profiles/AppSvr01/bin/startServer server1
```

Um die Administrationskonsole zu öffnen, sollte der Web-Browser auf die folgende Adresse verweisen:

`http://localhost:9060/ibm/console"/> verweisen.`

Alternativ kann die Administrationskonsole auch über **Start** > **Programme** > **IBM WebSphere** > **Application Server V7.0** > **Profile** > **AppSvr01** > **Administrationskonsole** gestartet werden. Die Befehle **Server starten** und **Server stoppen** können auch über dieses Menü zum Starten und Stoppen der Server verwendet werden.

Beim ersten Öffnen der Administrationskonsole wird für die Anmeldung nach einem Benutzernamen gefragt. Hierfür kann ein beliebiger Name verwendet werden. Die Administrationskonsole ist in zwei Sektionen unterteilt. Die linke Seite enthält eine Baumhierarchie zum Navigieren durch die Konsole. Die rechte Seite zeigt die Informationen an, die sich auf den aktuell im Baum ausgewählten Knoten beziehen. Auf den Befehl `Navigieren zu` sollte die Baumhierarchie zu dem entsprechenden Knoten traversieren.

Unterstützung zum Erstellen von Scripts

Um die Ausführung der bereitgestellten Ant-Scripts zu unterstützen, sind Änderungen an den Eigenschaftendateien für WebSphere Application Server notwendig.

sas.client.props: Öffnen Sie die Datei `sas.client.props`, die sich im Verzeichnis `profiles/AppSvr01/properties` der WebSphere Application Server-Installation befindet. Um nicht bei jeder Ausführung der Scripts den Benutzernamen und das Kennwort eingeben zu müssen, ist es notwendig, die Anmeldequelle so einzurichten, dass sie Benutzernamen und Kennwort aus einer Eigenschaftendatei abruft. Setzen Sie die folgenden Eigenschaften oder fügen Sie sie bei Bedarf hinzu:

```
com.ibm.CORBA.loginSource=properties
# RMI/IIOP user identity
com.ibm.CORBA.loginUserid=websphere
com.ibm.CORBA.loginPassword=websphere
```

Wobei *websphere* sowohl Benutzername als auch Kennwort für die Administrationskonsole ist.

soap.client.props: Öffnen Sie die Datei `soap.client.props`, die sich auch im Verzeichnis `profiles/AppSvr01/properties` der WebSphere Application Server-Installation befindet. Um nicht bei jeder Ausführung der Scripts den Benutzernamen und das Kennwort eingeben zu müssen, ist es notwendig, die Anmeldequelle so einzurichten, dass sie Benutzernamen und Kennwort aus einer Eigenschaftendatei abruft. Geben Sie die folgenden Eigenschaften so an, dass sie mit den für WebSphere wie in „Konfiguration von WebSphere Application Server" auf Seite 8 konfigurierten Berechtigungsnachweisen übereinstimmen. Bei den Werten in dem nachfolgenden Beispiel handelt es sich einfach um Beispielwerte und das in dieser Datei angegebene Kennwort kann nicht verschlüsselt werden:

```
com.ibm.SOAP.loginUserid=websphere
com.ibm.SOAP.loginPassword=websphere
```

Wobei *websphere* sowohl Benutzername als auch Kennwort für die Administrationskonsole ist.

Um Zeitlimitüberschreitungen bei der Installation von EAR-Dateien zu vermeiden, sollte folgender Wert mindestens auf die folgende Zahl gesetzt sein:

```
com.ibm.SOAP.requestTimeout=3600
```

server.policy: Öffnen Sie die Datei `server.policy`, die sich im Verzeichnis `profiles/AppSvr01/properties` der WebSphere Application Server-Installation befindet. Fügen Sie am Ende dieser Datei die folgenden Zeilen hinzu:

```
grant codeBase "file:<CURAMSDEJ>/drivers/-" {
permission java.security.AllPermission;
};
```

Wobei *<CURAMSDEJ>* das SDEJ-Installationsverzeichnis bezeichnet.

```
grant codeBase "file:${was.install.root}/
profiles/<profile.name>/installedApps/
<cell.name>/<SERVER_MODEL_NAME>.ear/
guice-2.0.jar" { permission java.lang.RuntimePermission
"modifyThread"; permission java.lang.RuntimePermission
"modifyThreadGroup"; },
```

wobei *<profile.name>* der Name des angezielten WebSphere Application Server-Profils,

<cell.name> der Name der angezielten WebSphere Application Server-Zelle und

<*SERVER_MODEL_NAME*> der Name der Anwendung (Basisname der EAR-Datei) ist.

Datenquellen-Anmeldealias erstellen

Informationen zu diesem Vorgang

Unterstützte Datenbanken sind IBM DB2, IBM DB2 für z/OS sowie die Oracle-Datenbank. Die Administrationskonsole kann dazu verwendet werden, einen Anmeldealias sowohl für die DB2- als auch die Oracle-Datenquellen wie folgt zu konfigurieren:

Vorgehensweise

1. Navigieren Sie zu **Sicherheit** > **Globale Sicherheit**.
2. Erweitern Sie die Option **Java Authentication and Authorization Service** im Abschnitt **Authentifizierung** und wählen Sie die Option **J2C-Authentifizierungsdaten**.
3. Klicken Sie auf **Neu**, um die Konfigurationsanzeige zu öffnen.
4. Setzen Sie die folgenden Felder:

   Alias = dbadmin

   Benutzer-ID = <*database username*>

   Kennwort = <*database password*>

   Beschreibung = Alias für Datenbanksicherheit

   Wobei <*database username*> und <*database password*> auf den Benutzernamen und das Kennwort gesetzt sind, die für die Anmeldung bei der Datenbank verwendet werden.
5. Klicken Sie auf **OK**, um die Änderungen zu bestätigen.

DB2-Datenquellen konfigurieren

DB2-Umgebungsvariable einrichten:
Vorgehensweise

1. Navigieren Sie zu **Umgebung** > **WebSphere-Variablen**.
2. Wählen Sie den Link 'DB2UNIVERSAL_JDBC_DRIVER_PATH' aus der Liste von Umgebungsvariablen aus. Damit wird die Konfigurationsanzeige für diese Variable geöffnet.
3. Richten Sie das Feld **Wert** so ein, dass es auf das Verzeichnis verweist, das die Typ-4- bzw. Typ-2-Treiber enthält. Normalerweise ist dies das Verzeichnis `Treiber` unter der SDEJ-Installation, z.B. `D:\Curam\CuramSDEJ\drivers`.
4. Klicken Sie auf **OK**, um die Änderungen zu bestätigen.

Anbieter für Datenbanktreiber einrichten:
Vorgehensweise

1. Navigieren Sie zu **Ressourcen** > **JDBC** > **JDBC-Provider**.
2. *Hinweis:* An dieser Stelle sollte der geeignete Bereich ausgewählt werden, in dem die Datenquelle definiert werden soll.
3. Klicken Sie auf **Neu**, um einen neuen Treiber hinzuzufügen. Damit öffnet sich eine Konfigurationsanzeige.
4. Wählen Sie **DB2** aus der Drop-down-Liste **Datenbanktyp**.
5. Wählen Sie **DB2 Universal JDBC Driver Provider** aus der Drop-down-Liste **Anbietertyp**.

6. Wählen Sie die **XA-Datenquelle** aus der Drop-down-Liste **Implementierungstyp**.
7. Klicken Sie auf **Weiter**, um fortzufahren.
8. Überprüfen Sie die Eigenschaften in der Konfigurationsanzeige, die geöffnet wird. Es wird vermutlich nicht notwendig sein, eine von ihnen zu ändern, sofern Sie nicht planen, eine Verbindung zu einer **zOS**-Datenbank herzustellen. In diesem Fall überprüfen Sie, ob das Feld `${DB2UNIVERSAL_JDBC_DRIVER_PATH}` auf das richtige Verzeichnis für Ihr System verweist. Es sollte beispielsweise auf das Verzeichnis verweisen, das die DB2 Connect-Lizenz-JAR-Datei `db2jcc_license_cisuz.jar` enthält, die von IBM für die **zOS**-Konnektivität zur Verfügung gestellt wird.
9. Klicken Sie auf **Weiter** und anschließend auf **Fertigstellen**, um die Änderungen zu bestätigen.

Datenquelle für den Datenbanktreiber einrichten:
Informationen zu diesem Vorgang

Die folgenden Schritte sollten für jede der Datenquellen wiederholt werden, wobei `curamdb`, `curamsibdb` und `curamtimerdb` für <*DatasourceName*> (ohne spitze Klammern) eingesetzt werden:

Vorgehensweise

1. Wählen Sie die Option `DB2 Universal JDBC Driver Provider (XA)`, die jetzt in der Liste **JDBC-Provider** angezeigt wird. Damit wird die Konfigurationsanzeige für den Anbieter (Provider) geöffnet.
2. Wählen Sie den Link **Datenquellen** unter **Weitere Eigenschaften**.
3. Klicken Sie auf **Neu**, um eine neue Datenquelle hinzuzufügen.
4. Nehmen Sie in den Feldern folgende Einstellungen vor:

   Datenquellenname: <*DatasourceName*>

   JNDI-Name: *jdbc/*<*DatasourceName*>
5. Klicken Sie auf **Weiter**, um fortzufahren.
6. Nehmen Sie in den Feldern folgende Einstellungen vor:

   Treibertyp : 2 oder 4 je nach Erforderlichkeit,

   Datenbankname : Der Name der DB2-Datenbank,

   Servername : Der Name des DB2-Datenbankservers,

   Portnummer : Der DB2-Datenbankserverport.

   Lassen Sie alle anderen Felder unverändert, es sei denn, es ist eine bestimmte Änderung erforderlich. Klicken Sie auf **Weiter**.
7. Setzen Sie den Wert aus der Drop-down-Liste für **Komponentengesteuerter Authentifizierungsalias** auf: <*valid for database*>.

   Setzen Sie den Wert aus der Drop-down-Liste für den **Alias für Konfigurationszuordnung** auf: DefaultPrinicipalMapping

   Setzen Sie **Containergesteuerter Authentifizierungsalias** auf: <*valid for database*>.

   Wobei der verwendete Alias <*valid for database*> der ist, der unter „Datenquellen-Anmeldealias erstellen" auf Seite 21 eingerichtet wird.

   Lassen Sie alle anderen Felder unverändert, soweit keine bestimmte Änderung erforderlich ist. Klicken Sie auf **Weiter**, um fortzufahren.
8. Klicken Sie auf **Fertigstellen**, um die Änderungen zu bestätigen, und fahren Sie fort.

9. Wählen Sie die neu erstellte Datenquelle *DatasourceName* aus der angezeigten Liste.
10. Wählen Sie den Link **Angepasste Eigenschaften** unter **Weitere Eigenschaften**.
11. Wählen Sie den Eintrag `fullyMaterializeLobData`.
12. Setzen Sie den `Wert` auf `false`.
13. Klicken Sie auf **OK**, um die Änderung zu bestätigen.

Oracle-Datenquelle konfigurieren

Oracle-Umgebungsvariable einrichten:
Vorgehensweise

1. Navigieren Sie zu **Umgebung** > **WebSphere-Variablen**.
2. Wählen Sie den Link ORACLE_JDBC_DRIVER_PATH aus der Liste von Umgebungsvariablen aus. Damit wird die Konfigurationsanzeige für diese Variable geöffnet.
3. Richten Sie das Feld **Wert** so ein, dass es auf das Verzeichnis verweist, das den Typ-4-Treiber enthält. Dies ist das Verzeichnis `Treiber` der SDEJ-Installation, z.B. `D:\Curam\CuramSDEJ\drivers`.
4. Klicken Sie auf **OK**, um die Änderungen zu bestätigen.

XA-Datenbanktreiber einrichten:
Vorgehensweise

1. Navigieren Sie zu **Ressourcen** > **JDBC** > **JDBC-Provider**.
2. *Hinweis:* An dieser Stelle sollte der geeignete Bereich ausgewählt werden, in dem die Datenquelle definiert werden soll.
3. Klicken Sie auf **Neu**, um einen neuen Treiber hinzuzufügen. Damit öffnet sich eine Konfigurationsanzeige.
4. Wählen Sie **Oracle** aus der Drop-down-Liste **Datenbanktyp**.
5. Wählen Sie **Oracle-JDBC-Treiber** aus der Drop-down-Liste **Anbietertyp**.
6. Wählen Sie die **XA-Datenquelle** aus der Drop-down-Liste **Implementierungstyp**.
7. Setzen Sie das Feld **Name** auf Oracle-JDBC-Treiber (XA), sofern es nicht automatisch ausgefüllt ist.
8. Klicken Sie auf **Weiter**, um fortzufahren.
9. Überprüfen Sie den **Klassenpfad** und stellen Sie sicher, dass die Umgebungsvariable `ORACLE_JDBC_DRIVER_PATH` korrekt ist. Klicken Sie auf **Weiter**, um fortzufahren.
10. Überprüfen Sie die Eigenschaften in der Konfigurationsanzeige, die geöffnet wird. Es wird vermutlich nicht notwendig sein, eine von ihnen zu ändern.
11. KLicken Sie auf **Fertigstellen**, um die Änderungen zu bestätigen.

Nicht-XA-Datenbanktreiber einrichten:
Vorgehensweise

1. Navigieren Sie zu **Ressourcen** > **JDBC** > **JDBC-Provider**.
2. Klicken Sie auf **Neu**, um einen neuen Treiber hinzuzufügen. Damit öffnet sich eine Konfigurationsanzeige.
3. Wählen Sie **Oracle** aus der Drop-down-Liste **Datenbanktyp**.
4. Wählen Sie **Oracle-JDBC-Treiber** aus der Drop-down-Liste **Anbietertyp**.
5. Wählen Sie die **Datenquelle des Verbindungspools** aus der Drop-down-Liste **Implementierungstyp**.

6. Setzen Sie das Feld **Name** auf Oracle JDBC-Treiber, sofern es nicht automatisch ausgefüllt ist.
7. Klicken Sie auf **Weiter**, um fortzufahren.
8. Überprüfen Sie den **Klassenpfad** und stellen Sie sicher, dass die Umgebungsvariable `ORACLE_JDBC_DRIVER_PATH` korrekt ist. Klicken Sie auf **Weiter**, um fortzufahren.
9. Überprüfen Sie die Eigenschaften in der Konfigurationsanzeige, die geöffnet wird. Es wird vermutlich nicht notwendig sein, eine von ihnen zu ändern.
10. KLicken Sie auf **Fertigstellen**, um die Änderungen zu bestätigen.

Datenquellen für XA-Datenbanktreiber einrichten:
Informationen zu diesem Vorgang

Die folgenden Schritte sollten zweimal wiederholt werden, indem man *<DatasourceName>* (ohne spitze Klammern) durch die Option`curamdb` und anschließende Option`curamsibdb` ersetzt.

Vorgehensweise

1. Wählen Sie die Option`Oracle-JDBC-Treiber (XA)`, die jetzt in der Liste vorhandener Anbieter angezeigt wird. Damit öffnet sich wieder die Konfigurationsanzeige.
2. Wählen Sie den Link **Datenquellen** unter **Weitere Eigenschaften**.
3. Klicken Sie auf **Neu**, um eine neue Datenquelle hinzuzufügen.
4. Nehmen Sie in den Feldern folgende Einstellungen vor:

   Datenquellenname: *<DatasourceName>*

   JNDI-Name : *jdbc/<DatasourceName>*

   Klicken Sie auf **Weiter**.
5. Setzen Sie das **URL**-Wertfeld auf

   `jdbc:oracle:thin:@//serverName:port/databaseServiceName`, um mithilfe des Oracle-Servicenamens eine Verbindung zur Datenbank herzustellen,

   oder auf

   `jdbc:oracle:thin:@serverName:port:databaseName`, um mithilfe des Oracle-SID-Namens eine Verbindung zur Datenbank herzustellen.

   Dabei gilt Folgendes:

   serverName ist der Name der Servers, auf dem sich die Datenbank befindet.

   port ist die Portnummer, für die die Datenbank empfangsbereit ist.

   databaseName ist die SID der Datenbank.

   databaseServiceName ist der Servicename der Datenbank.

   Setzen Sie **Name der Helper-Klasse für Datenspeicher** auf `Helper-Klasse für Datenspeicher Oracle11g`.

   Lassen Sie alle anderen Felder unverändert, es sei denn, es ist eine bestimmte Änderung erforderlich. Klicken Sie auf **Weiter**.

   Anmerkung: Oracle empfiehlt die Verwendung des **URL**-Formats `jdbc:oracle:thin:/@//serverName:port/databaseServiceName` für die Verbindung zur Oracle-Datenbank mithilfe des Servicenamens. Dieses **URL**-Format (zusätzliches '/' vor dem '@' in der **URL**) wird von der WebSphere-Administrationskonsole jedoch nicht unterstützt. Deshalb sollte die oben beschriebene Oracle-Servicenamen-**URL** (ohne zusätzliches '/' vor dem '@' in der URL) wäh-

rend der Konfiguration der Oracle-Datenquelle von der Administrationskonsole aus verwendet werden, um mithilfe des Servicenamens eine Verbindung zur Oracle-Datenbank herzustellen.

6. Setzen Sie den **Authentifizierungsalias für XA-Wiederherstellung** auf: *<valid for database>*

   Setzen Sie den Wert aus der Drop-down-Liste für **Komponentengesteuerter Authentifizierungsalias** auf: *<valid for database>*.

   Wobei der verwendete Alias *<valid for database>* der ist, der unter „Datenquellen-Anmeldealias erstellen" auf Seite 21 eingerichtet wird.

   Lassen Sie alle anderen Felder unverändert, es sei denn, es ist eine bestimmte Änderung erforderlich. Klicken Sie auf **Weiter**.

7. Klicken Sie auf **Fertigstellen**, um die Änderungen zu bestätigen, und fahren Sie fort.

Datenquelle für nicht-XA-Datenbanktreiber einrichten: Vorgehensweise

1. Wählen Sie die Option `Oracle-JDBC-Treiber`, die jetzt auf der Liste vorhandener Anbieter angezeigt wird. Damit öffnet sich wieder die Konfigurationsanzeige.
2. Wählen Sie den Link **Datenquellen** unter **Weitere Eigenschaften**.
3. Klicken Sie auf **Neu**, um eine neue Datenquelle hinzuzufügen.
4. Nehmen Sie in den Feldern folgende Einstellungen vor:

   Name der Datenquelle :`curamtimerdb`

   JNDI-Name :`jdbc/curamtimerdb`

   Klicken Sie auf **Weiter**.

5. Setzen Sie das **URL**-Wertfeld auf

   `jdbc:oracle:thin:@//serverName:port/databaseServiceName`, um mithilfe des Oracle-Servicenamens eine Verbindung zur Datenbank herzustellen,

   oder auf

   `jdbc:oracle:thin:@serverName:port:databaseName`, um mithilfe des Oracle-SID-Namens eine Verbindung zur Datenbank herzustellen.

   Dabei gilt Folgendes:

   serverName ist der Name der Servers, auf dem sich die Datenbank befindet.

   port ist die Portnummer, für die die Datenbank empfangsbereit ist.

   databaseName ist die SID der Datenbank.

   databaseServiceName ist der Servicename der Datenbank.

   Setzen Sie **Name der Helper-Klasse für Datenspeicher** auf `Helper-Klasse für Datenspeicher Oracle11g`.

   Lassen Sie alle anderen Felder unverändert, es sei denn, es ist eine bestimmte Änderung erforderlich. Klicken Sie auf **Weiter**.

   Anmerkung: Oracle empfiehlt die Verwendung des **URL**-Formats `jdbc:oracle:thin:/@//serverName:port/databaseServiceName` für die Verbindung zur Oracle-Datenbank mithilfe des Servicenamens. Dieses **URL**-Format (zusätzliches '/' vor dem '@' in der **URL**) wird von der WebSphere-Administrationskonsole jedoch nicht unterstützt. Deshalb sollte die oben beschriebene Oracle-Servicenamen-**URL** (ohne zusätzliches '/' vor dem '@' in der URL) während der Konfiguration der Oracle-Datenquelle von der Administrationskonsole aus verwendet werden, um mithilfe des Servicenamens eine Verbindung zur Oracle-Datenbank herzustellen.

6. Setzen Sie den Wert aus der Drop-down-Liste für **Komponentengesteuerter Authentifizierungsalias** auf: *<valid for database>*.

   Wobei der verwendete Alias *<valid for database>* der ist, der unter „Datenquellen-Anmeldealias erstellen" auf Seite 21 eingerichtet wird.

   Lassen Sie alle anderen Felder unverändert, es sei denn, es ist eine bestimmte Änderung erforderlich. Klicken Sie auf **Weiter**.
7. Klicken Sie auf **Fertigstellen**, um die Änderungen zu bestätigen, und fahren Sie fort.

Masterkonfiguration speichern

Das *Speichern* wird durch Klicken auf den Link **Speichern** im Nachrichtenfenster **Nachricht(en)** ausgeführt. Dieses Fenster wird nur nach dem Vornehmen der Konfigurationsänderungen angezeigt.

Verwaltungssicherheit konfigurieren

Informationen zu diesem Vorgang

Standardmäßig wird von WebSphere Application Server die dateibasierte Benutzerregistry verwendet.

Vorgehensweise

1. Navigieren Sie zu **Sicherheit** > **Globale Sicherheit**.
2. Setzen Sie **Verfügbare Realmdefinitionen** auf **Eingebundene Repositorys** und klicken Sie auf die Schaltfläche **Konfigurieren**.
3. Setzen Sie **Primärer administrativer Benutzername** auf `websphere`.
4. Wählen Sie das Optionsfeld **Automatisch generierte Server-ID**.
5. Wählen Sie **Groß-/Kleinschreibung für Berechtigung ignorieren** und klicken Sie auf **OK**.
6. Geben Sie das Kennwort für den Standardbenutzer mit Verwaltungsaufgaben, z.B. `websphere`, geben Sie die Bestätigung ein und klicken Sie auf **OK**, um die Änderungen zu bestätigen.
7. Setzen Sie **Verfügbare Realmdefinitionen** auf **Eingebundene Repositorys** und klicken Sie auf die Schaltfläche **Als aktuelles Repository festlegen**.
8. Wählen Sie **Verwaltungssicherheit aktivieren**.
9. Wählen Sie **Anwendungssicherheit aktivieren**.
10. Wählen Sie **Java-2-Sicherheit verwenden, um den Anwendungszugriff auf lokale Ressourcen zu beschränken** und **Warnen, wenn Anwendungen angepasste Berechtigungen erteilt werden**.
11. Klicken Sie auf die Schaltfläche **Anwenden**, um die Änderungen zu bestätigen.
12. Setzen Sie **Verfügbare Realmdefinitionen** auf **Eingebundene Repositorys**.
13. Klicken Sie auf die Schaltfläche **Anwenden**, um die Änderungen zu bestätigen.
14. Navigieren Sie zu **Sicherheit** > **Globale Sicherheit**.
15. Erweitern Sie **Web- und SIP-Sicherheit** und wählen Sie **Single Sign-on (SSO)** aus.
16. Wählen Sie **Erfordert SSL** aus.
17. Klicken Sie auf **OK**, um die Änderung zu bestätigen.
18. Navigieren Sie zu **Sicherheit** > **Globale Sicherheit**.
19. Wählen Sie den Link **Angepasste Eigenschaften** aus.

20. Klicken Sie auf die Schaltfläche **Neu** und setzen Sie Namen und Wert wie folgt:

    Name: `com.ibm.ws.security.web.logoutOnHTTPSessionExpire`

    Wert: `true`
21. Klicken Sie auf die Schaltfläche **OK**, um die neue Eigenschaft hinzuzufügen.
22. Klicken Sie auf die Schaltfläche **Neu** und setzen Sie Namen und Wert wie folgt:

    Name: `com.ibm.ws.security.addHttpOnlyAttributeToCookies`

    Wert: `true`
23. Klicken Sie auf **OK**, um die Änderung zu bestätigen.
24. Speichern Sie die Änderungen in der Masterkonfiguration.

Anwendungsserver erneut starten

Dieser Schritt ist obligatorisch. Der Server muss erneut gestartet werden, damit die Sicherheitsänderungen wirksam werden und zusätzliche erforderliche Benutzer hinzugefügt werden können. Der Server kann mithilfe der Datei `stopServer.bat` gestoppt werden, die sich im Verzeichnis `profiles/AppSrv01/bin` der WebSphere Application Server-Installation befindet. Dieser Vorgang ist ähnlich dem Starten des Servers wie unter „Einführung” auf Seite 18 beschrieben.

Vor dem Neustart der Anwendungsservers, z.B. 'server1', ist es notwendig, die Registrierungs- und die Kryptografie-JAR-Dateien für WebSphere Application Server verfügbar zu machen. Die Registrierungs-JAR-Datei enthält Klassen, die für die Sicherheitskonfiguration notwendig sind. Die Kryptografie-JAR-Datei enthält Konfigurationseinstellungen und -daten, die für die Kennwortsicherheit erforderlich sind.

Die Datei `Registry.jar` befindet sich im Verzeichnis `lib` der SDEJ-Installation. Kopieren Sie diese Datei in das `lib`-Verzeichnis der WebSphere Application Server-Installation.

Die Datei `CryptoConfig.jar` kann generiert werden, indem das Ant-Ziel 'configtest' wie folgt ausgeführt wird: *build configtest -Dcrypto.ext.dir=filedir*. Kopieren Sie die Datei 'CryptoConfig.jar' aus der generierten Position. Kopieren Sie diese Datei in das Verzeichnis Java `jre/lib/ext`. Wenn Sie Anpassungen an die Kryptografie-Konfiguration für Cúram benötigen, lesen Sie die entsprechenden Angaben im Dokument *Cúram Security Handbook*.

Starten Sie nun den Anwendungsserver, in unserem Beispiel 'server1', und öffnen Sie die Administrationskonsole, um mit den Konfigurationsschritten fortzufahren. Da die Sicherheitskonfiguration vollständig ist und die Scripting-Änderungen vorgenommen wurden, ist es nun möglich, die SDEJ-Scripts für den Neustart von WebSphere Application Server zu verwenden. Weitere Details zum Neustarten des Servers finden Sie unter „WebSphere Server starten und stoppen” auf Seite 14.

Um mit der Konfiguration fortzufahren, muss jetzt die Administrationskonsole geöffnet werden. Nachdem nun die globale Sicherheit aktiviert ist, müssen Sie sich mit dem zuvor eingerichteten Benutzernamen und Kennwort bei der Konsole anmelden.

Benutzer konfigurieren

Informationen zu diesem Vorgang

Wie unter „Sicherheitskonfiguration" auf Seite 10 ausführlich beschrieben, wird die konfigurierte WebSphere Application Server-Benutzerregistry für die Authentifizierung von Benutzern mit Verwaltungsaufgaben und Datenbankbenutzern verwendet. Die WebSphere-Benutzer mit Verwaltungsaufgaben und Datenbankbenutzer müssen der Benutzerregistry wie folgt manuell hinzugefügt werden:

Vorgehensweise

1. Navigieren Sie zu **Benutzer und Gruppen** > **Benutzer verwalten**.
2. Klicken Sie auf die Schaltfläche **Erstellen**.
3. Geben Sie die Einzeldaten für den WebSphere-Benutzer mit Verwaltungsaufgaben ein und klicken Sie auf die Schaltfläche **Erstellen**.
4. Wiederholen Sie diese Schritte für den Datenbankbenutzer.

Ergebnisse

Hinweis: Wenn die WebSphere-Verwaltungssicherheit während der Profilerstellung aktiviert war, ist der Benutzer mit Verwaltungsaufgaben möglicherweise bereits in der Registry definiert.

JAAS-Anmeldemodul für das System einrichten

Die Anwendungssicherheit verwendet für die Authentifizierung ein JAAS(Java Authentication and Authorization Service)-Anmeldemodul. Dieses Anmeldemodul muss für die Konfigurationen DEFAULT, WEB_INBOUND and RMI_INBOUND konfiguriert werden. Wiederholen Sie die unten aufgeführten Schritte für jede dieser Konfigurationen.

Anmeldemodul hinzufügen:
Vorgehensweise

1. Navigieren Sie zu **Sicherheit** > **Globale Sicherheit**.
2. Erweitern Sie im Abschnitt **Authentifizierung** den Eintrag **Java Authentication and Authorization Service** und wählen Sie **Systemanmeldungen** aus.
3. Wählen Sie den entsprechenden Alias aus der Liste aus. Das Anmeldemodul sollte für die Aliasse DEFAULT, WEB_INBOUND und RMI_INBOUND konfiguriert sein.
4. Klicken Sie auf **Neu**, um ein neues Anmeldemodul zu konfigurieren.
5. Setzen Sie das Feld **Name der Modulklasse** auf `curam.util.security.CuramLoginModule`.
6. Wählen Sie die Option **Proxy für Anmeldemodul verwenden** aus.
7. Wählen Sie im Feld **Authentifizierungsstrategie** REQUIRED aus.
8. Geben Sie in die Tabelle **Angepasste Eigenschaften** Namenwertepaare für alle unten aufgeführten erforderlichen Eigenschaften ein und klicken Sie je nach Bedarf auf die Schaltfläche **Neu**.

Tabelle 1. Angepasste 'CuramLoginModule'-Eigenschaften

Name	Beispielwert	Beschreibung
exclude_usernames	websphere, db2admin	Erforderlich. Eine Liste von Benutzernamen, die von der Authentifizierung ausgeschlossen werden sollen. Der Standardbegrenzer ist ein Komma, was aber mit dem Begrenzer 'exclude_usernames_delimiter' überschrieben werden kann. Diese Liste sollte den WebSphere-Benutzer mit Administrationsrechten (wie in „Verwaltungssicherheit konfigurieren" auf Seite 26 angegeben) und den Datenbankbenutzer (wie in „Datenquellen-Anmeldealias erstellen" auf Seite 21 angegeben) enthalten. Jeder hier aufgeführte Benutzer sollte in der WebSphere Application Server-Benutzerregistry definiert sein.
exclude_usernames_delimiter	|	*Optional*. Ein Begrenzungszeichen für die Benutzernamenliste, die in 'exclude_usernames' bereitgestellt wird. In Fällen, wo die Benutzernamen eingebettete Kommas enthalten, wie bei LDAP-Benutzern, kann es hilfreich sein, wenn der Begrenzer nicht das standardmäßige Komma ist.
login_trace	true	*Optional*. Diese Eigenschaft sollte auf 'true' gesetzt sein, um für den Authentifizierungsprozess den Debugger auszuführen. Ist sie auf 'true' gesetzt, bewirkt der Aufruf des Anmeldemoduls, dass Informationen aus der Traceerstellung zur Datei `SystemOut.log` von WebSphere Application Server hinzugefügt werden.
module_name	DEFAULT, WEB_INBOUND oder RMI_INBOUND	*Optional*. Diese Eigenschaft sollte auf DEFAULT, WEB_INBOUND oder RMI_INBOUND gesetzt sein, je nach der Konfiguration, für die das Anmeldemodul gerade definiert wird. Sie wird nur dann verwendet, wenn für Traceerstellungszwecke 'login_trace' auf 'true' gesetzt ist.
check_identity_only	true	*Optional*. Wenn diese Eigenschaft auf 'true' gesetzt ist, führt das Anmeldemodul nicht die gewöhnlichen Authentifizierungsverifizierungen durch. Stattdessen stellt es nur fest, ob der Benutzer in der Datenbanktabelle vorhanden ist. In diesem Fall wird die konfigurierte WebSphere Application Server-Benutzerregistry nicht umgangen, sondern nach dem Anmeldemodul abgefragt. Diese Option ist für Fälle gedacht, in denen LDAP-Unterstützung erforderlich ist oder ein alternativer Authentifizierungsmechanismus verwendet werden soll. **Anmerkung:** Wenn Sie 'Authentifizierung nur anhand der Identität' angeben und LDAP verwenden, müssen unter Umständen zusätzliche Konfigurationsschritte ausgeführt werden. Nähere Informationen dazu finden Sie unter „Besondere Konfigurationsschritte bei der Verwendung von 'Authentifizierung nur anhand der Identität' und LDAP" auf Seite 10.
user_registry_enabled	true	*Optional*. Diese Eigenschaft wird zum Überschreiben des Verhaltens verwendet, die Benutzerregistry zu umgehen. Ist sie auf 'true' gesetzt, so wird die Benutzerregistry während des Authentifizierungsprozesses abgefragt. Ist sie auf 'false' gesetzt, wird die WebSphere Application Server-Benutzerregistry nicht abgefragt.
user_registry_enabled_types	EXTERNAL	*Optional*. Diese Eigenschaft wird zum Angeben einer durch Kommas begrenzten Liste externer Benutzertypen verwendet, die anhand der WebSphere Application Server-Benutzerregistry (z.B. LDAP) verarbeitet werden soll. Unter „WebSphere Application Server-Benutzerregistry" auf Seite 11 finden Sie weitere Informationen zur Verarbeitung der WebSphere Application Server-Benutzerregistry.

Tabelle 1. Angepasste 'CuramLoginModule'-Eigenschaften (Forts.)

Name	Beispielwert	Beschreibung
user_registry_disabled_types	EXTGEN,EXTAUTO	*Optional*. Diese Eigenschaft wird zum Angeben einer durch Kommas begrenzten Liste externer Benutzertypen verwendet, die nicht anhand der WebSphere Application Server-Benutzerregistry (z.B. LDAP) verarbeitet werden soll. Unter „WebSphere Application Server-Benutzerregistry" auf Seite 11 finden Sie weitere Informationen zur Verarbeitung der WebSphere Application Server-Benutzerregistry.

9. Klicken Sie auf **OK**, um das Hinzufügen des neuen Anmeldemoduls zu bestätigen.

Anmeldemodul verschieben:
Vorgehensweise

1. Navigieren Sie zu **Sicherheit** > **Globale Sicherheit**.
2. Erweitern Sie im Abschnitt **Authentifizierung Java Authentication and Authorization Service** und wählen Sie **Systemanmeldungen**.
3. Wählen Sie den entsprechenden Alias aus der Liste aus. Das Anmeldemodul sollte wie folgt für die Aliasse DEFAULT, WEB_INBOUND und RMI_INBOUND verschoben werden:
4. Klicken Sie auf die Schaltfläche **Reihenfolge festlegen**.
5. Wählen Sie die Eigenschaft **curam.util.security.CuramLoginModule** aus und klicken Sie auf die Schaltfläche **Nach oben verschieben**. Wiederholen Sie diesen Vorgang, bis der Eintrag `CuramLoginModule` der oberste Eintrag in der Liste ist.
6. Klicken Sie auf **OK**, um die Änderungen an der Reihenfolge zu bestätigen.

Cross-Cluster-Authentifizierung inaktivieren:
Informationen zu diesem Vorgang

Diese Eigenschaft bestimmt das Verhalten einer LTPA-Token2-Anmeldung mit Single Sign-on. Die Eigenschaft 'com.ibm.ws.security.webChallengeIfCustomSubjectNotFound' ist auf 'false' (falsch) gesetzt, um sicherzustellen, dass Websitzungen nahtlos zwischen den zwei Servern eines Clusters übertragen werden können (z.B. in einem Übernahmeszenario), ohne dass der Benutzer zur Eingabe von Sicherheitsberechtigungsnachweisen aufgefordert wird.

Vorgehensweise

1. Navigieren Sie zu **Sicherheit** > **Globale Sicherheit**.
2. Klicken Sie auf **Angepasste Eigenschaften** und wählen Sie die Eigenschaft **com.ibm.ws.security.webChallengeIfCustomSubjectNotFound** aus der Liste verfügbarer Eigenschaften.
3. Ändern Sie unter 'General Properties' den Wert der Eigenschaft **com.ibm.ws.security.webChallengeIfCustomSubjectNotFound** in *false*.
4. Klicken Sie auf **OK**, um die Hinzufügung zu bestätigen.

Änderungen speichern: Speichern Sie die Änderungen in der Masterkonfiguration wie unter „Masterkonfiguration speichern" auf Seite 26 beschrieben.

Serverkonfiguration

Port für JNDI-Suche konfigurieren:

Vorgehensweise

1. Navigieren Sie zu **Server** > **Servertypen** > **WebSphere-Anwendungsserver**.
2. Wählen Sie den entsprechenden Server aus der Liste aus, z.B. 'server1'.
3. Erweitern Sie im Abschnitt **Kommunikation** die Option **Ports** und klicken Sie auf die Schaltfläche **Details**.
4. Wählen Sie den Eintrag **BOOTSTRAP_ADDRESS** aus und setzen Sie **Port** auf den Wert der Eigenschaft 'curam.server.port' in ihrer `AppServer.properties`-Datei.
5. Klicken Sie auf **OK**, damit die Änderungen wirksam werden.
6. Speichern Sie die Änderungen in der Masterkonfiguration mithilfe der Option **Speichern** wie zuvor.

Pass-by-Reference für Object-Request-Broker konfigurieren:
Vorgehensweise

1. Navigieren Sie zu **Server** > **Servertypen** > **WebSphere-Anwendungsserver**.
2. Wählen Sie den entsprechenden Server aus der Liste aus, z.B. 'server1'.
3. Erweitern Sie im Abschnitt **Containereinstellungen** die Option **Containerservices** und klicken Sie auf den Link **ORB-Service**.
4. Wählen Sie die Option **Durch Referenz übergeben** aus dem Abschnitt **Allgemeine Eigenschaften**.
5. Klicken Sie auf **OK**, damit die Änderungen wirksam werden.
6. Speichern Sie die Änderungen in der Masterkonfiguration mithilfe der Option **Speichern** wie zuvor.

Java Virtual Machine konfigurieren:
Vorgehensweise

1. Navigieren Sie zu **Server** > **Servertypen** > **WebSphere-Anwendungsserver**.
2. Wählen Sie den entsprechenden Server aus der Liste aus.
3. Erweitern Sie im Abschnitt **ServerinfrastrukturJava- und Prozessverwaltung and Process Management**.
4. Wählen Sie den Link **Prozessdefinition** aus.
5. Wählen Sie im Abschnitt **Weitere Eigenschaften** den Link **Java Virtual Machine** aus.
6. Nehmen Sie in den Feldern folgende Einstellungen vor:

   Anfangsgröße des Heapspeichers :`1024`

   Maximale Größe des Heapspeichers: `1024`

   Klicken Sie auf **Anwenden**, um die Werte festzulegen.
7. Im Abschnitt **Weitere Eigenschaften** wählen Sie den Link **Angepasste Eigenschaften** aus.
8. Klicken Sie auf **Neu** und definieren Sie dann die Eigenschaften wie folgt:

   Name: `com.ibm.websphere.security.util.authCacheCustomKeySupport`

   Wert: `false`

   Klicken Sie auf **OK**, um die Eigenschaft hinzuzufügen.
9. *Der folgende Schritt ist nur für Plattformen, die keine Windows-Plattformen sind, erforderlich.*

   Klicken Sie auf **Neu** und definieren Sie dann die Eigenschaften wie folgt:

   Name : `java.awt.headless`

   Wert : `true`

Klicken Sie auf **OK**, um die Eigenschaft hinzuzufügen.

10. Speichern Sie die Änderungen in der Masterkonfiguration mithilfe der Option **Speichern** wie zuvor.

Zeitgeberservice konfigurieren:
Vorgehensweise

1. Navigieren Sie zu **Server** > **Servertypen** > **WebSphere-Anwendungsserver**.
2. Wählen Sie den entsprechenden Server aus der Liste aus.
3. Erweitern Sie im Abschnitt **ContainereinstellungenEJB-Containereinstellungen**.
4. Wählen Sie den Link **Einstellungen des EJB-Zeitgeberservice** aus.
5. Wählen Sie im Fenster **Schedulertyp** die Option **Interne Scheduler-Instanz für EJB-Zeitgeberservice verwenden** aus.
6. Nehmen Sie in den Feldern folgende Einstellungen vor:

   JNDI-Name der Datenquelle: `jdbc/curamtimerdb`

   Datenquellenalias: *<valid for database>*

   Wobei der verwendete Alias der ist, der unter „Datenquellen-Anmeldealias erstellen" auf Seite 21 eingerichtet wird.
7. Klicken Sie auf **OK**, um die Änderungen zu bestätigen.
8. Speichern Sie die Änderungen in der Masterkonfiguration mithilfe der Option **Speichern** wie zuvor.

Portzugriff einrichten:
Vorgehensweise

1. Navigieren Sie zu **Server** > **Servertypen** > **WebSphere-Anwendungsserver**.
2. Wählen Sie den entsprechenden Server aus der Liste aus.
3. Wählen Sie im Abschnitt **Kommunikation** den Link **Ports** aus.
4. Klicken Sie auf die Schaltfläche **Details**.
5. Klicken Sie auf **Neu** und setzen Sie für den Client-TCP/IP-Port die folgenden Felder:

   Name des benutzerdefinierten Ports : `CuramClientEndPoint`

   Host : `*`

   Port : `9044`

   Klicken Sie auf **OK**, damit die Änderungen Anwendung finden.
6. Klicken Sie auf **Neu** und setzen Sie für den Web-Services-TCP/IP-Port die folgenden Felder:

   Name des benutzerdefinierten Ports : `CuramWebServicesEndPoint`

   Host : `*`

   Port : `9082`

   Klicken Sie auf **OK**, damit die Änderungen Anwendung finden.
7. Navigieren Sie zu **Server** > **Servertypen** > **WebSphere-Anwendungsserver**.
8. Wählen Sie den entsprechenden Server aus der Liste aus.
9. Erweitern Sie in der Verzweigung **Einstellungen des Webcontainers** den Abschnitt **Containereinstellungen**.
10. Wählen Sie den Link **Transportketten für Webcontainer** aus.
11. Klicken Sie auf **Neu** und setzen Sie für die Client-Transportketten die folgenden Felder:

    Name : `CuramClientChain`

Transportkettenschablone : WebContainer-Secure

Klicken Sie auf **Weiter**.

Vorhandenen Port verwenden : CuramClientEndPoint

Klicken Sie auf **Weiter** und anschließend auf **Fertigstellen**.

12. Klicken Sie auf **Neu** und setzen Sie für die WebServices-Transportkette die folgenden Felder:

    Name : `CuramWebServicesChain`

    Transportkettenschablone : WebContainer

    Klicken Sie auf **Weiter**.

    Vorhandenen Port verwenden : CuramWebServicesEndPoint

    Klicken Sie auf **Weiter** und anschließend auf **Fertigstellen**.
13. Wählen Sie die neu erstellte **CuramClientChain** aus.
14. Wählen Sie den Link **HTTP Inbound Channel** aus.
15. Stellen Sie sicher, dass das Kontrollkästchen **Persistente (Keep-Alive-) Verbindungen verwenden** aktiviert ist.
16. Klicken Sie auf **OK**, um die Hinzufügung zu bestätigen.
17. Navigieren Sie zu **Umgebung** > **Virtuelle Hosts**.
18. Klicken Sie auf **Neu**, um durch Setzen der folgenden Felder `Virtueller Host` neu hinzuzufügen.

    Name = *client_host*

    Wiederholen Sie diesen Schritt und ersetzen Sie dabei *client_host* durch *webservices_host*.
19. Wählen Sie den Link **client_host** aus der Liste virtueller Hosts aus.

    Wählen Sie den Link **Hostalias** im Abschnitt **Weitere Eigenschaften**.

    Klicken Sie auf **Neu**, um durch Setzen der folgenden Felder einen neuen `Alias` hinzuzufügen.

    Hostname = *

    Port = *9044*

    Wobei *9044* der Port ist, der in Schritt 5 verwendet wurde. Wiederholen Sie diesen Schritt für den anderen verwendeten virtuellen Host und Port (z.B. webservices_host, 9082).
20. Klicken Sie auf **OK**, um die Hinzufügung zu bestätigen.
21. Speichern Sie die Änderungen in der Masterkonfiguration wie unter „Masterkonfiguration speichern" auf Seite 26 beschrieben.

Sicherheitsintegration für Sitzungen konfigurieren: Vorgehensweise

1. Navigieren Sie zu **Server** > **Servertypen** > **WebSphere-Anwendungsserver**.
2. Wählen Sie den entsprechenden Server aus der Liste aus.
3. Klicken Sie im Abschnitt **Containereinstellungen** auf **Sitzungsverwaltung**.
4. Inaktivieren Sie die Option **Sicherheitsintegration**. *Hinweis: Achten Sie darauf, dass die Sicherheitsintegration inaktiviert ist.*
5. Klicken Sie auf **OK**, damit die Änderungen wirksam werden.
6. Speichern Sie die Änderungen in der Masterkonfiguration mithilfe der Option **Speichern** wie zuvor.

Anmerkung:

Die obige Einstellung ist für IBM Cúram Social Program Management-Webanwendungen erforderlich.

Buskonfiguration

Service Integration Bus einrichten:
Vorgehensweise

1. Navigieren Sie zu **Serviceintegration** > **Busse**.
2. Klicken Sie auf **Neu** setzen Sie in **Schritt 1** das folgende Feld:

   Name: CuramBus

   Belassen Sie alle anderen Werte als Standardwerte und klicken Sie auf **Weiter**.
3. Rufen Sie den Assistenten **Bussicherheit konfigurieren**, Schritt 1.1, auf und klicken Sie auf **Weiter**.

   In **Schritt 1.2** des Assistenten **Bussicherheit konfigurieren** übernehmen Sie die Standardeinstellung und klicken Sie auf **Weiter**.

   In **Schritt 1.3** des Assistenten **Bussicherheit konfigurieren** übernehmen Sie die Standardeinstellungen, sofern sie geeignet sind, und klicken Sie auf **Weiter**.

   In **Schrittt 1.4** des Assistenten **Bussicherheit konfigurieren** überprüfen Sie Ihre Einstellungen und klicken Sie auf **Weiter**.
4. In Schritt 2 klicken Sie auf **Fertigstellen**, damit die Änderungen wirksam werden.
5. Wählen Sie den **CuramBus** aus der Liste mit Bussen aus, die jetzt angezeigt wird. Damit öffnet sich die Konfigurationsanzeige.
6. Wählen Sie **Bus-Member** im Abschnitt **Topologie** aus.
7. Klicken Sie auf **Hinzufügen**, woraufhin der Assistent **Neues Bus-Member hinzufügen** geöffnet wird.
8. Wählen Sie den Server aus, der dem Bus hinzugefügt werden soll, und klicken Sie auf **Weiter**.
9. Wählen Sie **Datenspeicher** aus und klicken Sie auf **Weiter**.
10. Wählen Sie die Option **Vorhandene Datenquelle verwenden** aus und setzen Sie die Optionen wie folgt:

    JNDI-Name der Datenquelle = jdbc/curamsibdb

    Schemaname = *username*

    Wobei *username* der Benutzername für die Datenbank ist.

    Heben Sie die Auswahl der Option **Tabellen erstellen** auf.

    Belassen Sie alle anderen Werte als Standardwerte und klicken Sie auf **Weiter**.
11. Übernehmen Sie die Standard-Optimierungsparameter als angemessene Werte und klicken Sie auf **Weiter**.
12. Klicken Sie auf **Fertigstellen**, um den Vorgang zu beenden und den Assistenten zu verlassen.
13. Navigieren Sie zu **Serviceintegration** > **Busse**.
14. Wählen Sie den **CuramBus** aus der Liste mit Bussen aus, die jetzt angezeigt wird. Damit öffnet sich die Konfigurationsanzeige.
15. Wählen Sie im Abschnitt **Weitere Eigenschaften Sicherheit** aus.
16. Wählen Sie im Abschnitt **Berechtigungsrichtlinie Benutzer und Gruppen in der Rolle Bus-Connector** aus.
17. Klicken Sie auf **Neu**, um den **Assistent für SIB-Sicherheitsressourcen** zu öffnen.

18. Wählen Sie das Optionsfeld **Integrierte Sondergruppen** aus und klicken Sie auf **Weiter**.
19. Aktivieren Sie die Kontrollkästchen **Server** und **AllAuthenticated** und klicken Sie auf **Weiter**.
20. Klicken Sie auf **Fertigstellen**, um den Vorgang zu beenden und den Assistenten zu verlassen.
21. Speichern Sie die Änderungen in der Masterkonfiguration wie unter „Masterkonfiguration speichern" auf Seite 26 beschrieben.

JMS(Java Message Service)-Konfiguration

JMS-Verbindungsfactorys einrichten:
Vorgehensweise

1. Navigieren Sie zu **Ressourcen** > **JMS** > **JMS-Provider**.
2. *Hinweis:* An dieser Stelle sollte der entsprechende Bereich ausgewählt werden, in dem die JMS-Ressourcen definiert werden sollen.
3. Wählen Sie den Link **Standard-Messaging-Provider** aus.
4. Wählen Sie im Abschnitt **Weitere Eigenschaften** den Link **Verbindungsfactorys** aus.
5. Klicken Sie auf **Neu** und setzen Sie die folgenden Felder:

   Name: CuramQueueConnectionFactory

   JNDI-Name: jms/CuramQueueConnectionFactory

   Beschreibung : Die Factory für alle Verbindungen zu Anwendungswarteschlangen.

   Busname: CuramBus

   Authentifizierungsalias für XA-Wiederherstellung: Gleicher Wert wie für die Datenquelle `jdbc/curamdb` (z.B. <*SERVERNAME*> /dbadmin)

   Alias für Konfigurationszuordnung: DefaultPrinicipalMapping

   Containergesteuerter Authentifizierungsalias: Gleicher Wert wie für den Authentifizierungsalias für die XA-Wiederherstellung.

   Belassen Sie alle anderen Werte als Standardwerte und klicken Sie auf **OK**, damit die Änderungen Anwendung finden.
6. Klicken Sie auf **Neu** und setzen Sie die folgenden Felder:

   Name: CuramTopicConnectionFactory

   JNDI-Name: jms/CuramTopicConnectionFactory

   Beschreibung : Die Factory für alle Verbindungen zu Anwendungswarteschlangen.

   Busname: CuramBus

   Authentifizierungsalias für XA-Wiederherstellung: Gleicher Wert wie für die Datenquelle `jdbc/curamdb` (z.B. <*SERVERNAME*> /dbadmin)

   Alias für Konfigurationszuordnung: DefaultPrinicipalMapping

   Containergesteuerter Authentifizierungsalias: Gleicher Wert wie für die Datenquelle `jdbc/curamdb` (z.B. <*SERVERNAME*> /dbadmin)

   Belassen Sie alle anderen Werte als Standardwerte und klicken Sie auf **OK**, damit die Änderungen Anwendung finden.
7. Speichern Sie die Änderungen in der Masterkonfiguration wie unter „Masterkonfiguration speichern" auf Seite 26 beschrieben.

Ergebnisse

Anmerkung: Mit den obigen manuellen Konfigurationsschritten ist es nicht möglich, die Sicherheit für die Warteschlange und die Topic-Verbindungsfactorys ordnungsgemäß zu konfigurieren. Für diesen Teil der Konfiguration müssen Sie das Tool `wsadmin` verwenden. Verlassen Sie dazu die Administrationskonsole und befolgen Sie diese Schritte:

1. Geben Sie die Einträge für die Warteschlange und die Topic-Verbindungsfactory in der Datei `resources.xml` für die Konfiguration von WebSphere Application Server ein. Diese Datei befindet sich in der Dateisystemhierarchie `%WAS_HOME%\profiles\<profile_name>\config`. Ihr Ort ist von Ihren Namenskonventionen und dem Bereich abhängig, in dem Sie Ihre JMS-Ressourcen definiert haben. Beispielsweise würde sich die Datei bei einem Knotenebenenbereich mit dem Profilnamen `AppSrv01`, dem Zellennamen `MyNodeCell` und dem Knotennamen `MyNode` an folgendem Ort befinden: `C:\WebSphere\profiles\AppSrv01\config\cells\MyNodeCell\nodes\MyNode\resources.xml`. In dieser Datei müssen Sie die `<factories>`-Entitäten für `CuramQueueConnectionFactory` und `CuramTopicConnectionFactory` suchen und die ID für jede von ihnen vermerken, die mit `J2CConnectionFactory_` beginnt und von einem numerischen Wert gefolgt wird, z.B. 1264085551611.
2. Rufen Sie das WebSphere-Script `wsadmin` auf. In diesen Beispielen ist die Sprache Jacl, so dass das *-lang jacl*-Argument möglicherweise mit Berechtigungsnachweisen für Anmeldung o.ä., je nach Ihrer lokalen Konfiguration, angegeben werden muss.
3. In `wsadmin` rufen Sie die folgenden Befehle auf, wieder mit den angenommen Definitionen auf Knotenebene, dem Zellennamen `MyNodeCell` und dem Knotennamen `MyNode`. Die Ressourcen-IDs werden in Ihrer Umgebung andere sein.
   a. Rufen Sie den Knoten und die Zellkennung auf: `$AdminConfig getid /Node:MyNode`
   b. Kombinieren Sie den Knoten und die Zellkennung aus dem vorherigen Schritt mit der Kennung der Verbindungsfactory, die Sie im obigen Schritt erhalten haben, um die Verbindungsfactory anzuzeigen: `$AdminTask showSIBJMSConnectionFactory CuramQueueConnectionFactory(cells/MyNodeCell/nodes/MyNode|resources.xml#J2CConnectionFactory_1264085551611)`

      Anhand der obigen Befehlsausgabe sollten Sie verifizieren, dass 'authDataAlias' nicht gesetzt ist (z. B. `authDataAlias=`), sonst bekommen Sie Probleme, wie folgendes Beispiel für eine wsadmin-Ausgabe zeigt:

```
{password=, logMissingTransactionContext=false,
readAhead=Default, providerEndpoints=,
shareDurableSubscriptions=InCluster,
targetTransportChain=, authDataAlias=, userName=,
targetSignificance=Preferred,
shareDataSourceWithCMP=false,
nonPersistentMapping=ExpressNonPersistent,
persistentMapping=ReliablePersistent, clientID=,
jndiName=jms/CuramQueueConnectionFactory,
manageCachedHandles=false,
consumerDoesNotModifyPayloadAfterGet=false,
category=, targetType=BusMember, busName=CuramBus,
description=None,
xaRecoveryAuthAlias=crouch/databaseAlias,
temporaryTopicNamePrefix=, remoteProtocol=,
producerDoesNotModifyPayloadAfterSet=false,
connectionProximity=Bus, target=,
temporaryQueueNamePrefix=,
name=CuramQueueConnectionFactory}
```

c. Um den 'authDataAlias' so einzurichten, dass er dieselben Informationen zur Verbindungsfactory verwendet wie oben, gehen Sie beispielsweise folgendermaßen vor: `$AdminTask modifySIBJMSConnectionFactory CuramQueueConnectionFactory(cells/MyNodeCell/nodes/MyNode|resources.xml#J2CConnectionFactory_1264085551611) {-authDataAlias crouch/databaseAlias}`

d. Speichern Sie die Änderungen: `$AdminConfig save`

e. Sie können den Befehl `showSIBJMSConnectionFactory` aufrufen, um die Änderung zu überprüfen.

f. Wiederholen Sie diese Schritte für `CuramTopicConnectionFactory`.

g. Starten Sie den Anwendungsserver erneut.

Erforderliche Warteschlangen einrichten:
Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte durch, indem Sie <*QueueName*> (ohne spitze Klammern) durch jeden der folgenden Warteschlangennamen ersetzen: DPEnactment, DPError, CuramDeadMessageQueue, WorkflowActivity, WorkflowEnactment und WorkflowError.

Vorgehensweise

1. Navigieren Sie zu **Serviceintegration** > **Busse** > **CuramBus**.
2. Wählen Sie im Abschnitt **Zielressourcen** den Link **Ziele** aus.
3. Klicken Sie auf **Neu**, um den Assistenten „Neues Ziel erstellen" zu öffnen.
4. Wählen Sie **Warteschlange** als Zieltyp aus und klicken Sie auf **Weiter**.
5. Setzen Sie die folgenden Warteschlangenattribute:

   Kennung: SIB_ <*QueueName*>

   Belassen Sie alle anderen Werte als Standardwerte und klicken Sie auf **Weiter**.
6. Verwenden Sie **Ausgewähltes Bus-Member** und klicken Sie auf **Weiter**.
7. Klicken Sie auf **Fertigstellen**, um die Erstellung der Warteschlange zu bestätigen.
8. Wählen Sie die neu hinzugefügte Warteschlange SIB_ <*QueueName*> aus, die jetzt in der Liste vorhandener Anbieter angezeigt wird. Damit öffnet sich wieder die Konfigurationsanzeige.

9. Verwenden Sie die folgende Tabelle, um über das Optionsfeld **Angeben** und das zugehörige Textfeld das Ausnahmeziel festzulegen.

Tabelle 2. Einstellungen für Ausnahmeziele

Name der Warteschlange	Ausnahmeziel
SIB_CuramDeadMessageQueue	System
SIB_DPEnactment	SIB_DPError
SIB_DPError	SIB_CuramDeadMessageQueue
SIB_WorkflowActivity	SIB_WorkflowError
SIB_WorkflowEnactment	SIB_WorkflowError
SIB_WorkflowError	SIB_CuramDeadMessageQueue

10. Klicken Sie auf **OK**, damit die Änderungen wirksam werden.
11. Navigieren Sie zu **Ressourcen** > **JMS** > **JMS-Provider**.
12. Wählen Sie den Link **Standard-Messaging-Provider** aus.
13. Wählen Sie im Abschnitt **Weitere Eigenschaften** den Link **Warteschlangen** aus.
14. Klicken Sie auf **Neu** und setzen Sie die folgenden Felder:

    Name: <*QueueName*>

    JNDI-Name: jms/ <*QueueName*>

    Busname: CuramBus

    Name der Warteschlange: SIB_ <*QueueName*>

    Übermittlungsmodus: Persistent

    Belassen Sie alle anderen Werte als Standardwerte und klicken Sie auf **OK**, damit die Änderungen Anwendung finden.

Ergebnisse

Speichern Sie die Änderungen in der Masterkonfiguration wie unter „Masterkonfiguration speichern" auf Seite 26 beschrieben.

Erforderliche Themen einrichten:
Vorgehensweise

1. Navigieren Sie zu **Ressourcen** > **JMS** > **JMS-Provider**.
2. Wählen Sie den Link **Standard-Messaging-Provider** aus.
3. Wählen Sie im Abschnitt **Weitere Eigenschaften** den Link **Thema** aus.
4. Klicken Sie auf **Neu** und setzen Sie die folgenden Felder:

    Name: CuramCacheInvalidationTopic

    JNDI-Name des Ziels: jms/CuramCacheInvalidationTopic

    Beschreibung: Cache Invalidation Topic

    Busname: CuramBus

    Topicbereich: Default.Topic.Space

    JMS-Übermittlungsmodus : Persistent

    Belassen Sie alle anderen Werte als Standardwerte und klicken Sie auf **OK**, damit die Änderungen Anwendung finden.
5. Speichern Sie die Änderungen in der Masterkonfiguration wie unter „Masterkonfiguration speichern" auf Seite 26 beschrieben.

Erforderliche Warteschlangen-Aktivierungsspezifikationen einrichten:

Informationen zu diesem Vorgang

Führen Sie wie beim Einrichten von Warteschlangen diese Schritte durch, indem Sie <*QueueName*> (ohne spitze Klammern) durch jeden der folgenden Warteschlangennamen ersetzen: DPEnactment, DPError, CuramDeadMessageQueue, WorkflowActivity, WorkflowEnactment und WorkflowError.

Vorgehensweise

1. Navigieren Sie zu **Ressourcen** > **JMS** > **JMS-Provider**.
2. Wählen Sie den Link **Standard-Messaging-Provider** aus.
3. Wählen Sie im Abschnitt **Weitere Eigenschaften** den Link **Aktivierungsspezifikationen** aus.
4. Erstellen Sie eine neue Spezifikation, indem Sie auf **Neu** klicken und die folgenden Felder setzen:

   Name: <QueueName>

   JNDI-Name: eis/ <*QueueName*> AS

   Zieltyp: Queue

   JNDI-Name des Ziels: jms/ <*QueueName*>

   Busname: CuramBus

   Authentifizierungsalias: Derselbe wie für die Datenquelle `jdbc/curamdb` (z.B. <*SERVERNAME*> /dbadmin)

   Belassen Sie alle anderen Werte als Standardwerte und klicken Sie auf **OK**, um den Port hinzuzufügen.

Ergebnisse

Speichern Sie die Änderungen in der Masterkonfiguration wie unter „Masterkonfiguration speichern" auf Seite 26 beschrieben.

Erforderliche Themen-Aktivierungsspezifikationen einrichten: Vorgehensweise

1. Fügen Sie wie bei den Aktivierungsspezifikationen für Warteschlangen im vorigen Abschnitt eine neue Aktivierungsspezifikation hinzu und setzen Sie die folgenden Felder:

   Name: CuramCacheInvalidationTopic

   JNDI-Name: eis/CuramCacheInvalidationTopicAS

   Zieltyp: Topic

   JNDI-Name des Ziels: jms/CuramCacheInvalidationTopic

   Busname: CuramBus

   Authentifizierungsalias: Derselbe wie für die Datenquelle `jdbc/curamdb` (z.B. <*SERVERNAME*> /dbadmin)
2. Belassen Sie alle anderen Werte als Standardwerte und klicken Sie auf **OK**, damit die Änderungen Anwendung finden.
3. Speichern Sie die Änderungen in der Masterkonfiguration wie unter „Masterkonfiguration speichern" auf Seite 26 beschrieben.

Aufzubewahrende Protokolldateien konfigurieren

Es besteht die Möglichkeit, die maximale Anzahl an aufzubewahrenden Protokolldateien zu konfigurieren, die von einem bestimmten Server verwaltet werden sollen. Gehen Sie dazu wie folgt vor:

1. Navigieren Sie zu **Server** > **Servertypen** > **WebSphere-Anwendungsserver**.

2. Wählen Sie den entsprechenden Server aus der Liste von Servern.
3. Wählen Sie **Protokollierung und Traceerstellung** aus dem Abschnitt **Fehlerbehebung**.
4. Wählen Sie **JVM-Protokolle** aus der Liste **Allgemeine Eigenschaften**.
5. Ändern Sie den Wert im Feld **Maximale Anzahl aufzubewahrender Protokolldateien** in '30', sowohl für die Datei `System.out` als auch für die Datei `System.err`.
6. Klicken Sie auf **OK**, damit die Änderungen Anwendung finden.
7. Speichern Sie die Änderungen in der Masterkonfiguration.

Nach dem Konfigurieren

Datenbanktabellen für den Service Integration Bus: Nach der Installation müssen Datenbanktabellen manuell erstellt werden, die für den Service Integration Bus erforderlich sind. Mit dem SIB-DDL-Generator stellt WebSphere Application Server ein Dienstprogramm bereit, mit dessen Hilfe die SQL zur Erstellung dieser Tabellen generiert werden kann.

Der Generator wird mithilfe des folgenden Befehls (z. B. für Windows) ausgeführt:

```
%WAS_HOME%\bin\sibDDLGenerator.bat
  -system system
  -platform platform
  -schema username
  -database database_name
  -user username
  -statementend ;
  -create
```

Wobei gilt:

- *system* ist die zu verwendende Datenbank, z.B. `oracle` oder `db2`,
- *platform* ist das Betriebssystem, z.B. `windows`, `unix` oder `zos`,
- *username* ist der Benutzername, der für den Zugriff auf die Datenbank erforderlich ist, wie in der `Bootstrap.properties`-Eigenschaft `curam.db.username` angegeben,
- *database_name* ist der Name der zu verwendenden Datenbank, wie in der `Bootstrap.properties`-Eigenschaft `curam.db.name` angegeben.

Beispiel:

```
c:/Websphere/AppServer/bin/sibDDLGenerator.bat
            -system db2 -platform windows
            -schema db2admin -database curam -user db2admin
            -statementend ; -create
```

Dies ist der Befehl zur Ausgabe einiger SQL-Anweisungen, die anschließend für die Zieldatenbank ausgeführt werden sollen.

Datenbanktabellen für den Zeitgeberservice: Nach der Installation müssen Datenbanktabellen manuell erstellt werden, die für den Zeitgeberservice erforderlich sind. WebSphere Application Server stellt in seinem Verzeichnis `WAS_HOME /Scheduler` die DDL für diese Tabellen zur Verfügung.

Die auszuführenden DDL-Dateien sind *createTablespaceXXX.ddl* und *createSchemaXXX.ddl* in dieser Reihenfolge, wobei *XXX* der Produktname Ihrer Zieldatenbank ist.

Jede DDL-Datei enthält Anweisungen entsprechend der Ausführung für Ihre Zieldatenbank.

Fertigstellung

Der Anwendungsserver ist nun konfiguriert und steht für die Installation der IBM Cúram Social Program Management-Anwendung bereit. Melden Sie sich von der Administrationskonsole ab und starten Sie WebSphere Application Server mithilfe der Zielbeschreibung unter „WebSphere Server starten und stoppen" auf Seite 14 erneut.

Manuelle Anwendungsimplementierung

Für das Installieren einer Unternehmensanwendung in WebSphere Application Server kann die Administrationskonsole verwendet werden. Mit den unten aufgeführten Schritten wird beschrieben, wie mithilfe der Administrationskonsole eine Anwendung, EJB-Komponente oder ein Webmodul installiert werden kann.

Anmerkung: Ist die Installation einmal gestartet, so muss zum Abbrechen der Installation der Anwendung die Schaltfläche **Abbrechen** verwendet werden. Es reicht nicht aus, einfach auf eine andere Seite der Administrationskonsole zu wechseln, ohne zuerst auf einer Anwendungsinstallationsseite auf **Abbrechen** geklickt zu haben.

1. Navigieren Sie zu **Anwendungen** > **Neue Anwendung**.
2. Wählen Sie die Option **Neue Unternehmensanwendung** aus.
3. Klicken Sie auf das entsprechende Optionsfeld und geben Sie wahlweise über die Schaltfläche **Durchsuchen** oder das Fenster **Pfad der neuen Anwendung** den vollständigen Pfadnamen der Quellenanwendungsdatei oder EAR-Datei ein. Klicken Sie anschließend auf **Weiter**.

   Die Standardposition für die Anwendungs-EAR-Dateien ist

   `%SERVER_DIR%/build/ear/WAS/Curam.ear`
4. Wählen Sie im Fenster **Wie soll die Anwendung installiert werden?** das Optionsfeld **Schnell - Nur anfragen, wenn weitere Informationen erforderlich sind** aus. Klicken Sie auf **Weiter**.
5. Belassen Sie die Standardwerte, da sie für Schritt 1 gedacht sind, wählen Sie *Installationsoptionen* aus und klicken Sie auf **Weiter**.
6. In Schritt 2, **Servern Module zuordnen**, wählen Sie für jedes aufgeführte Modul aus der Liste **Cluster und Server** einen Zielserver oder -Cluster aus. Aktivieren Sie dazu das Kontrollkästchen neben dem jeweiligen Modul, wählen Sie den Server oder Cluster aus und klicken Sie auf **Anwenden**.
7. Um den folgenden Schritt oder die folgenden Schritte durchführen zu können, klicken Sie auf **Weiter** und anschließend auf **Fertigstellen**, um die Installation abzuschließen. Dieser Schritt kann einige Minuten in Anspruch nehmen. Anschließend sollte die Nachricht *Cúram-Anwendung erfolgreich installiert* erscheinen.
8. Speichern Sie die Änderungen in der Masterkonfiguration. (Weitere Details hierzu finden Sie unter „Masterkonfiguration speichern" auf Seite 26.)
9. Navigieren Sie zu **Anwendungen** > **Anwendungstypen** > **WebSphere-Unternehmensanwendungen** und wählen Sie die neu installierte Anwendung aus.
10. Wählen Sie im Abschnitt **Detaileigenschaften** die Option **Laden von Klassen und Erkennung von Dateiaktualisierungen** aus.
11. Setzen Sie die Eigenschaft **Reihenfolge der Klassenlader** auf **Mit dem lokalen Klassenlader geladene Klassen zuerst (übergeordneter zuletzt)**.

12. Setzen Sie die Eigenschaft **Klassenladerrichtlinie für WAR-Dateien** auf **Einzelner Klassenlader für gesamte Anwendung**.
13. Klicken Sie auf **OK**.
14. Navigieren Sie zu **Benutzer und Gruppen -> Benutzer verwalten**. Klicken Sie auf **Erstellen...** und geben Sie Benutzer-ID, Kennwort, Vornamen und Nachnamen ein. Klicken Sie anschließend auf **Erstellen**.

    Weitere Informationen bezüglich der Identifikationsdaten, die an dieser Stelle von der Anwendung erwartet werden, und ihrer Änderung finden Sie unter „SYSTEM-Benutzernamen ändern" auf Seite 16.
15. Gehen Sie zurück zur Unternehmensanwendung (**Anwendungen** > **Anwendungstypen** > **WebSphere-Unternehmensanwendungen**, wählen Sie die neu installierte Anwendung), wählen Sie aus dem Abschnitt **Detaileigenschaften** die Option **Zuordnung von Sicherheitsrollen zu Benutzern/Gruppen** aus und ordnen Sie mithilfe der folgenden Schritte die MDB-Benutzerrolle einem Benutzernamen und Kennwort zu:

    Anmerkung: Der Benutzername, den Sie für die Zuordnung zur MDB-Benutzerrolle verwenden, muss in Ihrer Benutzerregistry bereits definiert sein.
    a. Wählen Sie **Auswählen** für die MDB-Benutzerrolle und klicken Sie auf **Benutzer zuordnen...**.
    b. Geben Sie den entsprechenden Benutzernamen in das Feld **Suchbegriff** ein und klicken Sie auf **Suchen**.
    c. Wählen Sie die ID aus der Liste **Verfügbar:** und klicken Sie auf **>>**, um es zur Liste **Ausgewählt:** hinzuzufügen. Klicken Sie anschließend auf **OK**.
    d. Klicken Sie auf **OK**.
16. Nachdem die MDB-Benutzerrolle zugeordnet ist, kann nun die Benutzer-RunAs-Rolle aktualisiert werden. Wählen Sie dazu im Abschnitt **Detaileigenschaften** die Option **RunAs-Rollen für Benutzer** aus.
17. Geben Sie in die Felder **Benutzername** und **Kennwort** jeweils einen entsprechenden Benutzernamen und ein Kennwort ein. Wählen Sie **Auswählen** für die MDB-Benutzerrolle und klicken Sie auf **Anwenden**.
18. Klicken Sie auf **OK**.
19. Speichern Sie die Änderungen in der Masterkonfiguration.
20. Nach der Implementierung ist ein Start der Anwendung notwendig, bevor sie verwendet werden kann. Navigieren Sie zu **Anwendungen** > **Anwendungstypen** > **WebSphere-Unternehmensanwendungen**, aktivieren Sie das Kontrollkästchen für die neu installierte Anwendung und klicken Sie auf die Schaltfläche **Start**. Dieser Schritt kann einige Minuten in Anspruch nehmen. Anschließend ändert sich der Anwendungsstatus, wodurch angezeigt wird, dass sie jetzt gestartet ist.
21. Testen Sie als letzten Schritt die Anwendungsimplementierung. Rufen Sie beispielsweise mit einem Web-Browser die URL der implementierten Anwendung auf, z. B. `https://localhost:9044/Curam`.

WebSphere Network Deployment

IBM WebSphere Application Server Network Deployment bietet hochwertige Implementierungsservices, darunter Clustering, Ersterkennungsservices und Hochverfügbarkeit für verteilte Konfigurationen. Für weitere Informationen zur Installation von WebSphere Network Deployment sollte das Dokument *Cúram Third Party Tools Installation Guide* (Cúram-Handbuch für die Installation von Tools von Fremdanbietern) zu Rate gezogen werden.

Profile erstellen

Nach dem Installieren von WebSphere Network Deployment ist es in den meisten Fällen notwendig, mindestens zwei Profile zu erstellen. Eines agiert als Deployment Manager für den Knoten, das andere oder die anderen als die eingebundenen Server.

Dies geschieht mithilfe des Assistenten für die Profilerstellung, der über die Datei `pct<hardware platform>` aus dem Verzeichnis `bin/ProfileCreator` der WebSphere Application Server-Installation gestartet wird.

Die erste wichtige Auswahl, die bei diesem Assistenten getroffen werden muss, betrifft die Frage, welches der folgenden Elemente erstellt werden soll:

1. Ein Deployment Manager-Profil oder
2. ein Anwendungsserverprofil.

Mit dem letzteren wählt man die Aktivierung der Verwaltungssicherheit aus. Es wird empfohlen, die Verwaltungssicherheit bei der Profilerstellung zu aktivieren. Diese Einstellungen können nachträglich geändert werden.

Knoten einbinden

Für das Einbinden eines Anwendungsserverprofils ist das Starten des Deployment Manager als Ziel erforderlich.

Der Deployment Manager kann durch Ausführen des folgenden Befehls aus dem Verzeichnis `profiles/<deployment manager profile name>/bin` der WebSphere Network Deployment-Installation gestartet werden:

```
startServer dmgr
```

Um Ihr Anwendungsserverprofil zum Knoten des Deployment Manager hinzuzufügen, verwenden Sie den folgenden Befehl aus dem Verzeichnis `profiles/<Application Server profile name>/bin` der WebSphere Application Server-Installation:

```
addNode <deploymgr host> <deploymgr port>
```

Wobei der *<deploymgr host>* und der *<deploymgr port>* den empfangsbereiten Host und Port für den SOAP-Connector des Deployment Manager darstellen. Details zum SOAP-Connector finden Sie in der Deployment Manager-Administrationskonsole wie folgt:

1. Navigieren Sie zu **Server** > **Servertypen** > **WebSphere-Anwendungsserver**.
2. Wählen Sie den entsprechenden Server aus der Liste aus.
3. Erweitern Sie **Ports** im Abschnitt **Kommunikation** und klicken Sie auf die Schaltfläche **Details**.
4. Die erforderlichen Details werden als **Host** und **Port** für die **SOAP_CONNECTOR_ADDRESS** aufgelistet.

Knotenkonfiguration

Vor dem Implementieren einer Anwendung auf dem registrierten Knoten muss zunächst der Server konfiguriert werden. Dies geschieht mithilfe der Deployment Manager-Administrationskonsole. Die Konfiguration wird daraufhin mit den eingebundenen Servern des Knotens synchronisiert.

Der Knotenagent, der die Kommunikation zwischen dem Deployment Manager und seinen eingebundenen Servern ermöglichen soll, muss gestartet werden. Dies erfolgt über die Befehle `startNode.bat` oder `startNode.sh` im Verzeichnis `profiles/<federated profile name>/bin` der WebSphere Application Server-Installation.

Nach dem Starten des Knotenagenten wird die Steuerung der Server dieses Knotens vollständig an den Deployment Manager übergeben. Um einen Server in der Deployment Manager-Administrationskonsole zu starten oder zu stoppen, gehen Sie wie folgt vor:

1. Navigieren Sie zu **Server** > **Servertypen** > **WebSphere-Anwendungsserver**.
2. Heben Sie die Auswahl des Servers, der gestartet bzw. gestoppt werden soll, in der Liste auf und klicken Sie je nach Erforderlichkeit auf die Schaltfläche **Start** bzw. **Stoppen**.

Der nächste Prozessschritt besteht in der Konfiguration der eingebundenen Server. Wie schon erwähnt, erfolgt die gesamte Konfiguration über die Deployment Manager-Administrationskonsole. Unter „WebSphere Application Server manuell konfigurieren" auf Seite 19 finden Sie eine Beschreibung der manuellen WebSphere Application Server-Konfiguration für eine Basisinstallation. Abgesehen von den unten angegebenen Abweichungen sollte dieser Beschreibung gefolgt werden. Beim Speichern der Masterkonfiguration achten Sie darauf, dass Sie die Synchronisation über die Administrationskonsole manuell erzwingen.

1. Navigieren Sie zu **Systemadministration** > **Änderungen an Master-Repository speichern**.
2. Aktivieren Sie das Kontrollkästchen **Änderungen mit Knoten synchronisieren**.
3. Klicken Sie auf die Schaltfläche **Speichern**. Die Synchronisation kann einige Zeit in Anspruch nehmen.
4. Prüfen Sie das System und/oder die Protokolle von WebSphere Application Server auf die Vollständigkeit der Synchronisation. Die Nachrichten können sich je nach Release von WebSphere Application Server unterscheiden; suchen Sie jedoch nach Nachrichten ähnlich den folgenden:

```
ADMS0208I: Die Konfigurationssynchronisation für die Zelle ist abgeschlossen.
```

   Nachdem die Synchronisation abgeschlossen ist, überprüfen Sie den Serverstatus und die verschiedenen WebSphere Application Server-Protokolle, um sich von der erfolgreichen Durchführung zu überzeugen.

Unter „Verwaltungssicherheit konfigurieren" auf Seite 26 finden Sie Details zu der Sicherheitseinrichtung, die für die manuelle Konfiguration erforderlich ist. Für diese Einrichtung muss die Datei `Registry.jar` in ein Verzeichnis innerhalb der WebSphere Application Server-Installation kopiert werden. Die Datei `Registry.jar` muss aus `CuramSDEJ/lib` in das Verzeichnis `lib` der Deployment Manager-Installation und in alle eingebundenen Installationen kopiert werden.

„Verwaltungssicherheit konfigurieren" auf Seite 26 Diese Sicherheitseinrichtung erfordert auch, dass die Datei `CryptoConfig.jar` in das Verzeichnis `java/jre/lib/ext` der WebSphere Application Server-Installation kopiert wird. Für jede andere WebSphere Application Server-Installation der Umgebung sollte die Datei `CryptoConfig.jar` in dieselbe Verzeichnisstruktur kopiert werden.

Anmerkung: Vor dem Erstellen der Datei `Curam.ear` für die Implementierung ist es sinnvoll, die *BOOTSTRAP_ADDRESS* des Servers darauf hinzuweisen, dass sie

dorthin installiert werden. Die *BOOTSTRAP_ADDRESS* befindet sich in derselben Liste von Ports wie die bereits beschriebene *SOAP_CONNECTOR_ADDRESS*.

Standardmäßig hat die von der Anwendung erwartete *BOOTSTRAP_ADDRESS* den Wert '2809'. Um dieses Problem zu beheben, ändern Sie entweder diese Adresse oder die entsprechende Eigenschaft in Ihrer `AppServer.properties`-Datei.

Die zu ändernde Eigenschaft ist der Wert 'curam.server.port' in der Datei `AppServer.properties`. Das Ändern dieses Wertes wirkt sich beim Erstellen einer EAR-Datei auf den Portwert in der `web.xml`-Datei aus. Weitere Informationen zu der Datei `web.xml` finden Sie im Dokument *Cúram Web Client Reference Manual* (Cúram-Referenzhandbuch zum Webclient).

Bereitstellungen auf dem Knoten

Zum Schluss sollten die Anwendungen anhand der Anweisungen unter „Manuelle Anwendungsimplementierung” auf Seite 41 manuell auf dem erforderlichen Server implementiert werden. Daraufhin ist es möglich, Anwendungen mithilfe der Deployment Manager-Administrationskonsole zu starten und zu stoppen.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM-Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden. Für die in diesem Handbuch beschriebenen Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing

IBM Europe, Middle East & Africa

Tour Descartes

2, avenue Gambetta

92066 Paris La Defense

France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden.

Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen. Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar.

Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht. Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation

Dept F6, Bldg 1

294 Route 100

Somers NY 10589-3216

U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Bereitstellung des in diesem Dokument beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen.

IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Alle von IBM angegebenen Preise sind empfohlene Richtpreise und können jederzeit ohne weitere Mitteilung geändert werden. Händlerpreise können u. U. von den hier genannten Preisen abweichen.

Diese Veröffentlichung dient nur zu Planungszwecken. Die in dieser Veröffentlichung enthaltenen Informationen können geändert werden, bevor die beschriebenen Produkte verfügbar sind.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellensprache geschrieben sind und Programmiertechniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Musterprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. IBM kann daher die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programm nicht garantieren oder implizieren. Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch Ihre Verwendung der Musterprogramme entstehen.

Kopien oder Teile der Musterprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihres Unternehmens) (Jahr). Teile des vorliegenden Codes wurden aus Musterprogrammen der IBM Corp. abgeleitet.

© Copyright IBM Corp. _Jahreszahl oder Jahreszahlen eingeben_. Alle Rechte vorbehalten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farbabbildungen.

Hinweise zur Datenschutzrichtlinie

IBM Softwareprodukte, einschließlich Software as a Service-Lösungen ("Softwareangebote"), können Cookies oder andere Technologien verwenden, um Informationen zur Produktnutzung zu erfassen, die Endbenutzererfahrung zu verbessern und Interaktionen mit dem Endbenutzer anzupassen oder zu anderen Zwecken. In vielen Fällen werden von den Softwareangeboten keine personenbezogenen Daten erfasst. Einige der IBM Softwareangebote können Sie jedoch bei der Erfassung personenbezogener Daten unterstützen. Wenn dieses Softwareangebot Cookies zur Erfassung personenbezogener Daten verwendet, sind nachfolgend nähere Informationen über die Verwendung von Cookies durch dieses Angebot zu finden.

Je nachdem, welche Konfigurationen implementiert wurden, ist es möglich, dass dieses Softwareangebot Sitzungscookies und persistente Cookies zum Erfassen der Namen, Benutzernamen, Kennwörter, Profilnamen oder anderer personenbezogener Daten einzelner Benutzer für die Sitzungsverwaltung, Authentifizierung, Single-Sign-on-Konfiguration oder für einen besseren Bedienungskomfort und/oder andere Zwecke der Nutzungsverfolgung bzw. funktionale Einsatzmöglichkeiten. Diese Cookies oder ähnliche Technologien können nicht inaktiviert werden.

Wenn die für dieses Softwareangebot genutzten Konfigurationen Sie als Kunde in die Lage versetzen, personenbezogene Daten von Endbenutzern über Cookies und andere Technologien zu erfassen, müssen Sie sich zu allen gesetzlichen Bestimmungen in Bezug auf eine solche Datenerfassung, einschließlich aller Mitteilungspflichten und Zustimmungsanforderungen, rechtlich beraten lassen.

Weitere Informationen zur Nutzung verschiedener Technologien, einschließlich Cookies, für diese Zwecke finden Sie in der "IBM Online-Datenschutzerklärung, Schwerpunkte" unter http://www.ibm.com/privacy und in der "IBM Online-Da-

tenschutzerklärung" unter http://www.ibm.com/privacy/details im Abschnitt "Cookies, Web-Beacons und sonstige Technologien" und unter "IBM Software Products and Software-as-a-Service Privacy Privacy Statement" unter http://www.ibm.com/software/info/product-privacy.

Marken

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der International Business Machines Corporation. Weitere Produkt- oder Servicenamen können Marken von IBM oder anderen Herstellern sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Website "Copyright and trademark information" unter http://www.ibm.com/legal/us/en/copytrade.shtml.

Apache ist eine eingetragene Marke der Apache Software Foundation.

Microsoft und Windows sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke der Open Group in den USA und/oder anderen Ländern.

Oracle, Java und alle Java-basierten Marken sind eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Andere Namen können Marken der jeweiligen Rechtsinhaber sein. Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken oder Dienstleistungsmarken anderer Unternehmen sein.

Gedruckt in Deutschland