

IBM Security Directory Integrator
Version 7.2.0.1

*Installations- und
Administratorhandbuch*



IBM Security Directory Integrator
Version 7.2.0.1

*Installations- und
Administratorhandbuch*



Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen in „Bemerkungen“ auf Seite 427 gelesen werden.

Impressum

Diese Ausgabe bezieht sich auf version 7.2.0.1 des Lizenzprogramms *IBM Security Directory Integrator (5724-K74)* und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs *IBM Security Director Integrator, Version 7.2.0.1, Installation and Administration Guide*, IBM Form SC27-2705-03, herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2003, 2014

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
TSC Germany
Kst. 2877
Juni 2014

Inhaltsverzeichnis

Informationen zu dieser Veröffentlichung ix

Zugriff auf Veröffentlichungen und Terminologie	ix
Eingabehilfen	xi
Technische Schulung	xi
Informationen zur Unterstützung	xii
Erklärung zu geeigneten Sicherheitsvorkehrungen	xii

Kapitel 1. Einführung 1

Editionen von IBM Security Directory Integrator	1
---	---

Kapitel 2. Installationsanweisungen für IBM Security Directory Integrator 3

Installationsvorbereitung	3
Erforderlicher Plattenspeicherplatz	3
Speicherbedarf.	3
Plattformvoraussetzungen	3
Komponenten in IBM Security Directory Integrator.	3
Weitere Voraussetzungen	6
Rootberechtigung oder Administratorrechte	6
Security Enhanced Linux (SELinux).	6
Authentifizierung von AMC unter UNIX/Linux	7
Grafikpakete für UNIX-Systeme	8
Voraussetzungen für den Konfigurationseditor auf AIX-Betriebssystemen	8
Voraussetzungen für ein Upgrade von Version 7.1.1 auf Version 7.2 auf Windows 2012-Betriebssystemen.	9
IBM Security Directory Integrator installieren	9
Geeignetes Installationsprogramm starten	10
Plattformspezifisches Installationsprogramm von IBM Security Directory Integrator verwenden	13
Installation mit dem grafisch orientierten Installationsprogramm ausführen	13
Anzeigenfolge bei der Installation	13
Anzeigenfolge bei der Deinstallation	34
Anzeigenfolge beim Hinzufügen von Komponenten	39
Anzeigenfolge bei der Migration	42
Installation über die Befehlszeile ausführen.	44
Speicherbedarf für temporäre Dateien während der Installation	46
Unbeaufsichtigte Installation ausführen	46
Einschränkung für Servicenamen auf UNIX-Systemen	47
Schritte für den Installationsabschluss	47
Aktualisierungsseite des Konfigurationseditors	47
Plug-ins	48
Administration and Monitoring Console (AMC)	48
Dokumentation	49
Migration	49
Lokale Hilfedateien installieren.	49
AMC in angepasster ISC SE-Instanz oder IBM Dashboard Application Services Hub implementieren	51

Installation oder Aktualisierung mit Eclipse-Aktualisierungsmanager vornehmen	52
Schritte für den Installationsabschluss	54
Deinstallation.	55
Deinstallationsprogramm starten	55
Unbeaufsichtigte Deinstallation ausführen	56
Standardinstallationspositionen.	57
Standardlösungsverzeichnis	57

Kapitel 3. Update Installer 59

Datei ".registry"	61
Installation von Fixpacks	64
Rollback durchführen	64
Fehlerbehebung	64

Kapitel 4. Unterstützte Plattformen. 65

Kapitel 5. Migration 67

Dateien auf andere Position migrieren	67
Welche Dateien müssen zur Verwendung an einer anderen Position nicht modifiziert werden?	67
Welche Dateien müssen modifiziert werden, damit sie an einer anderen Position verwendet werden können?	68
Welche Dateien sollten unter normalen Umständen nicht an einer anderen Position verwendet werden?	69
Dateien mit verschlüsselten Daten migrieren	70
Dateien auf eine neuere Version migrieren	70
Migration mit Unterstützung durch das Installationsprogramm	70
Migration mit Unterstützung durch ein Tool	71
Manuelle Migration	72
Wichtige Daten sichern	86
Durch das Installationsprogramm gesicherte Dateien.	86
Upgrade von Version 6.0 auf Version 7.1.	86
Upgrade von Version 6.1 auf Version 7.1.	86
Upgrade von Version 7.0 auf Version 7.1.	86
Upgrade von Version 7.1 auf Version 7.1.1	87
Upgrade von Version 7.1.1 auf Version 7.2	87
Sicherungstools	87
Manuelle Sicherung	88
Konfigurationseinstellungen von AMC 7.x auf eine andere AMC-Implementierung migrieren	88
EventHandler in entsprechende Fertigungslinie konvertieren	89
TCP-Server-Connector	91
Mailbox-Connector	91
JMX-Connector	91
SNMP-Server-Connector	92
IBM Security Directory Server-Änderungsprotokollconnector.	92
HTTP-Server-Connector	93
LDAP-Server-Connector	94

Änderungserkennungsconnector für Sun Directory	94
Änderungserkennungsconnector für Active Directory	95
DSMLv2-SOAP-Server-Connector	95
B-Baum-Tabellen und B-Baum-Connector auf Systemspeicher migrieren	97
Cloudscape-Datenbank auf Derby migrieren	97
Dateien "global.properties" und "solution.properties" mit dem Migrationstool migrieren	99
Eigenschaftendateien der Kennwort-Plug-ins mit dem Migrationstool migrieren	100

Kapitel 6. Sicherheit 103

Schlüssel, Zertifikate und Schlüsselspeicher verwalten	103
Hintergrund	103
Öffentliche/private Schlüssel und Zertifikate	104
Geheime Schlüssel	104
Schlüsselspeicher	104
Schlüssel für SSL	105
Schlüssel für die Verschlüsselung	106
Tools	106
Inhalt eines Schlüsselspeichers auflisten	107
Schlüssel erstellen	107
Unterstützung für Secure Sockets Layer (SSL)	110
SSL-Konfiguration von IBM Security Directory Integrator-Komponenten als Server	111
SSL-Konfiguration von IBM Security Directory Integrator-Komponenten als Client	113
SSL-Clientauthentifizierung	114
SSL-Konfiguration für IBM Security Directory Integrator und Microsoft Active Directory	114
Zusammenfassung der Eigenschaften für die SSL-Aktivierung und die PKCS#11-Unterstützung	116
Beispiel für SSL	117
IBM Security Directory Integrator-Komponente als Server	117
IBM Security Directory Integrator-Komponente als Client	118
Ferne Server-API	119
Server-API konfigurieren	120
Zugriff auf ferne Server-API in einem virtuellen privaten Netz	123
Zugriffsoptionen für die Server-API	123
SSL-Fernzugriff auf Server-API	124
Spezielle SSL-Eigenschaften der Server-API verwenden	124
Java-Standardystemeigenschaften für SSL verwenden	125
Server-API-Authentifizierung	126
Lokale Clientsitzung	126
Ferne Clientsitzung	126
JAAS-Authentifizierung	127
SSL-basierte Authentifizierung	127
Auf Benutzername/Kennwort basierende Authentifizierung	128
Unterstützung der LDAP-Authentifizierung	130
Konfiguration der LDAP-Authentifizierung	130

LDAP-Authentifizierungslogik	131
Unterstützung von LDAP-Gruppen	132
Hostbasierte Authentifizierung	134
Zusammenfassung der Optionen für die Server-API-Authentifizierung	135
JMX-Schicht der Server-API	136
Beispiele für Konfiguration der Server-API-Authentifizierung	136
Server-API-Autorisierung	138
Berechtigungsrollen	138
Benutzerregistry der Server-API	140
Prüffunktionen des Servers	144
Geltungsbereich der Prüfung	144
Benachrichtigungen unterdrücken	145
Benachrichtigungen senden	145
Sicherheit für IBM Security Directory Integrator-Serverinstanz	146
Stashdatei	147
Sicherheitsmodi für den Server	148
Mit verschlüsselten IBM Security Directory Integrator-Konfigurationsdateien arbeiten	149
Verschlüsselte IBM Security Directory Integrator-Konfigurationsdatei neu erstellen	151
Verschlüsselte IBM Security Directory Integrator-Konfigurationsdatei bearbeiten	152
Standardverschlüsselung der Datei 'global.properties' oder 'solution.properties'	152
Verschlüsselung von Eigenschaften in externen Eigenschaftendateien	153
Verschlüsselungsdienstprogramm von IBM Security Directory Integrator	153
Sicherheit für IBM Security Directory Integrator-Systemspeicher	155
Verschiedene Funktionen für Konfigurationsdateien	157
Kennwortschutz für Komponenten	158
Kennwörter für konfigurierte Eigenschaften speichern	158
Ausgabe von Attributen als unverschlüsselter Text während Traceerstellung verhindern	159
Verschlüsselung von IBM Security Directory Integrator-Server-Hooks	160
Ferner Konfigurationseditor und SSL	160
Fernes Konfigurationseditor verwenden	160
Übersicht über Konfigurationsdateien und Eigenschaften für Sicherheit	162
Sicherheit für die Webverwaltungskonsole	166
Sonstige Sicherheitsaspekte	166
HTTP-Basisauthentifizierung	166
Spezielle Aspekte für SSL bei Lotus Domino	166
Zertifikate für Web-Service-Suite von IBM Security Directory Integrator	167
IBM WebSphere MQ Everyplace-Authentifizierung mit Minizertifikaten	168

Kapitel 7. Regelsteuerkomponente für Verbindungswiederherstellung 169

Regeln für die Verbindungswiederherstellung	170
Konfiguration von benutzerdefinierten Regeln	172
Hinweise zu Ausnahmbedingungen	173
Allgemeine Konfiguration der Verbindungswiederherstellung	174

Kapitel 8. Systemwarteschlange	177
Konfiguration der Systemwarteschlange	177
Apache ActiveMQ-Parameter	178
Konfiguration	178
Protokollierung.	179
SSL mit ActiveMQ verwenden.	179
Parameter für IBM WebSphere MQ Everyplace	180
Parameter für IBM WebSphere MQ	180
Parameter für Microbroker	181
Parameter für JMS-Scripttreiber	181
JavaScript-Objekt "env"	182
JavaScript-Objekt "ret"	183
JavaScript-Beispiel für Fiorano MQ	183
Beispiel für Konfiguration der Systemwarte-	
schlange	184
Sicherheit und Authentifizierung	184
Konfigurationsdienstprogramm für IBM Web-	
Sphere MQ Everyplace	185
Sicherheit für Warteschlange durch Authentifi-	
zierung von IBM WebSphere MQ Everyplace-	
Nachrichten	186
Unterstützung für DNS-Namen in der Konfigu-	
ration der IBM WebSphere MQ Everyplace-War-	
teschlange	187
Konfiguration von Hochverfügbarkeit für den	
IBM WebSphere MQ Everyplace-Transport von	
Kennwortänderungen	188
Funktionen im Konfigurationsdienstprogramm	
für IBM WebSphere MQ Everyplace für die fer-	
ne Konfiguration	188
Kapitel 9. Verschlüsselung und FIPS-	
Modus	191
IBM Security Directory Integrator zur Ausführung	
im FIPS-Modus konfigurieren	191
Unterstützung des symmetrischen Chiffrierwerts	
FIPS-Verschlüsselung	192
Connectors, Funktionskomponenten, Par-	
ser	192
IBM Security Directory Integrator-Server	
und FIPS	194
SSL- und PKI-Zertifikate konfigurieren	202
Verschlüsselung und Entschlüsselung mit dem	
Dienstprogramm "CryptoUtils" vornehmen	203
Mit Zertifikaten arbeiten.	203
Unterschiede zwischen selbst signierten und	
von Zertifizierungsstellen signierten Zertifi-	
katen	204
Zertifikate mit PKI und SSL konfigurieren	
.	205
Verschlüsselungsschlüssel auf Hardwareeinheiten	
verwenden	206
IBMPCS11 verwenden	207
Auffüllung aktivieren oder inaktivieren.	207
Verschlüsselungsartefakte (Schlüssel, Zertifikate,	
Schlüsselspeicher, verschlüsselte Dateien) verwal-	
ten	208
Kapitel 10. IBM Security Directory In-	
tegrator-Server-API konfigurieren.	211
Server-ID	211

Ausnahmebedingung für kennwortgeschützte Kon-	
figurationen	212
Server-RMI	212
Zeitlimitintervall für Konfigurationsladevorgang	
	212

Kapitel 11. Eigenschaften 213

Mit Eigenschaften arbeiten	213
Globale Eigenschaften	214
Lösungseigenschaften	215
Java-Eigenschaften.	215
Systemeigenschaften	216

Kapitel 12. Systemspeicher 219

Eigenschaftsspeicher	223
Managementsystem für relationale Datenbanken ei-	
nes anderen Anbieters als Systemspeicher	223
Oracle	224
MS SQL Server	225
IBM DB2	226
IBM solidDB	226
Derby als Systemspeicher verwenden	226
Apache Derby-Instanzen konfigurieren	228
Apache Derby im Netzmodus starten	228
Benutzerauthentifizierung im Systemspeicher	
aktivieren	229
Erstellungsanweisungen für Systemspeicherta-	
bellen	229
Apache Derby-Datenbanken sichern.	230
Fehlerbehebung für Apache Derby	231

Kapitel 13. Befehlszeilenoptionen. 233

Konfigurationseditor	233
Server	234
CLI - Dienstprogramm "tdisrvctl".	238
Befehlszeilenreferenz	239
Operationen	240

Kapitel 14. Protokollierung und Debug 253

Scriptbasierte Protokollierung	254
Protokollierung mit Log4J-Standardklasse	255
Protokollebenen und Protokollebenensteuerung	
	259
Log4J-Standardparameter	260
Eigene Protokollstrategien erstellen	261

Kapitel 15. Traceerstellung und First-Failure Data Capture 263

Erweiterungen für die Traceerstellung	263
Wissenswertes über die Traceerstellung.	263
Traceerstellung konfigurieren	264
Tracestufen dynamisch festlegen	264
Nützliche JLOG-Parameter	266

Kapitel 16. Verwaltung und Überwachung 267

Installation und Konfiguration.	267
AMC in Integrated Solutions Console imple-	
mentieren	267

AMC mit dem Installationsprogramm von IBM Security Directory Integrator als Windows-Dienst oder UNIX-Prozess implementieren	268
AMC in einer vorhandenen IBM WebSphere Application Server-Umgebung implementieren	269
AMC und Action Manager starten sowie Anmeldung durchführen	269
AMC aktivieren	270
Action Manager über Fernzugriff ausführen	271
AMC-Protokolle	272
AMC in ISC	273
Action Manager	274
Action Manager aktivieren	281
Action Manager-Status in Echtzeit	282
Auslöser für eine bestimmte Regel über AMC erzwingen	283
Sicherheit bei AMC und Action Manager	283
AMC und SSL	284
AMC und ferne IBM Security Directory Integrator-Server	286
AMC und Rollenverwaltung	287
AMC und Kennwörter	288
AMC und verschlüsselte Konfigurationen	288
AMC-Benutzerschnittstelle	289
Bei der Konsole an- und abmelden	289
Layout der AMC-Konsole	290
Bei der Konsole abmelden	291
AMC-Tabellen verwenden	291
Dropdown-Menü "Aktion auswählen"	292
Blättern	292
Sortieren	292
Suchen	293
Filtern	293
Server	294
Server hinzufügen	295
Server modifizieren	295
Konsoleigenschaften	296
Lösungsansichten	297
ACLs konfigurieren	298
Lokale Variablen	299
Lösungsansicht hinzufügen	299
Konfigurationsdateien (zum Laden/Entladen von Konfigurationen)	302
Angepasstes Laden	303
Statusüberwachung und Action Manager	303
Status überwachen	304
Details der Lösungsansicht	304
Komponenten anzeigen	308
Bevorzugte Lösungsansichten anzeigen	308
Details der Lösungsansicht in AMC aktualisieren	308
Action Manager	309
Konfigurationsregeln hinzufügen/bearbeiten	309
Aktion hinzufügen/modifizieren	311
Variable für Ereignisdaten ersetzen	316
Zusammenfassung der Regeln anzeigen	318
Eigenschaftsspeicher	318
Protokollverwaltung	320

Bevorzugte Lösungsansichten	321
Befehlszeilendienstprogramme für AMC und Action Manager	321
Beispielablauf der Erstellung von Lösungsansicht und Regeln	327

Kapitel 17. Touchpoint-Server 335

Touchpoint-Konzepte	335
Touchpoint-Server	335
Touchpoint-Provider	336
Touchpoint-Typ	336
Touchpoint-Instanz	338
Touchpoint-Schablone	342
Ressourcenpersistenz	347
Touchpoint-Schema	348
Touchpoint-Konfiguration	352
Instanzkonfiguration	352
Zielkonfiguration	353
Kommunikationsprotokoll der Touchpoint-Instanz	353
Touchpoint-Instanz mit der Rolle "Provider"	353
Touchpoint-Instanz mit der Rolle "Initiator"	355
Touchpoint-Instanz mit der Rolle "Intermediary"	355
Darstellung von Eintragsobjekten als HTTP-Inhalt	355
Schema für Touchpoint-Statuseintrag	356
Eigenschaftenblattdefinitionen	357
XML-Schemapositionen	358
Abläufe bei einem Fehler	358
Konfiguration	360
Authentifizierung	361
Beispiele	362
Im Lieferumfang enthaltenes Beispiel	362
Beispielschritte für die Erstellung einer Touchpoint-Instanz mit einem JDBC-Connector	362
Touchpoint-Instanz mit der Rolle "Provider"	363
Touchpoint-Instanz mit der Rolle "Initiator"	364
Touchpoint-Instanz mit der Rolle "Intermediary"	365

Kapitel 18. Tombstone Manager 367

Tombstones konfigurieren	367
Konfigurationsanzeige des Konfigurationseditors	368
Konfigurationsanzeige für Fertigungslinien	370
Tombstone Manager	371

Kapitel 19. Mehrere IBM Security Directory Integrator-Services 375

IBM Security Directory Integrator als Windows-Dienst	375
Dienst installieren und deinstallieren	375
Dienst installieren	376
Dienst deinstallieren	376
Dienst starten und stoppen	377
Protokollierung	377
Dienst konfigurieren	377
IBM Security Directory Integrator als Linux/UNIX-Dienst	380

Befehlszeilenunterstützung 382

Anhang A. Beispiele für Eigenschaftendateien. 383

Log4J.properties 384

jlog.properties 385

derby.properties 387

global.properties 387

Anhang B. Überwachung mit externen Tools 393

IBM Security Directory Integrator mit ITM überwachen 394

 Kurzdarstellung der ITM-Architektur 394

 Vorhandene Agentenkonfiguration in ITM Agent

 Builder 6.2 importieren 395

 IBM SDI-Agenten für ITM mit ITM Agent Builder 6.2 erstellen 395

 ITM-Agenten generieren. 404

 ITM-Agenten konfigurieren. 405

 IBM Security Directory Integrator-Daten überwachen 406

Schwellenwerte definieren 408

Links zwischen Tabellen erstellen. 412

 Zweck von Links 412

 Erstellung von Links 413

Angepasste Benachrichtigungen an ITM senden 419

Einschränkungen 420

IBM SDI mit OMNIBus überwachen. 420

Eigenschaftendatei für EIF-Testmonitor konfigurieren 420

Bewertung der Ereignisse bestimmen 421

Mit der Datei "EventPropertyFile.properties" arbeiten 421

Angepasste Benachrichtigungen an OMNIBus senden 423

Anhang C. Eingabehilfefunktionen in IBM Security Directory Integrator. . . 425

Bemerkungen 427

Index 431

Informationen zu dieser Veröffentlichung

Die Informationen in der vorliegenden Veröffentlichung benötigen Sie, um Lösungen mit Komponenten zu entwickeln, die Bestandteil von IBM® Security Directory Integrator sind.

Die IBM Security Directory Integrator-Komponenten wurden für Netzadministratoren konzipiert, in deren Zuständigkeitsbereich die Verwaltung von Benutzerverzeichnissen und anderen Ressourcen fällt. Es wird davon ausgegangen, dass Sie über praktische Erfahrung mit der Installation und der Verwendung sowohl von IBM Security Directory Integrator als auch von IBM Security Directory Server verfügen.

Die Informationen sind darüber hinaus für Benutzer gedacht, die für die Entwicklung, Installation und Verwaltung von Lösungen unter Verwendung von IBM Security Directory Integrator zuständig sind. Außerdem müssen Sie mit den Konzepten und der Verwaltung der Systeme vertraut sein, mit denen die entwickelte Lösung verbunden wird. Abhängig von der jeweiligen Lösung könnte es sich hierbei unter anderem um eines der folgenden Produkte, Systeme und Konzepte handeln:

- IBM Security Directory Server
- IBM Security Identity Manager
- IBM Java™ Runtime Environment (JRE) oder Oracle Java Runtime Environment
- Microsoft Active Directory
- Betriebssysteme Windows und UNIX
- Sicherheitsmanagement
- Internetprotokolle, zum Beispiel Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS) und Transmission Control Protocol/Internet Protocol (TCP/IP)
- Lightweight Directory Access Protocol (LDAP) und Verzeichnisservices
- Eine unterstützte Benutzerregistry
- Authentifizierungs- und Berechtigungskonzepte
- SAP ABAP-Anwendungsserver

Zugriff auf Veröffentlichungen und Terminologie

Lesen Sie die Beschreibungen der IBM Security Directory Integrator Version 7.2.0.1-Bibliothek und der Referenzliteratur, auf die Sie online zugreifen können.

In diesem Abschnitt finden Sie die folgenden Informationen:

- Eine Liste der Veröffentlichungen in der „IBM Security Directory Integrator-Bibliothek“.
- Links zu „Onlineveröffentlichungen“ auf Seite xi.
- Einen Link zur „IBM Terminologiewebsite“ auf Seite xi.

IBM Security Directory Integrator-Bibliothek

Die folgenden Dokumente sind in der IBM Security Directory Integrator-Bibliothek verfügbar:

- *IBM Security Directory Integrator Version 7.2.0.1 Federated Directory Server Administration Guide*
Diese Veröffentlichung enthält Informationen zum Entwerfen, Implementieren und Verwalten von Datenintegrationslösungen über die Federated Directory Server-Konsole. Sie enthält ferner Informationen zur Verwendung des Protokolls 'System for Cross-Domain Identity Management' (SCIM) und der Schnittstelle für das Identitätsmanagement.
- *IBM Security Directory Integrator Version 7.2.0.1 Erste Schritte*
Diese Veröffentlichung enthält ein kurzes Lernprogramm und eine Einführung in IBM Security Directory Integrator. Sie umfasst Beispiele zur Erstellung der benötigten Interaktionsabläufe und praktische Übungen, mit deren Hilfe Sie sich mit IBM Security Directory Integrator vertraut machen können.
- *IBM Security Directory Integrator Version 7.2.0.1 Benutzerhandbuch*
Diese Veröffentlichung enthält Informationen zur Verwendung von IBM Security Directory Integrator. Darüber hinaus enthält sie Anweisungen zum Entwurf von Lösungen mithilfe des Security Directory Integrator-Entwicklertools (des Konfigurationseditors) sowie zur Ausführung vordefinierter Lösungen über die Befehlszeile. Sie bietet außerdem Informationen zu den verwendeten Schnittstellen und Konzepten sowie zur Erstellung von Fertigungslinien.
- *IBM Security Directory Integrator Version 7.2.0.1 Installations- und Administratorhandbuch*
Diese Veröffentlichung enthält vollständige Informationen zur Installation, Migration von einer Vorversion, Konfiguration der Protokollierungsfunktion und zum Sicherheitsmodell, das der fernen Server-API von IBM Security Directory Integrator zugrunde liegt. Darüber hinaus enthält sie Informationen dazu, wie Lösungen implementiert und verwaltet werden.
- *IBM Security Directory Integrator Version 7.2.0.1 Reference Guide*
Diese Veröffentlichung enthält detaillierte Informationen zu den einzelnen Komponenten von IBM Security Directory Integrator (Connector, Funktionskomponenten, Parser, Objekte usw.), die die Bausteine der Fertigungslinie darstellen.
- *IBM Security Directory Integrator Version 7.2.0.1 Problem Determination Guide*
Diese Veröffentlichung enthält Informationen zu den Tools, Ressourcen und Verfahren in IBM Security Directory Integrator, die Sie bei der Identifikation und Lösung von Problemen unterstützen können.
- *IBM Security Directory Integrator Version 7.2.0.1 Message Guide*
Diese Veröffentlichung enthält eine Liste aller Informationsnachrichten, Warnungen und Fehlernachrichten zu IBM Security Directory Integrator.
- *IBM Security Directory Integrator Version 7.2.0.1 Password Synchronization Plug-ins Guide*
Diese Veröffentlichung enthält umfassende Informationen zur Installation und Konfiguration der folgenden fünf IBM Plug-ins zur Kennwortsynchronisation: Windows Password Synchronizer, Sun Directory Server Password Synchronizer, IBM Security Directory Server Password Synchronizer, Domino Password Synchronizer und Password Synchronizer für UNIX und Linux. Darüber hinaus sind Konfigurationsanweisungen für den LDAP- und den JMS-Kennwortspeicher enthalten.
- *IBM Security Directory Integrator Version 7.2.0.1 Release Notes*
Diese Veröffentlichung enthält Beschreibungen neuer Funktionen sowie kurzfristig aktualisierte Informationen zu IBM Security Directory Integrator, die nicht mehr in die Dokumentation aufgenommen werden konnten.

Onlineveröffentlichungen

IBM stellt Produktveröffentlichungen, wenn das Produkt freigegeben wird und wenn die Veröffentlichungen aktualisiert werden, an den folgenden Positionen bereit:

IBM Security Directory Integrator-Bibliothek

Auf der Website mit der Produktdokumentation (<http://www-01.ibm.com/support/knowledgecenter/SSCQGF/welcome>) wird die Einführungsseite angezeigt und Sie finden Informationen über die Navigation in dieser Bibliothek.

IBM Security Systems Documentation Central

IBM Security Systems Documentation Central bietet eine alphabetische Liste aller IBM Security Systems-Produktbibliotheken sowie Links zur Online-dokumentation für bestimmte Versionen der einzelnen Produkte.

IBM Publications Center

Die Website mit dem IBM Publications Center (<http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>) bietet angepasste Suchfunktionen, mit denen Sie alle benötigten IBM Veröffentlichungen finden.

Zugehörige Informationen

Referenzinformationen zu IBM Security Directory Integrator finden Sie hier:

- IBM Security Directory Integrator verwendet den JNDI-Client von Oracle. Informationen zum JNDI-Client finden Sie in der Veröffentlichung *Java Naming and Directory Interface™ Specification* unter <http://download.oracle.com/javase/7/docs/technotes/guides/jndi/index.html>.
- Informationen, die Sie bei der Beantwortung von Fragen zu IBM Security Directory Integrator unterstützen, finden Sie unter https://www-947.ibm.com/support/entry/myportal/over-accesspubsview/software/security_systems/tivoli_directory_integrator.

IBM Terminologiewebsite

Die IBM Terminologiewebsite enthält die konsolidierte Terminologie aus Produktbibliotheken an einer einzigen Position. Sie finden die Terminologiewebsite unter der folgenden Adresse: <http://www.ibm.com/software/globalization/terminology>.

Eingabehilfen

Funktionen zur behindertengerechten Bedienung (Eingabehilfefunktionen) unterstützen Benutzer mit körperlichen Behinderungen, wie z. B. eingeschränkter Bewegungsfähigkeit oder Sehkraft, beim erfolgreichen Einsatz von Softwareprodukten. Dieses Produkt unterstützt behindertengerechte Tools, die die Elemente der Benutzerschnittstelle ansagen und die Navigation in dieser Schnittstelle vereinfachen. Sie können alle Funktionen der grafischen Benutzerschnittstelle anstatt mit der Maus auch über die Tastatur aufrufen.

Weitere Informationen finden Sie im Anhang zu den Eingabehilfen im *Directory Integrator - Konfiguration*.

Technische Schulung

Informationen zu technischen Schulungen finden Sie auf der folgenden IBM Education-Website unter <http://www.ibm.com/software/tivoli/education>.

Informationen zur Unterstützung

IBM Support bietet Unterstützung bei codebezogenen Problemen sowie bei Fragen zur Routine, zur kurzfristigen Installation oder zur Verwendung. Sie können direkt auf die IBM Software Support-Website unter folgender Adresse zugreifen: <http://www.ibm.com/software/support/probsub.html>.

Fehlerbehebung bietet detaillierte Informationen zu folgenden Themen:

- Welche Informationen vor der Kontaktaufnahme mit dem IBM Support zusammengestellt werden sollten.
- Die verschiedenen Methoden zur Kontaktaufnahme mit dem IBM Support.
- Die Verwendung von IBM Support Assistant.
- Anweisungen und Ressourcen zur Fehlerbestimmung, um das Problem einzugrenzen und zu beheben.

Erklärung zu geeigneten Sicherheitsvorkehrungen

Zur Sicherheit von IT-Systemen gehört der Schutz von Systemen und Informationen in Form von Vorbeugung, Erkennung und Reaktion auf unbefugten Zugriff innerhalb des Unternehmens und von außen. Unbefugter Zugriff kann dazu führen, dass Informationen geändert, gelöscht, veruntreut oder missbräuchlich verwendet werden. Ebenso können Ihre Systeme beschädigt oder missbräuchlich verwendet werden, einschließlich zum Zweck von Attacken. Kein IT-System oder Produkt kann umfassend als sicher betrachtet werden. Kein einzelnes Produkt, kein einzelner Service und keine einzelne Sicherheitsmaßnahme können eine unbefugte Verwendung oder einen unbefugten Zugriff mit vollständiger Wirksamkeit verhindern. IBM Systeme, Produkte und Services werden als Teil eines umfassenden Sicherheitskonzepts entwickelt, sodass die Einbeziehung zusätzlicher Betriebsprozesse erforderlich ist. IBM übernimmt keine Gewähr dafür, dass Systeme, Produkte oder Services vor zerstörerischen oder unzulässigen Handlungen Dritter geschützt sind oder dass Systeme, Produkte oder Services Ihr Unternehmen vor zerstörerischen oder unzulässigen Handlungen Dritter schützen.

Kapitel 1. Einführung

Bevor Sie mit den Installations- und Verwaltungstasks beginnen, machen Sie sich mit den allgemeinen Konzepten von IBM Security Directory Integrator vertraut.

Einen Überblick über die allgemeinen Konzepte von IBM Security Directory Integrator enthält der Abschnitt "IBM Security Directory Integrator - Konzepte" in der Veröffentlichung *Directory Integrator - Konfiguration*.

Detailliertere Informationen zu den Konzepten von IBM Security Directory Integrator finden Sie im Handbuch *Referenzinformationen*.

Editionen von IBM Security Directory Integrator

Der folgende Abschnitt enthält Informationen zu den verschiedenen Editionen des Produkts.

IBM Security Directory Integrator Version 7.2 wird in zwei verschiedenen Editionen angeboten, für die unterschiedliche Lizenzvereinbarungen gelten:

- General Purpose Edition: Die Lizenzierung für diese Edition erfolgt auf Prozessorbasis.
- Identity Edition: Die Lizenzierung für diese Edition erfolgt auf Benutzerbasis.

Im Unterschied zu früheren Versionen von IBM Security Directory Integrator sind in Version 7.2 die General Purpose Edition und die Identity Edition in Inhalt, Funktionalität und Leistungsspektrum identisch. Sie bieten lediglich unterschiedliche Lizenzvereinbarungen an.

Kapitel 2. Installationsanweisungen für IBM Security Directory Integrator

Bei der Installation von IBM Security Directory Integrator müssen Sie vorab die Voraussetzungen prüfen, dann die Software installieren und abschließend bestimmte Tasks ausführen, damit die Software funktioniert.

Installationsvorbereitung

Sie müssen vor der Installation von IBM Security Directory Integrator sicherstellen, dass Ihr System die Mindestvoraussetzungen erfüllt.

Das Installationsprogramm von IBM Security Directory Integrator arbeitet mit der Technologie von InstallAnywhere 2012 SP1.

Erforderlicher Plattenspeicherplatz

Unter dem hier aufgeführten Link können Sie den erforderlichen Plattenspeicherplatz überprüfen.

Weitere Informationen finden Sie in der IBM Security Directory Integrator-Dokumentation im Abschnitt zu den Softwarevoraussetzungen.

Speicherbedarf

Unter dem hier aufgeführten Link können Sie die Voraussetzungen für den Speicherbedarf überprüfen.

Weitere Informationen finden Sie in der IBM Security Directory Integrator-Dokumentation im Abschnitt zu den Softwarevoraussetzungen.

Plattformvoraussetzungen

Unter dem hier aufgeführten Link können Sie die Plattformvoraussetzungen überprüfen.

Weitere Informationen finden Sie in der IBM Security Directory Integrator-Dokumentation im Abschnitt zu den Softwarevoraussetzungen.

Komponenten in IBM Security Directory Integrator

Nachstehend erhalten Sie Informationen zu den verfügbaren Komponenten.

Die folgenden Komponenten sind (mit einigen Ausnahmen) im Rahmen der Installation von IBM Security Directory Integrator verfügbar und auswählbar:

Laufzeitserver

Hierbei handelt es sich um eine Regelsteuerkomponente, die zur Implementierung und Ausführung von IBM Security Directory Integrator-Integrationslösungen verwendet wird.

Konfigurationseditor

Hierbei handelt es sich um eine Entwicklungsumgebung für Erstellung, Debug und Erweiterung von IBM Security Directory Integrator-Integrationslösungen.

Anmerkung: Bei den folgenden Betriebssystemen wird der Konfigurationseditor von IBM Security Directory Integrator nicht unterstützt:

- Linux PPC
- Linux 390
- AIX PPC 64

Informationen zur Entwicklung von Lösungen ohne einen lokalen Konfigurationseditor finden Sie unter „Fernen Konfigurationseditor verwenden“ auf Seite 160.

Aktualisierungssite des Konfigurationseditors (Eclipse-Aktualisierungssite für den Konfigurationseditor)

Verwenden Sie den Ordner "Aktualisierungssite des Konfigurationseditors", um den Konfigurationseditor von IBM Security Directory Integrator in einer vorhandenen Eclipse-Installation zu installieren. Verwenden Sie das Eclipse-Tool für den Software-Update sowie diesen Ordner als lokale Aktualisierungssite. Die Aktualisierungssite des Konfigurationseditors wird nur bei einer Implementierung unter Eclipse 3.5.1 oder höher unterstützt.

Anmerkung: Bei den folgenden Betriebssystemen wird die Aktualisierungssite des Konfigurationseditors von IBM Security Directory Integrator nicht unterstützt:

- Linux PPC
- Linux 390
- AIX PPC 64

Informationen zur Entwicklung von Lösungen ohne einen lokalen Konfigurationseditor finden Sie unter „Fernen Konfigurationseditor verwenden“ auf Seite 160.

Java-API-Dokumentation

Dies ist eine vollständige HTML-Dokumentation der IBM Security Directory Integrator-Internas. Sie enthält grundlegendes Referenzmaterial für die Skripterstellung in Lösungen sowie für die Entwicklung von angepassten Komponenten.

Beispiele

Bestimmte IBM Security Directory Integrator-Funktionen oder -Komponenten werden in einer Reihe von kurzen und anschaulichen Beispielkonfigurationen herausgestellt.

Hilfesystem (zur lokalen Bereitstellung der Hilfe zu IBM Security Directory Integrator; in der Standardeinstellung erfolgt dies online)

Anmerkung: Diese Komponente ist veraltet und wird in einer zukünftigen Version von IBM Security Directory Integrator entfernt.

Sie können das IBM Hilfesystem der Benutzerschnittstelle auf der Basis der Eclipse-Technologie (Built on Eclipse), das zuvor "IBM Eclipse Help System" oder auch "IEHS" genannt wurde, lokal installieren und dieses System alternativ zum globalen Onlinehilfeservice nutzen. Bei dieser Option müssen die Hilfedateien von IBM Security Directory Integrator nach der Installation manuell heruntergeladen und implementiert werden.

Falls Ihre Plattform die Systemvoraussetzungen erfüllt, können Sie mit dem Download und den Installationsanweisungen fortfahren, die im Abschnitt „Lokale Hilfedateien installieren“ auf Seite 49 aufgeführt sind.

Integrierte Webplattform (enthält Integrated Solutions Console Standard Edition) Version 8.1.0.3

IBM Security Directory Integrator beinhaltet eine integrierte, einfache Web-Server-Plattform, die manchmal als LWI bezeichnet wird. Diese Serverplattform basiert auf der Architektur von Eclipse und Open Services Gateway Initiative (OSGI); sie unterstützt die Ausführung von Webanwendungen und Web-Services. Die Laufzeit stellt eine sichere Infrastruktur mit geringem Speicherbedarf und einer Minimalkonfiguration bereit. Die integrierte Webplattform enthält Integrated Solution Console Standard Edition (ISC SE), die als Standardalternative für die Implementierung von Administration and Monitoring Console (AMC) in einer vorhandenen ISC-Installation verwendet wird. Die integrierte Webplattform stellt eine einfache, auf OSGI basierende Infrastruktur für das Hosting von Webanwendungen und Web-Services bereit, die folgende Merkmale aufweist:

- Minimale Speicheranforderungen
- Minimalkonfiguration
- Kompatibilität mit ISC auf OSGI-Basis

Anmerkung: Die AMC-Komponente und die integrierte Webplattform sind veraltet und werden in einer zukünftigen Version von IBM Security Directory Integrator entfernt.

Administration and Monitoring Console (AMC)

Dies ist eine browserbasierte Anwendung für die Überwachung und Verwaltung aktiver IBM Security Directory Integrator-Server. AMC wird in Integrated Solutions Console (ISC) ausgeführt. Bei früheren Releases wurde AMC in Form einer Servletanwendung bereitgestellt, die in einer integrierten oder vorhandenen Instanz von IBM WebSphere Application Server (WAS) implementiert wurde.

Anmerkung: Die AMC-Komponente ist veraltet und wird in einer zukünftigen Version von IBM Security Directory Integrator entfernt. IBM WebSphere Application Server unterstützt ISC SE 7.2.0.2 und IBM Dashboard Application Services Hub Version 3.1.

Password Synchronization Plug-ins

Eine in IBM Security Directory Integrator integrierte Lösung, die Kennwörteränderungen auf verschiedenen Systemen abfangen kann.

Die folgenden zusätzlichen Komponenten werden automatisch installiert und können nicht ausgewählt werden:

Java Runtime Environment (JRE) 7.0.4

Dies ist eine Untergruppe von Java Development Kit (JDK), die aus den zentralen ausführbaren Dateien sowie weiteren Dateien besteht, die die Java-Standardplattform bilden. Die JRE umfasst die Java Virtual Machine (JVM), Kernklassen und Unterstützungsdateien.

Anmerkung: Die JRE, die für eines der installierten IBM Security Directory Integrator -Pakete verwendet wird, ist von einer etwaigen systemweiten JRE oder einem JDK, die möglicherweise auf Ihrem System installiert sind, unabhängig. Die JRE wird unabhängig davon installiert, welche Funktionen ausgewählt sind. Das Deinstallationsprogramm benötigt die JRE, weshalb sie in jedem Fall installiert wird.

Sonstige Einstellungen

Diese Komponente enthält das Lizenzpaket, das Deinstallationsprogramm und das Aktualisierungsprogramm.

Das Lizenzpaket von IBM Security Directory Integrator beinhaltet die Lizenzdateien für The IBM Security Directory Integrator.

Weitere Voraussetzungen

Sie müssen sicherstellen, dass Ihr System gewisse weitere Voraussetzungen erfüllt, die nachfolgend beschrieben sind.

Rootberechtigung oder Administratorrechte

Stellen Sie sicher, dass Sie die entsprechenden Anforderungen an Root- oder Administratorberechtigungen erfüllen.

Beachten Sie die folgenden Unterschiede zwischen einer Installation von IBM Security Directory Integrator mit Administratorrechten und einer Installation ohne Administratorrechten:

- Jeder Benutzer, der IBM Security Directory Integrator installiert, muss bei der Installation an der angegebenen Position über Schreibrechte verfügen.
- Benutzer ohne Administratorrechte verfügen über andere Konfigurationseditorverknüpfungen als Benutzer mit Verwaltungsaufgaben.
- Für Benutzer ohne Administratorrechte werden bei der Installation von IBM Security Directory Integrator die Fenster "AMC-Service" und "Server als Service registrieren" nicht angezeigt.
- Nachdem IBM Security Directory Integrator unter Verwendung einer bestimmten Benutzer-ID ohne Rootberechtigung installiert wurde, muss dieselbe Benutzer-ID für jede weitere Pflege dieser Installation (z. B. Deinstallation oder Migration auf neuere Versionen) verwendet werden.

Security Enhanced Linux (SELinux)

Mit den hier aufgeführten Anweisungen können Sie die Einstellungen ändern und SELinux ausführen. Dies trägt zur fehlerfreien Ausführung von SELinux bei.

RedHat Linux (RHEL) ist mit einer Sicherheitsfunktion namens Linux (kurz "SELinux") ausgestattet. SELinux bietet eine Absicherung, die den Host vor bestimmten zerstörerischen Angriffen schützt. In RHEL Version 5.0 ist SELinux standardmäßig aktiviert. Es hat sich herausgestellt, dass die SELinux-Standardinstellungen in RHEL 5.0 eine ordnungsgemäße Ausführung von Java verhindern. Falls Sie versuchen, das IBM Security Directory Integrator-Installationsprogramm für RHEL 5.0 auszuführen, wird möglicherweise ein Fehler angezeigt, der der folgenden Ausgabe ähnelt:

```
# ./install_sdiv72_linux_x86_64.bin  
  
Initializing Wizard.....  
Verifying JVM...  
  
No Java Runtime Environment (JRE) was found on this system.
```

Dieser Fehler beruht darauf, dass die Java Runtime Environment (JRE), die von InstallAnywhere 2012 SP1 in das Verzeichnis /tmp extrahiert wird, nicht die zur Ausführung erforderlichen Berechtigungen besitzt. So können Sie diesen Fehler vermeiden:

1. Inaktivieren Sie SELinux: `setenforce 0`.
2. Führen Sie das IBM Security Directory Integrator-Installationsprogramm aus.
3. Aktivieren Sie SELinux erneut: `setenforce 1`.

Zur Aktivierung bzw. Inaktivierung von SELinux können Sie auch die Konfigurationsdatei `/etc/selinux/config` bearbeiten. Die Standardeinstellungen für die Datei `/etc/selinux/config` sehen in etwa wie die folgenden Zeilen aus:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - SELinux is fully disabled.
SELINUX=enforcing
# SELINUXTYPE= type of policy in use. Possible values are:
# targeted - Only targeted network daemons are protected.
# strict - Full SELinux protection.
SELINUXTYPE=targeted
```

Bei einer Modifizierung von SELINUX entweder in SELINUX=permissive oder in SELINUX=disabled ist die Ausführung des IBM Security Directory Integrator-Installationsprogramms zulässig. Beide Modifizierungen der Eigenschaft SELINUX (entweder in SELINUX=permissive oder in SELINUX=disabled) wirken sich jedoch auf die Sicherheitsstufe für den Host aus.

Das Installationsprogramm von IBM Security Directory Integrator verwendet eine JRE an der Position `installationsverzeichnis/jvm`, die nicht mit den Standardeinstellungen für SELinux ausgeführt werden kann. Das Installationsprogramm bemüht sich, die Probleme mit den Standardeinstellungen für SELinux zu umgehen, indem es versucht, die Sicherheitsberechtigungen für die IBM Security Directory Integrator-JRE zu ändern, die das Installationsprogramm blockieren. Das Installationsprogramm von IBM Security Directory Integrator gibt einen Befehl aus, der die Sicherheitsberechtigungen für die IBM Security Directory Integrator-JRE so ändert, dass diese ausgeführt werden kann. Das IBM Security Directory Integrator-Installationsprogramm gibt hierzu den folgenden Befehl aus:

```
chcon -R -t textrel_shlib_t installationsverzeichnis/jvm/jre
```

Anmerkung: Falls das Installationsprogramm den Befehl `chcon` nicht absetzen kann oder beim Absetzen des Befehls ein Fehler auftritt, müssen Sie die Berechtigungen manuell bearbeiten.

Fehler, die Ähnlichkeit mit der folgenden Ausgabe haben, deuten darauf hin, dass der Befehl `chcon` nicht ordnungsgemäß ausgeführt wurde:

```
[root@dyn9-37-225-164 V7.2]# ./ibmdisrv
Failed to find VM - aborting
```

```
[root@dyn9-37-225-164 V7.2]# ./ibmditk
Failed to find VM - aborting
```

```
[root@dyn9-37-225-164 V7.2]# bin/amc/start_tdiamc.sh
Failed to find VM - aborting
```

Authentifizierung von AMC unter UNIX/Linux

Bei der Arbeit mit AMC gab es bestimmte Einschränkungen für Benutzer ohne Rootberechtigung. Nachstehend erhalten Sie Informationen dazu sowie zu der verfügbaren Ausweidlösung.

Bei einigen UNIX-Plattformen ist AMC in ISC SE selbst dann nicht in der Lage, Benutzer zu authentifizieren, wenn korrekte Berechtigungsnachweise angegeben wurden. Dieses Verhalten tritt auf, wenn AMC mit der ID eines Benutzers ohne Rootberechtigung ausgeführt wird und das Betriebssystem eine Kennwortdatenbank verwendet (beispielsweise eine Datei `/etc/shadow`). Weitere Informationen zu diesem Problem sowie eine Ausweidlösung enthält der Abschnitt zu Authentifizierungsfehlern unter UNIX bei der Ausführung von LWI als Benutzer ohne Rootberechtigung im Handbuch *Fehlerbehebung*.

Grafikpakete für UNIX-Systeme

Mit den hier aufgeführten Anweisungen können Sie den Fehler beheben, der bei der Ausführung des Konfigurationseditors ohne die erforderlichen Grafikpakete generiert wurde.

Wenn die erforderlichen Grafikpakete auf UNIX-Systemen nicht installiert sind, kann später der folgende Fehler auftreten, wenn Sie den Befehl `ibmditk` zum Starten des Konfigurationseditors ausführen:

```
No fonts found; this probably means that the fontconfig library is not correctly configured.
You may need to edit the fonts.conf configuration file.
More information about fontconfig can be found in the fontconfig(3) manual page
and on http://fontconfig.org
```

Führen Sie die folgenden Schritte aus, um solche Fehler zu vermeiden:

1. Stellen Sie sicher, dass die folgenden Grafikpakete auf UNIX-Systemen installiert sind:
 - `libgtk-x11-2.0.so.0`
 - `libgthread-2.0.so.0`
2. Führen Sie den folgenden Befehl aus:

```
export LD_LIBRARY_PATH=/usr/sfw/lib/:usr/lib:/lib
```
3. Installieren Sie die folgende Datei bzw. stellen Sie sicher, dass die folgende Datei installiert ist:

```
/etc/fonts/fonts.conf
```
4. Führen Sie den folgenden Befehl aus:

```
export FONTCONFIG_PATH=/etc/fonts
```

Voraussetzungen für den Konfigurationseditor auf AIX-Betriebssystemen

Mit den hier aufgeführten Anweisungen können Sie die RPMs unter AIX installieren.

Wenn der Konfigurationseditor als Plug-in in Eclipse auf einem AIX-Betriebssystem installiert ist, wird der Editor nicht gestartet und eine Protokolldatei wird erstellt.

Zur Verwendung des Konfigurationseditors müssen die `gtk+`-RPMs und die entsprechenden Abhängigkeiten unter AIX verfügbar sein. Installieren Sie unter AIX die folgenden RPMs:

```
atk-1.12.3-2.aix5.2.ppc.rpm
cairo-1.8.8-1.aix5.2.ppc.rpm
expat-2.0.1-1.aix5.2.ppc.rpm
fontconfig-2.4.2-1.aix5.2.ppc.rpm
freetype2-2.3.9-1.aix5.2.ppc.rpm
gettext-0.10.40-6.aix5.1.ppc.rpm
glib2-2.12.4-2.aix5.2.ppc.rpm
gtk2-2.10.6-4.aix5.2.ppc.rpm
libjpeg-6b-6.aix5.1.ppc.rpm
libpng-1.2.32-2.aix5.2.ppc.rpm
libtiff-3.8.2-1.aix5.2.ppc.rpm
pango-1.14.5-4.aix5.2.ppc.rpm
pixman-0.12.0-3.aix5.2.ppc.rpm
xcursor-1.1.7-3.aix5.2.ppc.rpm
xft-2.1.6-5.aix5.1.ppc.rpm
xrender-0.9.1-3.aix5.2.ppc.rpm
zlib-1.2.3-3.aix5.1.ppc.rpm
```

Anmerkung: Die installierten RPMs müssen die hier aufgeführten Versionen aufweisen, da frühere oder spätere Versionen möglicherweise nicht kompatibel sind.

Führen Sie die folgenden Schritte aus, um diese RPM-Versionen zu installieren:

1. Laden Sie die RPMs in ein neues Verzeichnis herunter. Sie finden die RPMs unter der folgenden Adresse: <ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/>.
2. Installieren Sie die heruntergeladenen RPMs mithilfe des folgenden Befehls. Wenn bereits eine Version einer RPM installiert ist, führt der Befehl ein Upgrade bzw. ein Downgrade auf die heruntergeladene Version durch.

```
rpm -U *.rpm --force
```
3. Stellen Sie sicher, dass die Umgebungsvariable LIBPATH einen Pfad zum Abschluss der Bibliotheken enthält. Beispiel: LIBPATH=/opt/freeware/64/lib/.

Voraussetzungen für ein Upgrade von Version 7.1.1 auf Version 7.2 auf Windows 2012-Betriebssystemen

Nachstehend erhalten Sie Informationen zu den Voraussetzungen für Versionsupdates.

Wenn Sie die Durchführung eines Upgrades von IBM Security Directory Integrator Version 7.1.1 auf Version 7.2 auf einem Windows 2012-Betriebssystem planen, müssen Sie vor dem Start des Installationsprogramms für Version 7.2 sicherstellen, dass der **Windows 7-Kompatibilitätsmodus** für die Datei "uninstaller.exe" von Version 7.1.1 aktiviert ist. Weitere Informationen finden Sie in den technischen Hinweisen unter <http://www-01.ibm.com/support/docview.wss?uid=swg21634336>.

IBM Security Directory Integrator installieren

Mit dem Installationsprogramm können Sie IBM Security Directory Integrator entweder vollständig installieren, ausschließlich die benötigten IBM Security Directory Integrator-Komponenten installieren, ein Upgrade für eine frühere Version (bei den Versionen 7.0, 7.1 und 7.1.1) durchführen oder Komponenten zu einer vorhandenen IBM Security Directory Integrator-Installation hinzufügen.

Anmerkung:

- Ein direktes Upgrade von IBM Security Directory Integrator von Version 6.x oder einer älteren Version auf Version 7.2 wird nicht unterstützt. Sie müssen zunächst ein Upgrade von Version 6.x auf Version 7.1.1 und anschließend von Version 7.1.1 auf Version 7.2 durchführen.
- Bei den folgenden Betriebssystemen wird der Konfigurationseditor von IBM Security Directory Integrator nicht unterstützt:
 - Linux PPC
 - Linux 390
 - AIX (64-Bit)

Weitere Informationen zur Verwendung des Produkts mit einem lokal installierten Konfigurationseditor finden Sie in „Fernes Konfigurationseditor verwenden“ auf Seite 160.

Bei einem Upgrade von einer früheren Version deinstalliert IBM Security Directory Integrator die frühere Version. Vom Benutzer erstellte Dateien werden durch die Deinstallation jedoch nicht entfernt. Nach Abschluss der Neuinstallation sind die vom Benutzer erstellten Dateien weiterhin verfügbar. Konfigurationsdateien wie `global.properties` und `am_config.properties` werden auf IBM Security Directory Integrator 7.2 migriert, wodurch eventuell vorgenommene benutzerdefinierte Änderungen an der Konfiguration erhalten bleiben.

Anmerkung: Obwohl das Installationsprogramm von IBM Security Directory Integrator einige der vordefinierten Konfigurations- und Eigenschaftendateien sichert

und wiederherstellt, ist es ein bewährtes Verfahren, die eigenen Dateien und Datenbanken, die kritische Daten enthalten, vor dem Start der Installation manuell zu sichern.

Die IBM Security Directory Integrator 7.2-Installation verwendet weiterhin die Komponenten, die in vorherigen Versionen von IBM Security Directory Integrator verfügbar waren:

- Administration and Monitoring Console (AMC).
- Konfigurationseditor
- Beispiele
- IBM Hilfesystem der Benutzerschnittstelle auf der Basis der Eclipse-Technologie (Built on Eclipse)
- Java-API-Dokumentation
- Laufzeitserver

Anmerkung: Im weiteren Verlauf der vorliegenden Veröffentlichung *Installation und Verwaltung* bezeichnet die Variable *tdi-installationsverzeichnis* die Installationsverzeichnisposition, die vom Benutzer während der Installation in der **Anzeige für die Zielposition** ausgewählt wurde. Im Abschnitt „Standardinstallationspositionen“ auf Seite 57 können Sie nachlesen, an welchen Positionen IBM Security Directory Integrator normalerweise installiert wird.

Geeignetes Installationsprogramm starten

Sie können das Installationsprogramm über das Launchpad oder durch eine direkte Installation starten. Nachstehend erhalten Sie detaillierte Informationen zu den Schritten zum Starten.

Zum Starten des IBM Security Directory Integrator-Installationsprogramms können Sie eines der folgenden Verfahren verwenden:

Installationsprogramm über Launchpad starten

Das IBM Security Directory Integrator-Launchpad bietet grundlegende Einführungsinformationen für die Installation sowie Links zu detaillierteren Angaben über verschiedenste Themen bezüglich Installation, Migration und Installationsabschluss. Außerdem können Sie über das Launchpad das IBM Security Directory Integrator-Installationsprogramm starten.

Anmerkung: Damit das Launchpad verwendet werden kann, muss ein unterstützter Web-Browser installiert und konfiguriert sein. Ist dies nicht der Fall, können Sie das Launchpad nicht verwenden. Sie können jedoch das plattformspezifische Installationsprogramm auch dann direkt verwenden. Anweisungen zur Verwendung des Installationsprogramms von IBM Security Directory Integrator finden Sie unter „Plattformspezifisches Installationsprogramm von IBM Security Directory Integrator verwenden“ auf Seite 13.

1. Öffnen Sie das IBM Security Directory Integrator-Launchpad, indem Sie den folgenden Befehl an der Eingabeaufforderung eingeben:

- Bei Windows-Plattformen:

```
Launchpad.bat
```

- Bei allen anderen Plattformen:

```
Launchpad.sh
```

Über das Menü auf der linken Seite des Launchpads können Sie in den Launchpadfenstern navigieren. Klicken Sie auf eine Menüoption, um Informationen zu dieser Option anzuzeigen.

Anmerkung zu den Abbildungen: Aus technischen Gründen können die im vorliegenden Handbuch dargestellten Abbildungen von der jeweils verwendeten Benutzerumgebung abweichen oder ggf. nur in englischer Sprache verfügbar sein. Folgende Menüoptionen stehen zur Verfügung:

Willkommen!

Das Fenster "Willkommen!" der Installation enthält Links zu den folgenden Zielen:

- Website von IBM Security Directory Integrator
- Dokumentation der Version IBM Security Directory Integrator
- Unterstützungswebsite
- IBM Security Directory Integrator-Newsgroup

Screenshotname: Adminst-1



Bei den folgenden Optionen auf der linken Seite handelt es sich um Fenster des IBM Security Directory Integrator-Launchpads:

Releaseinformationen

Dieses Fenster enthält eine Liste mit einigen neuen und verbesserten Komponenten, die in diesem Release verfügbar sind, sowie Links zur Dokumentation über das Release.

Erforderliche Informationen

Dieses Fenster enthält Links zu Angaben über Plattformunterstützung und Hardwarevoraussetzungen.

Installationsszenarios

Dieses Fenster enthält eine Beschreibung der IBM Security Directory Integrator-Komponenten, die zur Installation verfügbar

sind. Während der Installation können Sie einige oder alle dieser Komponenten installieren. Das Fenster enthält außerdem eine Beschreibung der Komponenten "Password Synchronisation Plug-ins", die zur Installation verfügbar sind.

Migrationsinformationen

Dieses Fenster enthält einen Link zu Informationen zur Migration von IBM Security Directory Integrator 7.0, 7.1 oder 7.1.1 auf Version 7.2. Außerdem enthält es Angaben über die Migration des Derby-Systemspeichers.

Anmerkung: Ein direktes Upgrade von IBM Security Directory Integrator von Version 6.x oder einer älteren Version auf Version 7.2 wird nicht unterstützt. Sie müssen zunächst ein Upgrade von Version 6.x auf Version 7.1.1 und anschließend von Version 7.1.1 auf Version 7.2 durchführen.

IBM Security Directory Integrator installieren

Dieses Fenster enthält Links zum IBM Security Directory Integrator-Installationsprogramm sowie Links zur Dokumentation für die Installation, die Migration und die unterstützten Plattformen. Anweisungen zur Verwendung des IBM Security Directory Integrator-Installationsprogramms enthält der Abschnitt „Plattformspezifisches Installationsprogramm von IBM Security Directory Integrator verwenden“ auf Seite 13.

IBM Security Directory Integrator Password Synchronization Plug-ins installieren

Dieses Fenster enthält Links zum Installationsprogramm für die IBM Security Directory Integrator-Komponente "Password Synchronizer Plug-ins" sowie Links zur Dokumentation für die Installation und die unterstützten Plattformen.

Anmerkung: Auf Linux PPC- und Linux 390-Plattformen ist dieses Fenster nicht verfügbar.

Beenden

Beendet das Launchpad, ohne eine Installation auszuführen.

2. Klicken Sie im Installationsfenster auf die Option **Installationsprogramm** für IBM Security Directory Integrator. Hierdurch wird das Installationsprogramm gestartet. Anweisungen zur Verwendung des Installationsprogramms enthält der Abschnitt „Plattformspezifisches Installationsprogramm von IBM Security Directory Integrator verwenden“ auf Seite 13.

Installationsprogramm direkt starten

Sie können das Installationsprogramm direkt starten, indem Sie die ausführbare Datei für die Installation verwenden:

1. Suchen Sie im Verzeichnis tdi_installer auf der Produkt-CD nach der ausführbaren Datei für die Installation für Ihre Plattform.

Windows Intel

install_sdiv72_win_x86.exe

Windows (64-Bit)

install_sdiv72_win_x86_64.exe

AIX install_sdiv72_aix_ppc.bin

AIX (64-Bit)

install_sdiv72_aix_ppc_64.bin

Linux (64-Bit)

install_sdiv72_linux_x86_64.bin

Power PC Linux

install_sdiv72_ppclinux.bin

Solaris Sparc

install_sdiv72_solaris_sparc.bin

Solaris (Intel)

install_sdiv72_solaris_x86_64.bin

2. Doppelklicken Sie auf die ausführbare Datei oder geben Sie an der Eingabeaufforderung den Namen der ausführbaren Datei ein. Hierdurch wird das Installationsprogramm gestartet. Informationen zur Verwendung des Installationsprogramms enthält der Abschnitt „Plattformspezifisches Installationsprogramm von IBM Security Directory Integrator verwenden“.

Nachdem Sie das Installationsprogramm gestartet haben, können Sie mit dem im Abschnitt „Plattformspezifisches Installationsprogramm von IBM Security Directory Integrator verwenden“ beschriebenen Prozess beginnen.

Plattformspezifisches Installationsprogramm von IBM Security Directory Integrator verwenden

Mit den hier aufgeführten Anweisungen können Sie eine plattformspezifische Instanz von IBM Security Directory Integrator installieren.

Das plattformspezifische Installationsprogramm von IBM Security Directory Integrator wird entweder über das Launchpad oder über die Befehlszeile gestartet. Mit dem IBM Security Directory Integrator-Installationsprogramm kann eine neue IBM Security Directory Integrator-Kopie installiert, eine Komponente zu einer vorhandenen IBM Security Directory Integrator-Instanz hinzugefügt oder ein Upgrade ausgehend von einer vorherigen IBM Security Directory Integrator-Version durchgeführt werden. Die Standardinstallationsposition auf Ihrem Computer für IBM Security Directory Integrator variiert je nach Plattform.

Während der Installation protokolliert das Installationsprogramm seine Aktionen in Dateien (sdiv72install.log und sdiv72debug.log), die sich im Verzeichnis des Systems für temporäre Dateien befinden (normalerweise /tmp bzw. /var/tmp bei UNIX-Plattformen).

Installation mit dem grafisch orientierten Installationsprogramm ausführen

Mit den hier aufgeführten Anweisungen können Sie IBM Security Directory Integrator mithilfe des grafisch orientierten Installationsprogramms installieren.

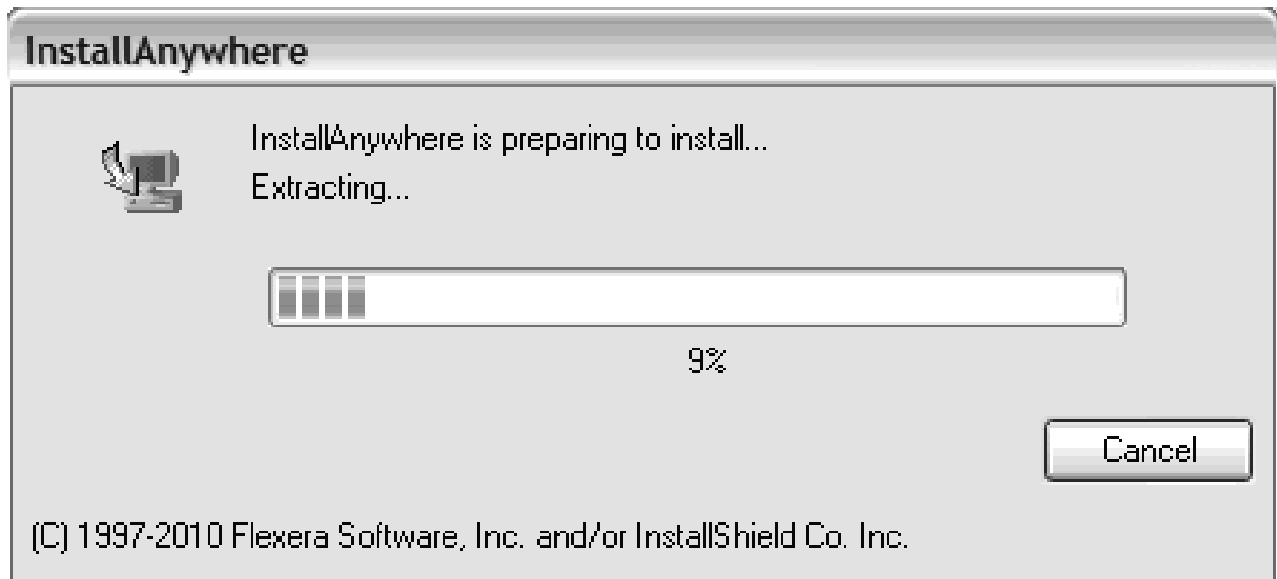
Anzeigenfolge bei der Installation

Mit den hier aufgeführten Anweisungen können Sie die Anzeigenfolge installieren.

Anzeige für Initialisierungsvorbereitung

Sie rufen die ausführbare Datei des Installationsprogramms entweder über die Befehlszeile oder (nur bei Windows) durch Doppelklicken auf die aus-

föhrbare Datei auf. Zunachst wird die folgende Anzeige aufgerufen, auf die eine Begrüßungsanzeige folgt:



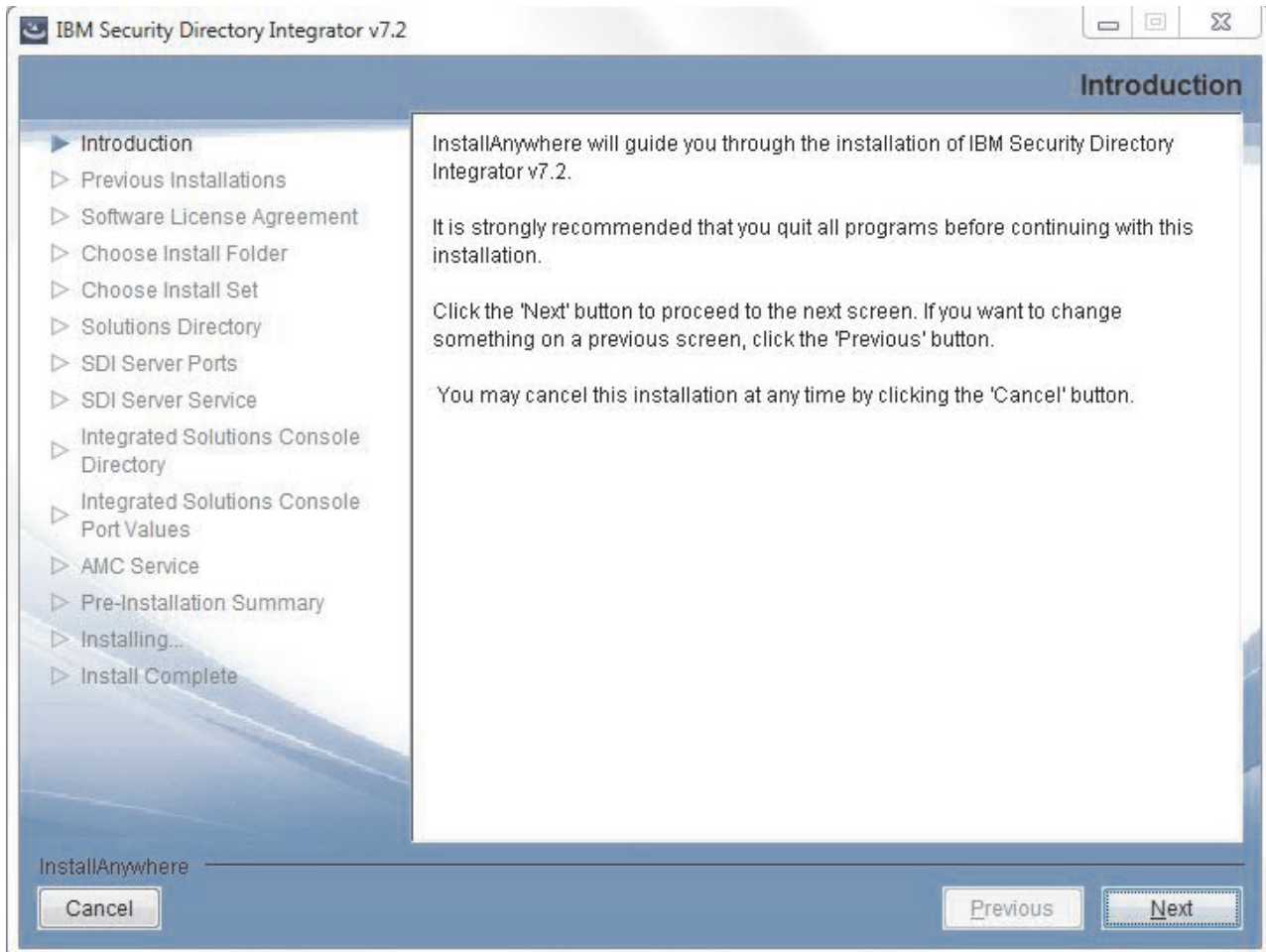
Anmerkung: Die Begrüßungsanzeige enthält möglicherweise auch eine Dropdown-Liste mit Auswahlmöglichkeiten für die Sprache, falls das zugrunde liegende System mehrere Sprachen unterstützt. (Die Standardeinstellung ist Englisch.)





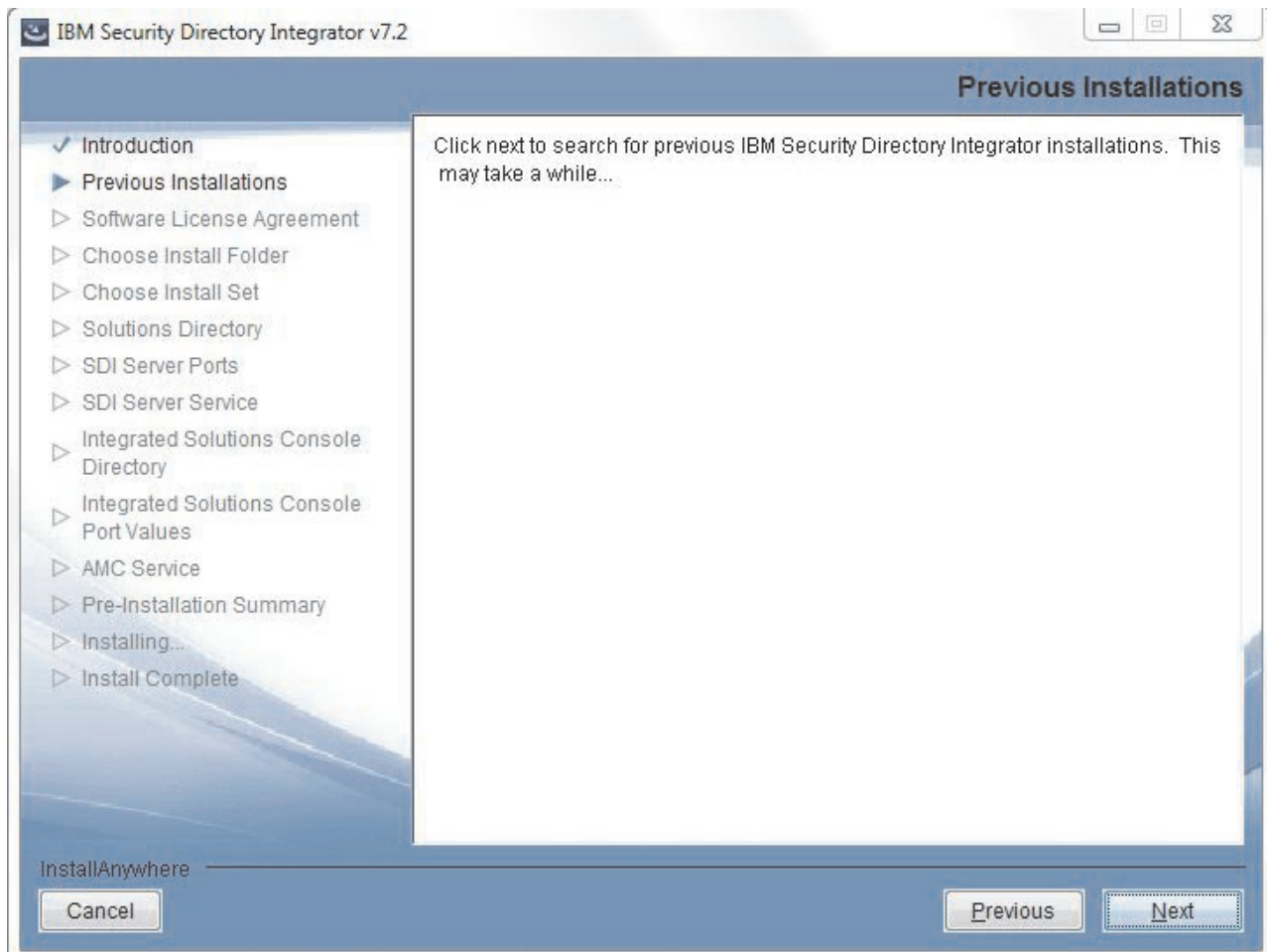
Einführungsanzeige

Dies ist die Eingangsanzeige für das Installationsprogramm. Hierbei handelt es sich um die Standardanzeige, die durch das InstallAnywhere-Installationsprogramm bereitgestellt wird. In dieser Anzeige können Sie die Installation durch Auswahl der Schaltfläche **Weiter** fortsetzen oder das Installationsprogramm durch Auswahl der Schaltfläche **Abbrechen** verlassen.



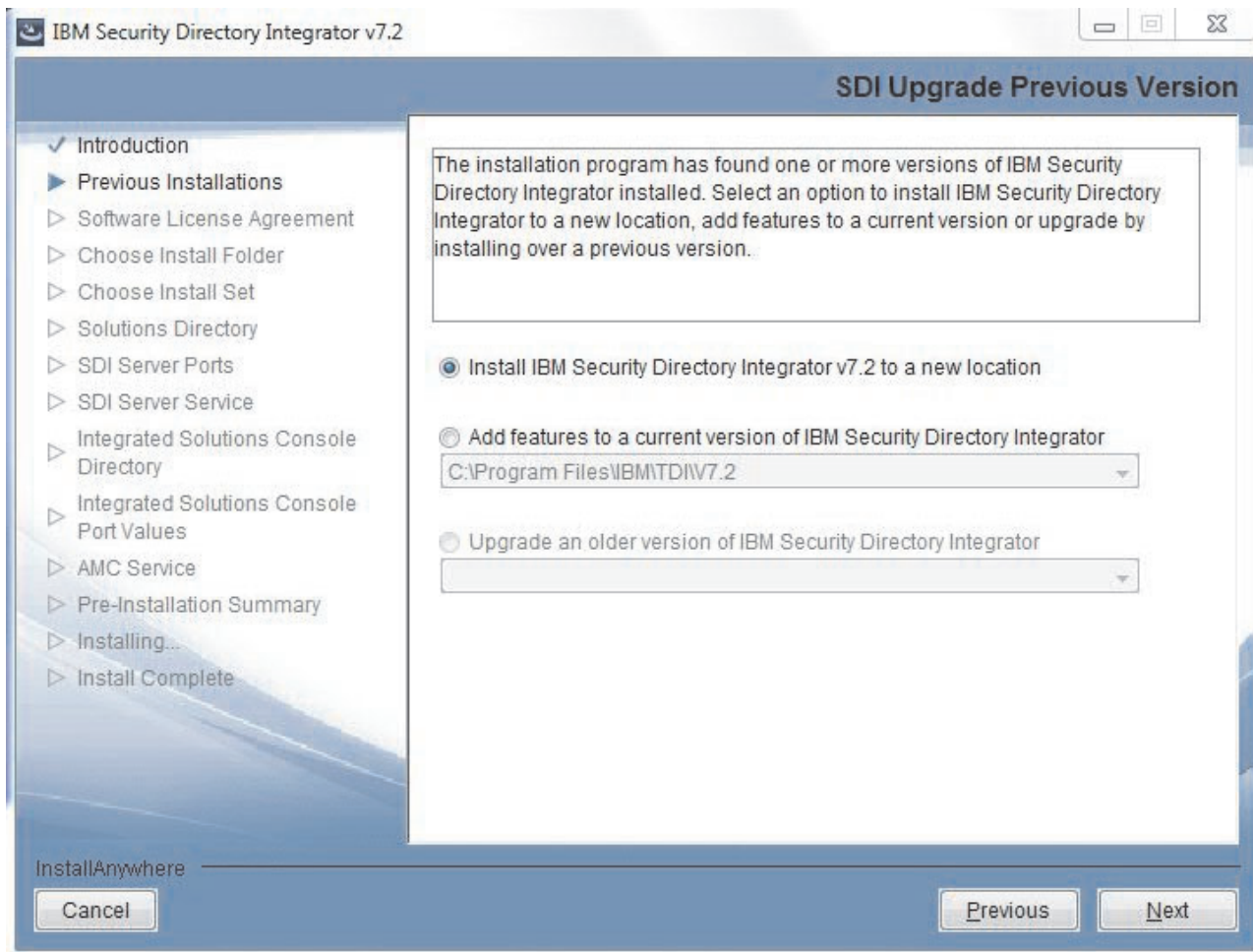
Anzeige "Vorhergehende Installationen"

In dieser Anzeige werden Sie darüber informiert, dass es einige Zeit dauern kann, bis Vorversionen von IBM Security Directory Integrator erkannt wurden.



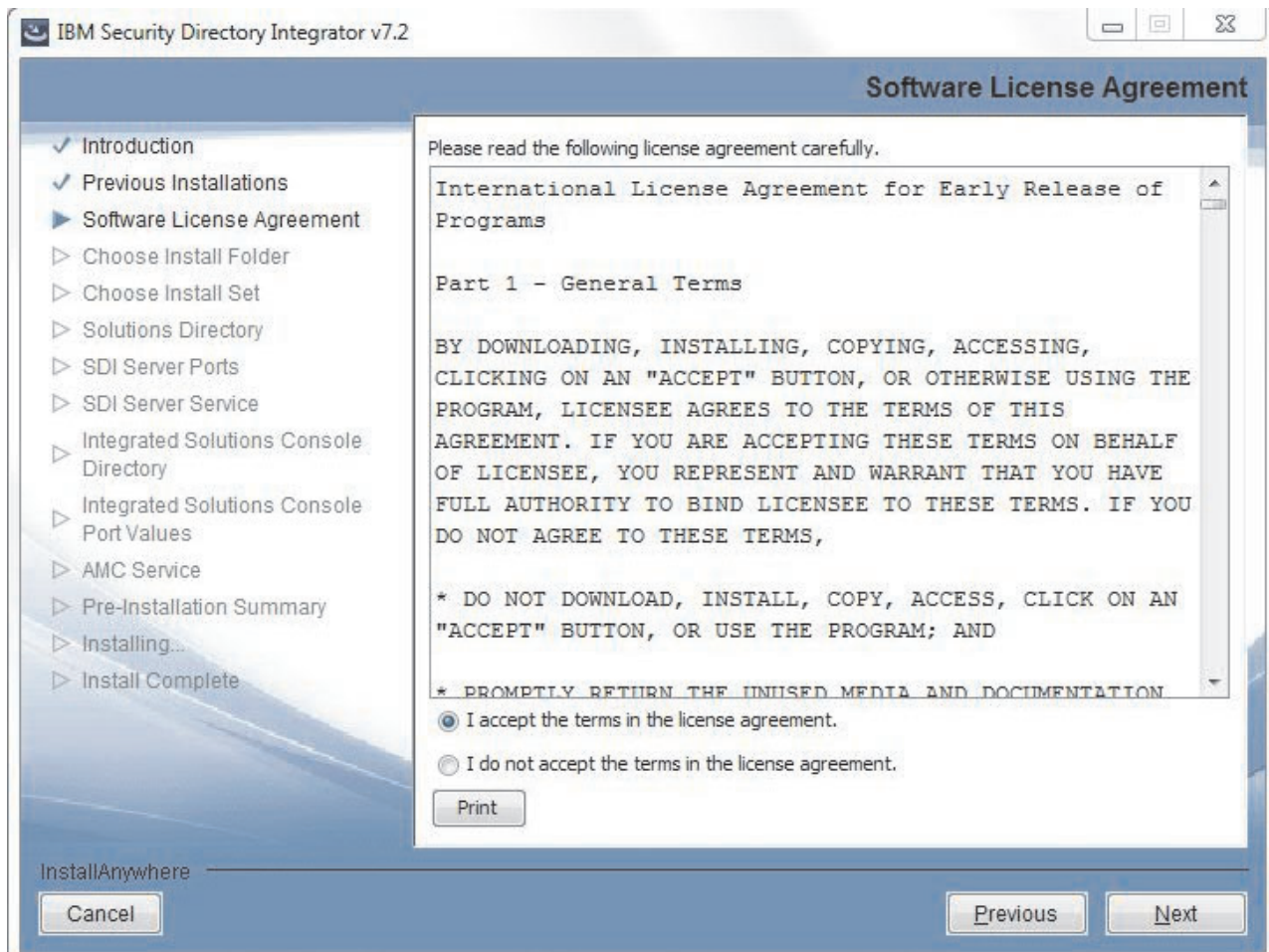
Falls eine Vorversion gefunden wird, erhalten Sie eine Reihe von Upgradeoptionen.

Anmerkung: Ein direktes Upgrade von IBM Security Directory Integrator von Version 6.x oder einer älteren Version auf Version 7.2 wird nicht unterstützt. Sie müssen zunächst ein Upgrade von Version 6.x auf Version 7.1.1 und anschließend von Version 7.1.1 auf Version 7.2 durchführen.



Anzeige "Softwarelizenzvereinbarung"

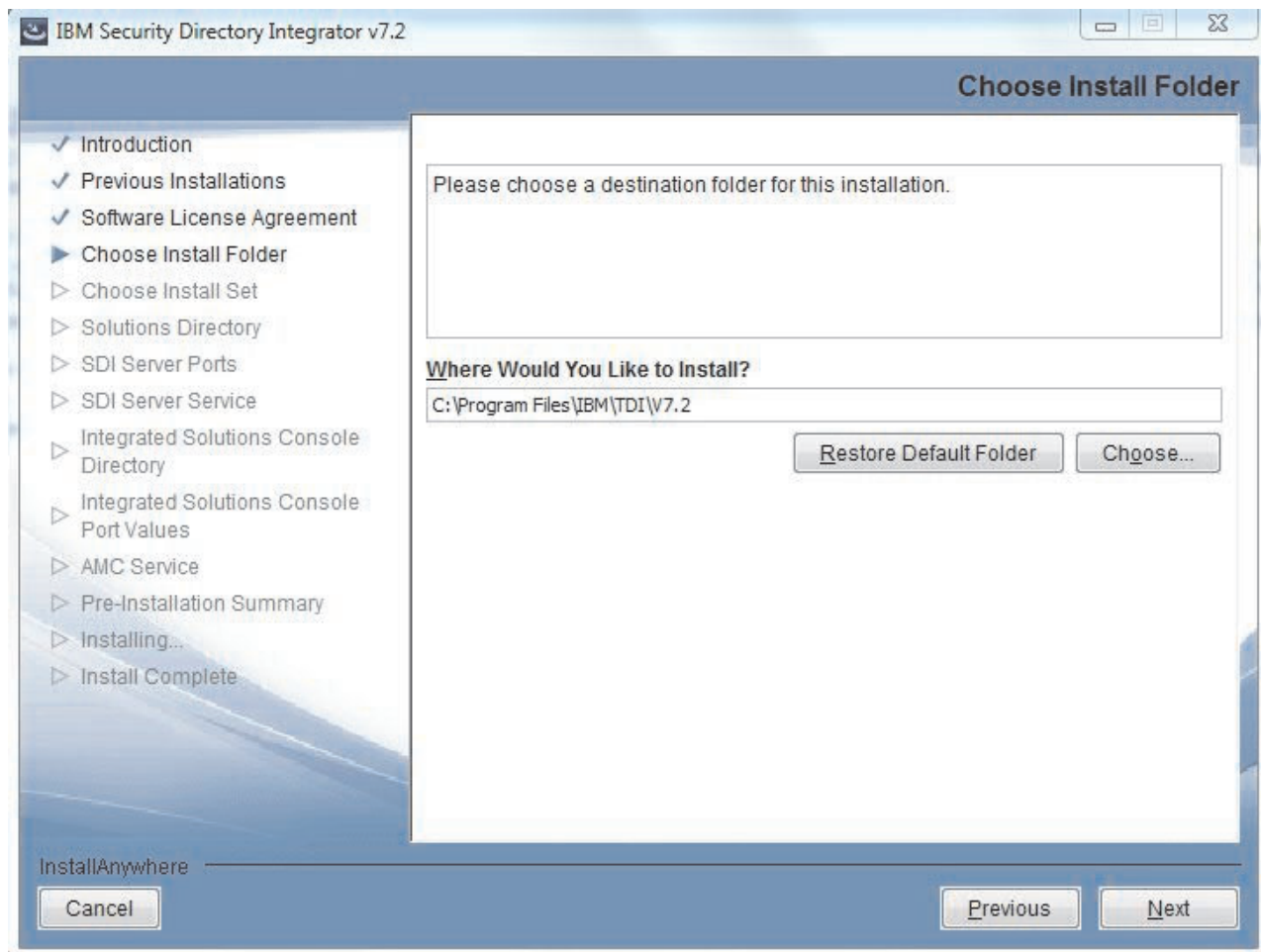
Die Lizenzanzeige wird durch das IBM Lizenztool bereitgestellt. Diese Anzeige wird bei einer **Neuinstallation von Security Directory Integrator Version 7.2** sowie beim **Upgrade einer älteren Security Directory Integrator-Version** angezeigt.



Anzeige "Installationsordner auswählen"

Anmerkung:

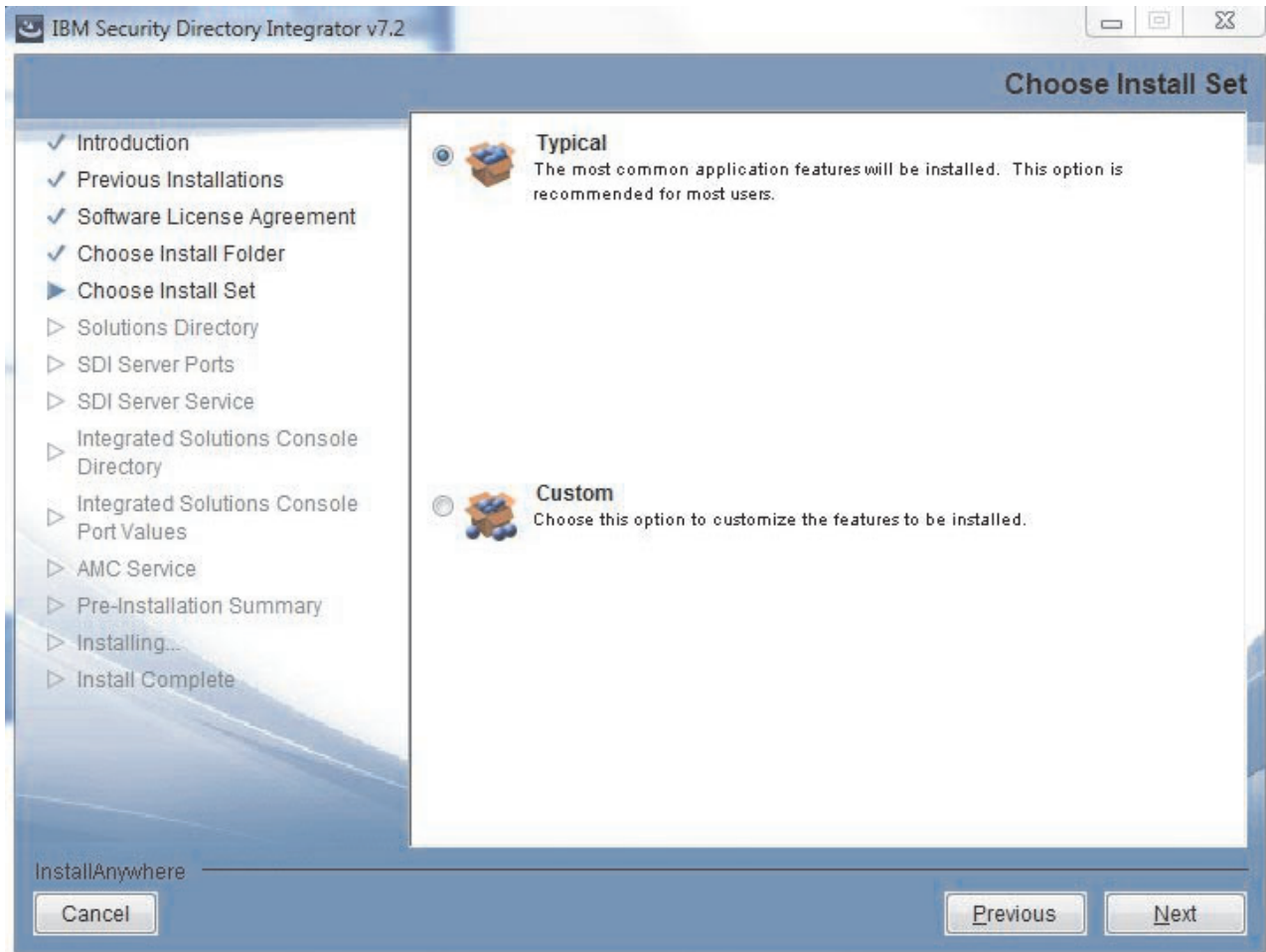
1. Diese Anzeige wird bei einem von IBM Security Directory Integrator Version 7.0, 7.1 oder 7.1.1 ausgehenden Upgrade nicht angezeigt. Sie wird ebenfalls nicht aufgerufen, wenn Sie Komponenten zu einer vorhandenen Instanz von IBM Security Directory Integrator Version 7.2 hinzufügen.
2. Die Anzeige für die Zielposition enthält den zuletzt eingegebenen Wert, falls Sie mit den weiteren Anzeigen im Assistenten fortfahren und später zu dieser Anzeige zurückkehren.
3. Im Installationspfad werden Zeichen, die nicht zum ASCII-Zeichensatz gehören, sowie die folgenden Zeichen nicht unterstützt:
";|*?!#&\$',=^@%+



Anzeige "Installationsgruppe auswählen"

Eine Installation des Typs "Standard" beinhaltet den Laufzeitserver, den Konfigurationseditor, die Javadocs, die Beispiele und AMC. Die Aktualisierungssite des Konfigurationseditors, das IBM Hilfesystem der Benutzerschnittstelle auf der Basis der Eclipse-Technologie (Built on Eclipse) sowie die Komponente "Password Synchronization Plug-ins" sind in dieser Installation nicht enthalten.

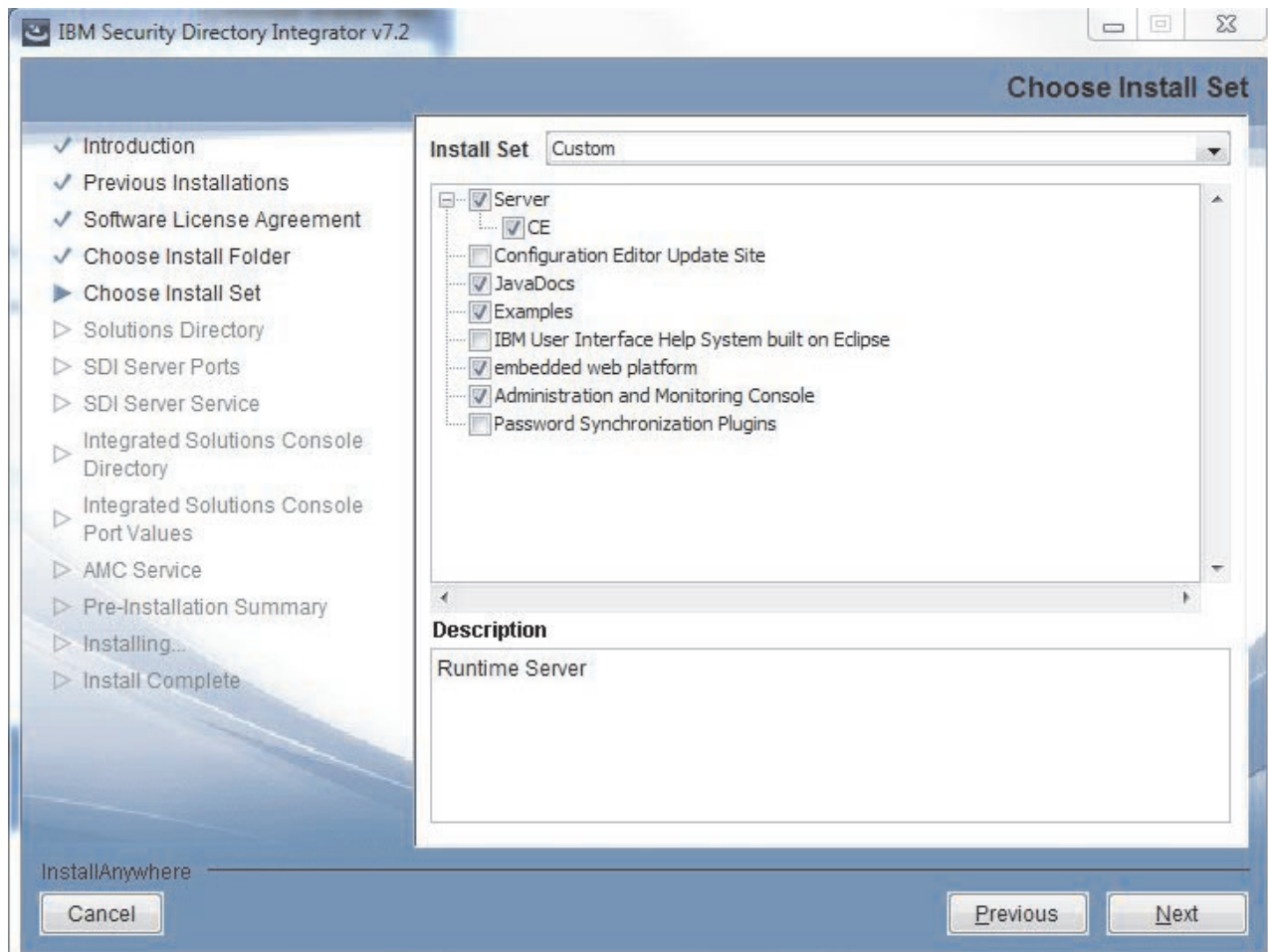
Falls Sie die Option **Standard** auswählen, wird die Anzeige für die Komponentenauswahl übersprungen. Außerdem werden die Produktpaketkomponenten für die integrierte Webplattform und ISC automatisch installiert. Die Anzeige für das ISC-Verzeichnis wird übersprungen.



Anzeige für Komponentenauswahl

In dieser Anzeige können Sie angeben, welche Komponenten installiert werden sollen. Bei Bedarf kann jede Komponente einzeln installiert werden. Die einzige Ausnahme besteht darin, dass bei Auswahl des Konfigurationseeditors der Server ausgewählt wird, weil es sich beim Konfigurationseeditor um eine Unterkomponente des Servers handelt.

Falls eine Komponente auf der jeweiligen Plattform nicht unterstützt wird, ist sie in der Anzeige für die Komponentenauswahl nicht aufgeführt.



Die folgende Liste vermittelt einen Überblick über die einzelnen Komponenten:

Laufzeitserver

Hierbei handelt es sich um eine Regelsteuerkomponente, die zur Implementierung und Ausführung von IBM Security Directory Integrator-Integrationslösungen verwendet wird.

Konfigurationseditor

Hierbei handelt es sich um eine Entwicklungsumgebung für Erstellung, Debug und Erweiterung von IBM Security Directory Integrator-Integrationslösungen. Ohne eine Installation des Laufzeitserver kann diese Komponente nicht installiert werden.

Aktualisierungssite des Konfigurationseditors

Als Muster für diese Komponente diente die Eclipse-Aktualisierungssite. Sie enthält die Dateien, die zur Installation des Konfigurationseditors in einer vorhandenen Eclipse-Installation benötigt werden, und wird außerdem zur Pflege eingesetzt. (Ist unter zLinux oder Linux PPC nicht verfügbar.)

Javadocs

Dies ist eine vollständige HTML-Dokumentation der IBM Security Directory Integrator-Interna. Sie enthält grundlegendes Referenzmaterial für die Scripterstellung in Lösungen sowie für die Entwicklung von angepassten Komponenten.

Beispiele

Bestimmte IBM Security Directory Integrator-Funktionen oder -Komponenten werden in einer Reihe von kurzen und anschaulichen Beispielkonfigurationen herausgestellt.

IBM Hilfesystem der Benutzerschnittstelle auf der Basis der Eclipse-Technologie (lokale Hilfe)

Sie können das IBM Hilfesystem der Benutzerschnittstelle auf der Basis der Eclipse-Technologie (Built on Eclipse), das zuvor "IEHS" genannt wurde, lokal installieren und als Alternative zum globalen Onlinehilfeservice verwenden. Bei dieser Option müssen die Hilfe-dateien von IBM Security Directory Integrator nach der Installation manuell heruntergeladen und implementiert werden.

Integrierte Webplattform

Das Paket mit der integrierten Webplattform (enthält ISC SE).

Administration and Monitoring Console

Dies ist eine browserbasierte Anwendung für die Überwachung und Verwaltung aktiver IBM Security Directory Integrator-Server.

Password Synchronization Plug-ins

Hierbei handelt es sich um die IBM Security Directory Integrator-Plug-ins zur Kennwortsynchronisation.

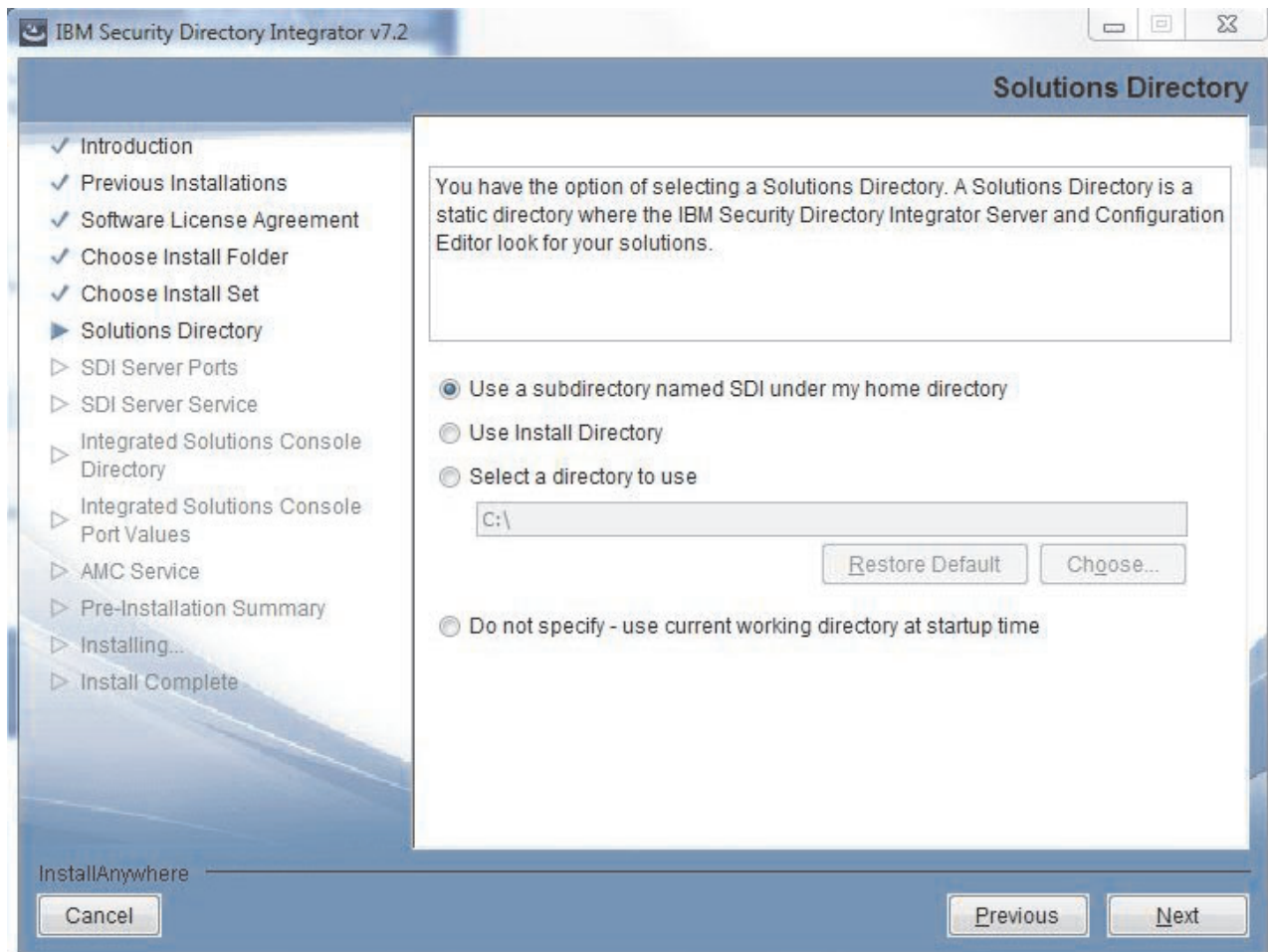
Anzeige für IBM Security Directory Integrator-Lösungsverzeichnis

Diese Anzeige wird nur bei Auswahl der Serverkomponente ausgegeben. Sie können dort das Standardlösungsverzeichnis für den Server auswählen. Das Lösungsverzeichnis ist ein statisches Verzeichnis, in dem die vom Benutzer erstellten und auszuführenden Lösungen enthalten sind. In dieser Anzeige ist standardmäßig die Option ausgewählt, mit der das Ausgangsverzeichnis des Benutzers als Lösungsverzeichnis festgelegt wird.

Bei Auswahl des Optionsfelds **Zu verwendendes Verzeichnis auswählen** müssen Sie ein gültiges Lösungsverzeichnis angeben. Der UNC-Pfad (Universal Naming Convention, allgemeine Namenskonvention) wird bei der Installation für das Lösungsverzeichnis unterstützt.

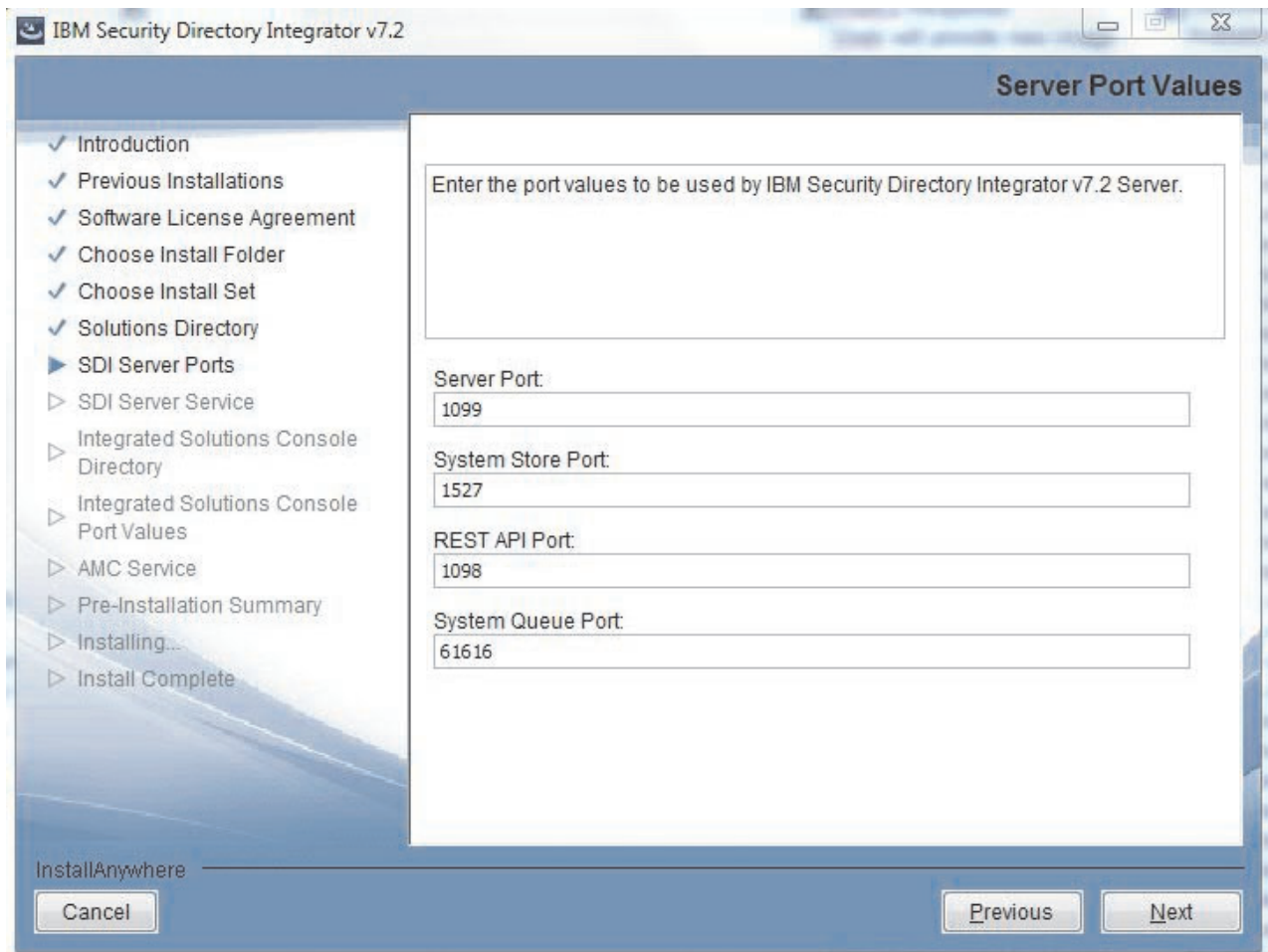
Anmerkung: Diese Anzeige wird bei einem von IBM Security Directory Integrator Version 7.0, 7.1 oder 7.1.1 ausgehenden Upgrade nicht aufgerufen.

Falls Sie Komponenten hinzufügen wollen und die Serverkomponente bereits installiert ist, wird diese Anzeige nicht ausgegeben.



Anzeige für Server-Port-Konfiguration

In dieser Anzeige werden von Ihnen 4 Server-Portnummern angefordert. Für diese Ports sind bereits Standardwerte angegeben. Das Installationsprogramm prüft, ob Sie eine gültige und verfügbare Portnummer angeben (siehe Server-Port-Konfiguration).



Anzeige für Registrierung des Servers als Systemservice

Diese Anzeige wird nur dann aufgerufen, wenn eine neue Instanz von IBM Security Directory Integrator installiert werden soll und die Installation des Servers als Komponente ausgewählt wurde oder wenn es sich um eine Upgrade-Installation handelt. Außerdem wird sie nur dann angezeigt, wenn Sie Administratorrechte besitzen.

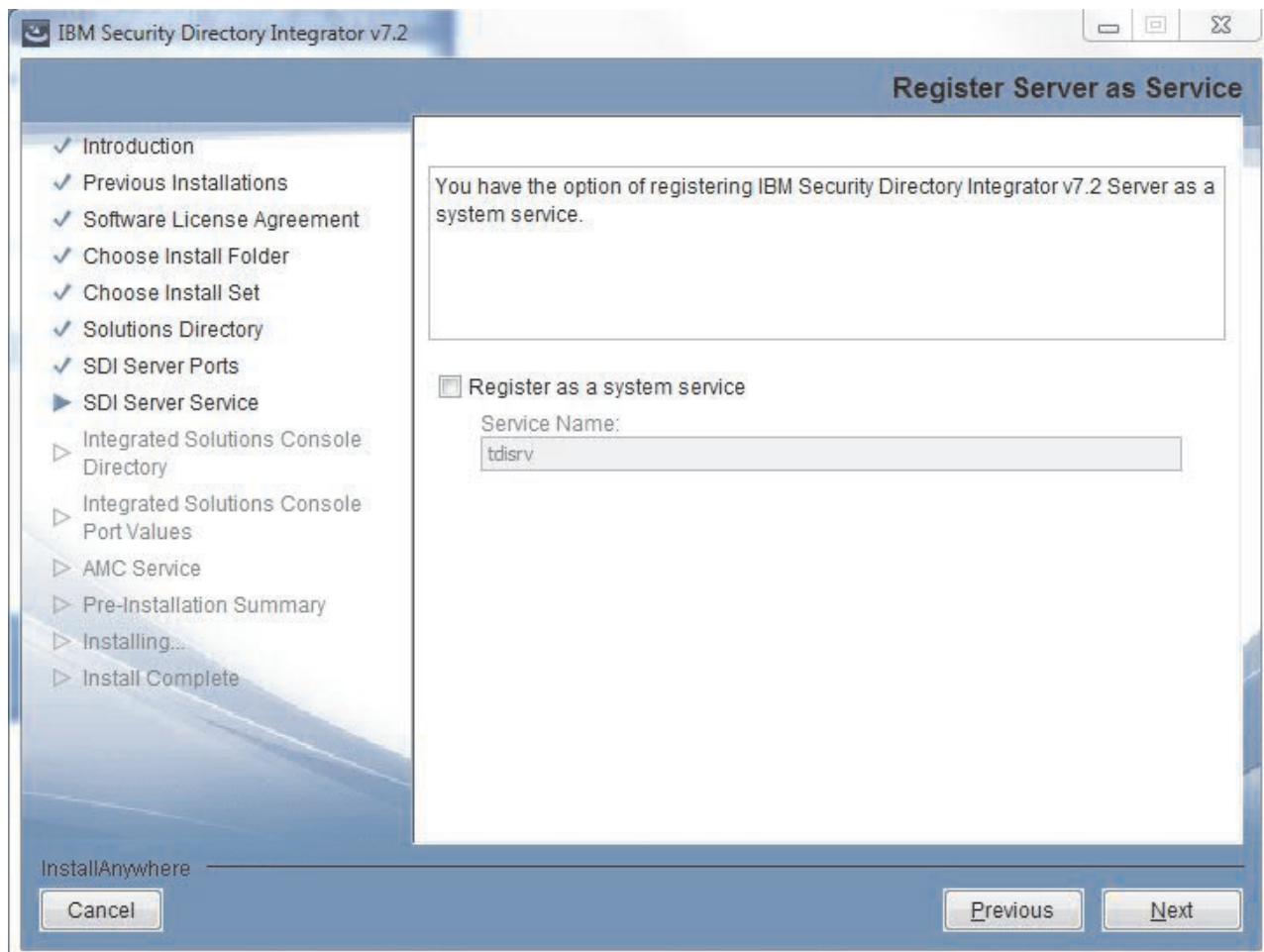
Falls das Markierungsfeld ausgewählt ist, wird nur der SERVER als Service für dieses Betriebssystem registriert.

In der Standardeinstellung ist das Markierungsfeld abgewählt. Die beiden Textfelder werden nur bei Auswahl des Markierungsfelds aktiviert. Das erste Textfeld ist für den Servicenamen vorgesehen. Das zweite Textfeld enthält die Portnummer, die der Server zur Ausführung als Systemservice verwendet.

Das Installationsprogramm versucht nach Möglichkeit, einen gültigen Standardwert für den Servicenamen bereitzustellen (Details zu diesem Prozess können Sie den Informationen zur Registrierung des Servers als Windows-Dienst oder als UNIX-Prozess entnehmen). Falls das Installationsprogramm keinen gültigen Servicenamen bestimmen kann, ist das Feld leer. Sie können erst dann mit der Installation fortfahren, nachdem Sie einen gültigen Servicenamen eingegeben haben.

Anmerkung: Wenn Sie die Installation auf einem UNIX-System durchführen, stellen Sie sicher, dass der Servicenamen die maximale Länge von vier

Zeichen nicht überschreitet. Wenn er vier Zeichen überschreitet, führt diese Einschränkung zu einem Fehler und die Installation kann nicht fortgesetzt werden.



Anzeige für IBM Security Directory Integrator-AMC-Implementierung

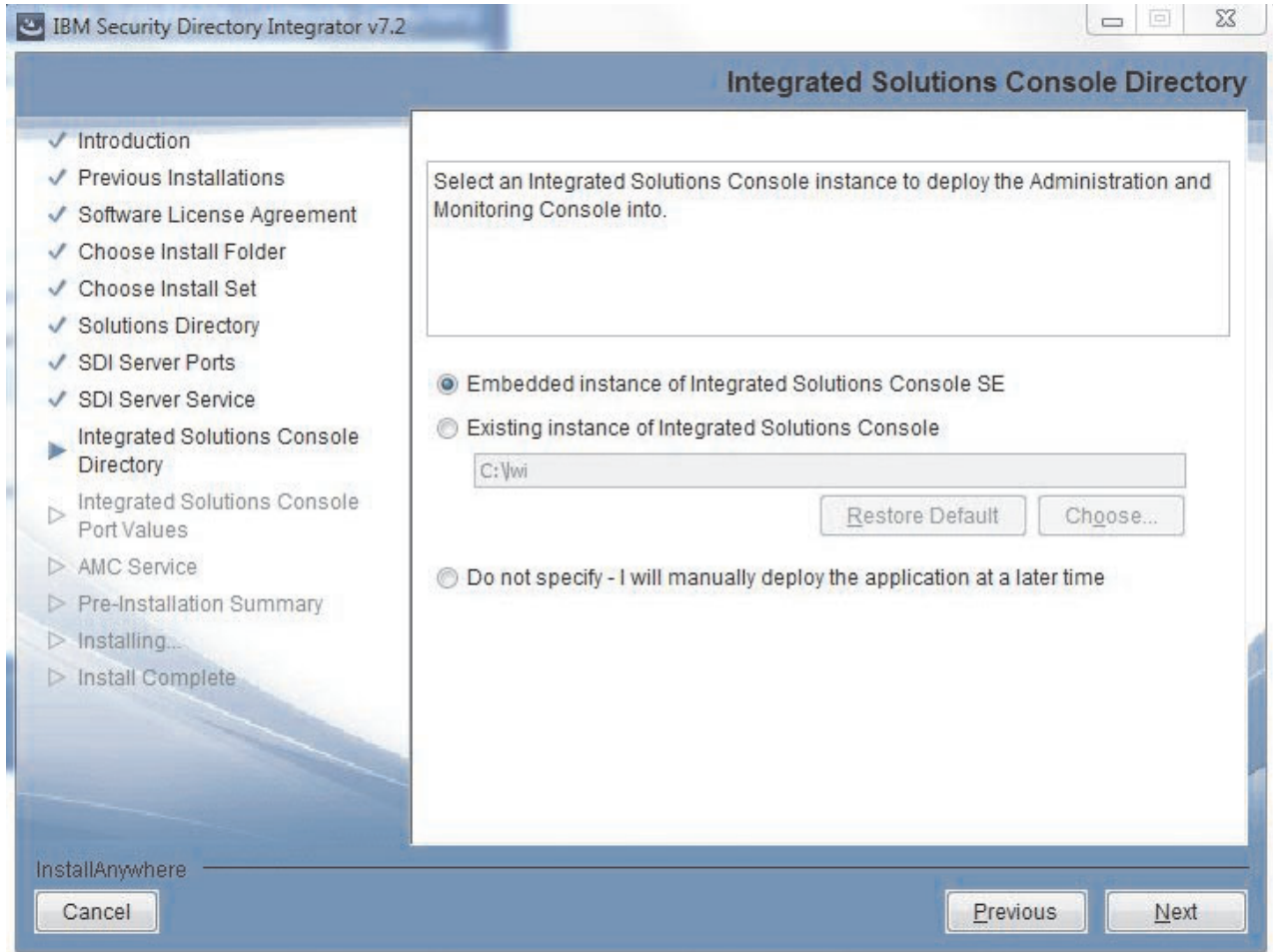
Diese Anzeige wird nur dann ausgegeben, wenn eine benutzerdefinierte Installationsgruppe und auch die Installation der AMC-Komponente ausgewählt wurde. Sie müssen die ISC-Instanz auswählen, in der AMC implementiert werden soll. Sie können die Implementierung von AMC in dem mit IBM Security Directory Integrator ausgelieferten ISC-Paket oder in einer bereits auf der Zielmaschine installierten ISC-Instanz auswählen bzw. angeben, dass AMC zu einem späteren Zeitpunkt implementiert werden soll. Bei Auswahl einer bereits installierten ISC-Instanz muss der Benutzer ein Verzeichnis auswählen, das die integrierte Webplattform (LWI) oder IBM WebSphere Application Server enthält, beispielsweise C:\Program Files\IBM\WebSphere\AppServer oder C:\dev\IBM\TDI\lwi.

Falls Sie die Komponente "Integrierte Webplattform" nicht zur Installation ausgewählt haben, ist diese Option abgeblendet.

Anmerkung:

1. Wenn Sie Komponenten hinzufügen wollen und die AMC-Komponente bereits installiert ist, wird diese Anzeige übersprungen.

- Bei einer Implementierung von AMC in IBM WebSphere Application Server wird die Rolle "Security Directory Integrator AMC Admin" (AMC-Administrator) anders als bei der Implementierung in der integrierten Webplattform nicht automatisch zugeordnet. Diese Rolle muss durch den ISC-Konsolenadministrator manuell zugeordnet werden.



Anzeige für ISC-Ports

Diese Anzeige wird entweder bei einer Standardinstallation oder bei einer benutzerdefinierten Installation ausgegeben, wenn Sie sich für die Implementierung von AMC in einer eingebetteten Instanz von ISC entschieden haben. Die ISC-Instanz könnte eine Instanz der eingebetteten ISC sein, die mit IBM Security Directory Integrator ausgeliefert wird, oder eine bereits auf dem Zielsystem vorhandene ISC-Instanz.

Falls Sie AMC in einer angepassten SE-Instanz implementieren, werden die für den HTTP-Port und den HTTPS-Port verwendeten Standardwerte wie folgt bestimmt:

Suchen Sie in den Dateien *für_tdi_ausgewählte_isc/conf/overrides/*.properties* nach dem ersten Vorkommen der Eigenschaften `com.ibm.pvc.webcontainer.port` und `com.ibm.pvc.webcontainer.port.secure` und verwenden Sie die zugeordneten Werte. Falls eine dieser Eigenschaften nicht in den Dateien ".properties" in diesem Verzeichnis definiert ist, suchen Sie in der Datei *für_tdi_ausgewählte_isc/conf/config.properties* nach ihnen. Wenn der

HTTP-Port nicht gefunden wird, wird standardmäßig Port 80 verwendet. Wird der HTTPS-Port nicht gefunden, wird standardmäßig Port 443 verwendet. Der Hilfeport hat denselben Wert wie der HTTP-Port.

Falls Sie AMC in einer angepassten AE-Instanz implementieren, werden die für den HTTP-Port und den HTTPS-Port verwendeten Standardwerte wie folgt bestimmt:

Suchen Sie im folgenden Verzeichnis nach Dateien namens "serverindex.xml":

für_tdi_ausgewählte_isc\profiles\AppSrv01\config\cells\nodes**

Suchen Sie in diesen Dateien für den HTTP-Port nach XML-Blöcken, die Ähnlichkeit mit dem folgenden Block haben:

```
<specialEndpoints xmi:id="NamedEndPoint_1200476459036"
  endPointName="WC_adminhost">
  <endPoint xmi:id="EndPoint_1200476459036" host="*" port="9060"/>
</specialEndpoints>
```

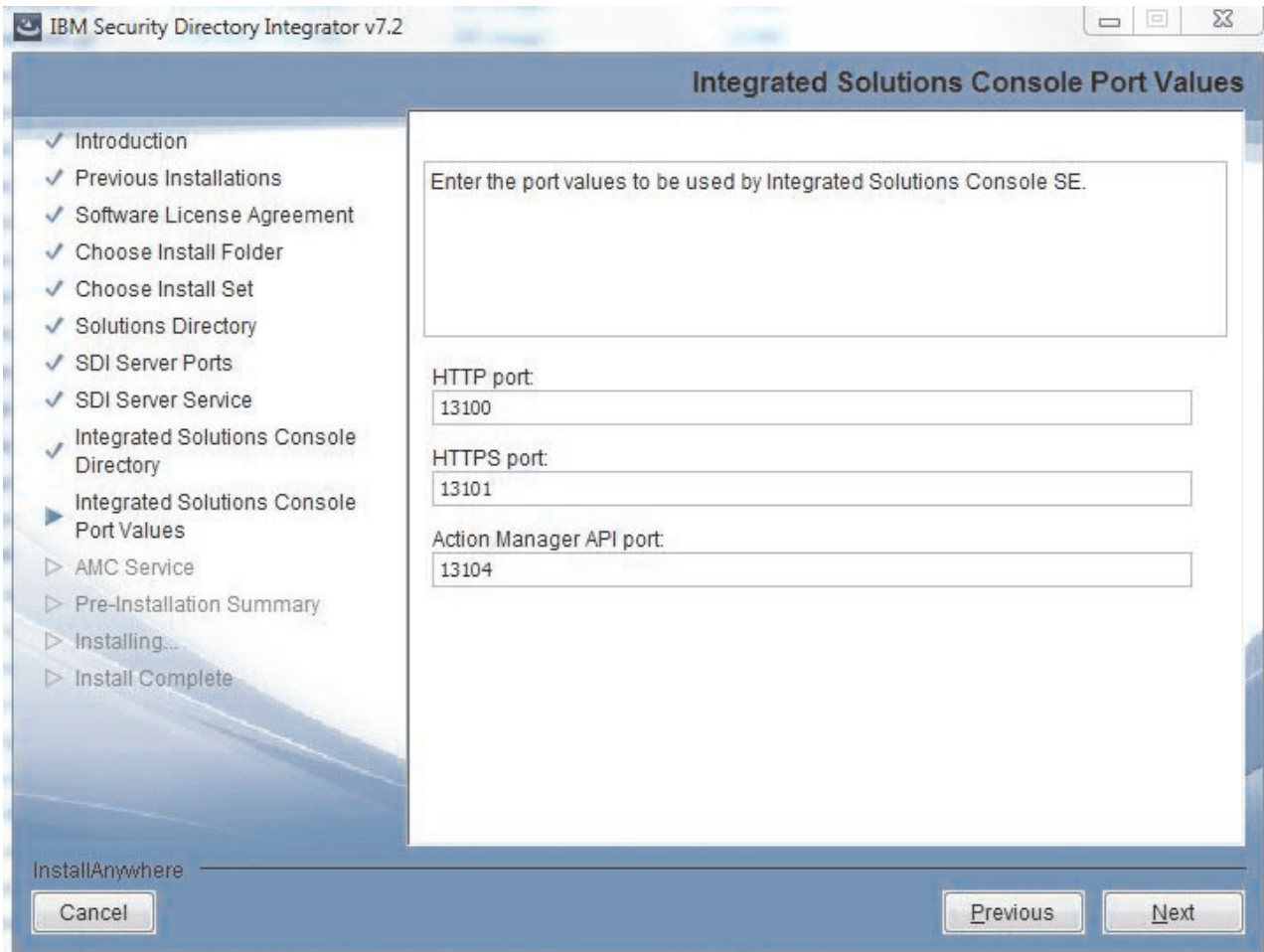
Suchen Sie für den HTTPS-Port nach Blöcken, die Ähnlichkeit mit dem folgenden Block haben:

```
<specialEndpoints xmi:id="NamedEndPoint_1200476459039"
  endPointName="WC_adminhost_secure">
  <endPoint xmi:id="EndPoint_1200476459039" host="*" port="9043"/>
</specialEndpoints>
```

Das Installationsprogramm sucht nach einem Tag "specialEndpoints", das für "endPointName" den Wert **WC_adminhost** oder **WC_adminhost_secure** enthält, und verwendet die zugeordneten Portwerte aus den eingebetteten Tags "endPoint". Falls der HTTP-Port durch dieses Verfahren nicht ermittelt werden kann, wird der Wert 9060 verwendet. Kann der HTTPS-Port nicht ermittelt werden, wird standardmäßig der Wert 9043 verwendet. Als Wert für den Hilfeport wird der Wert des HTTP-Ports festgelegt.

Die angezeigten Werte sind die Standardwerte für die integrierte ISC SE-Instanz.

Die Eingabe von bereits verwendeten Ports ist in der Anzeige nicht zulässig. In diesem Fall werden Sie in einer Warnung aufgefordert, einen anderen Portwert auszuwählen.

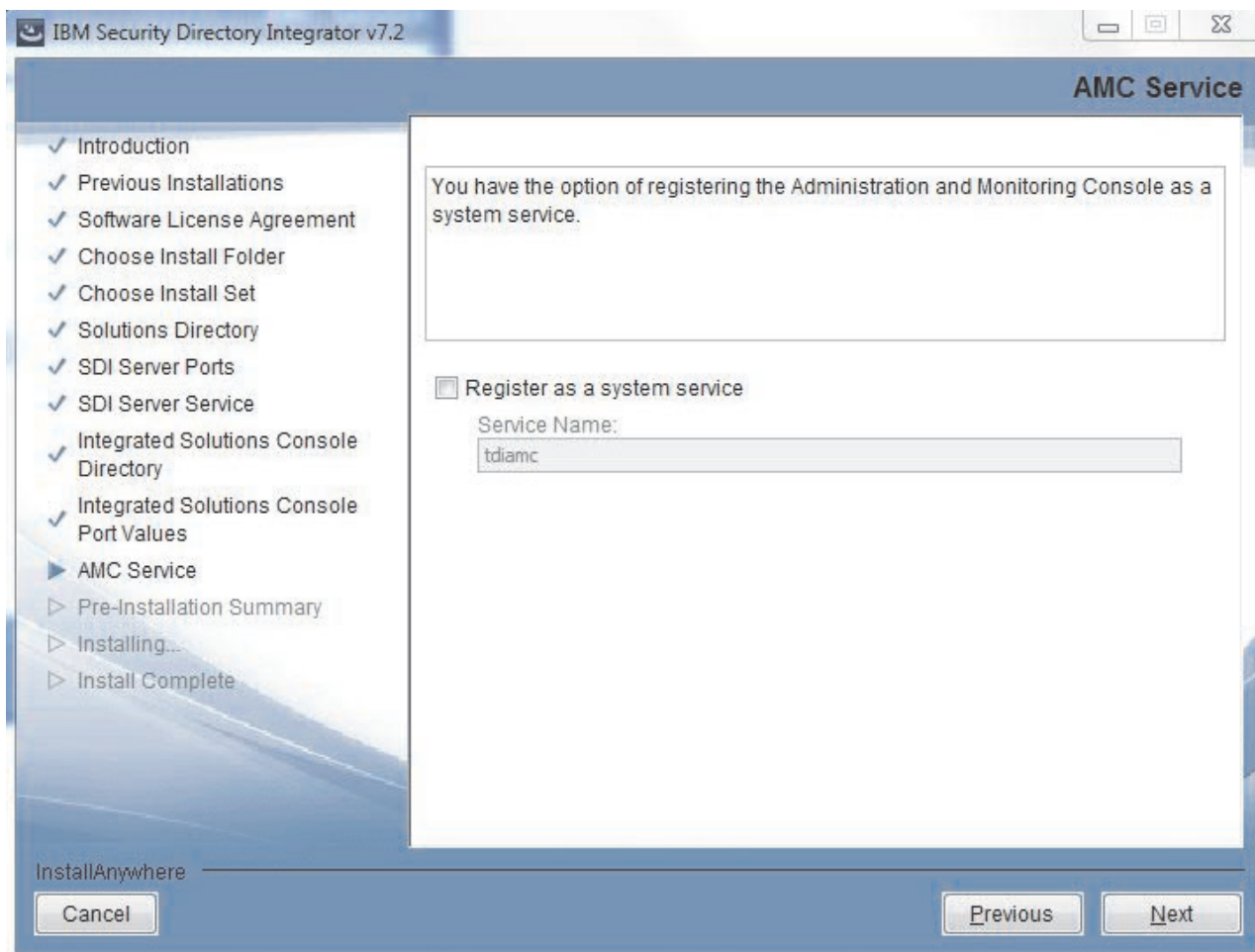


Anzeige für die Registrierung von AMC als Service

Falls das Markierungsfeld ausgewählt ist, wird AMC als Service für dieses Betriebssystem registriert.

In der Standardeinstellung ist das Markierungsfeld abgewählt.

Diese Anzeige wird nur dann ausgegeben, wenn die Komponenten "Integrierte Webplattform" und "Administration and Monitoring Console" (AMC) ausgewählt waren und Sie Administratorrechte besitzen.



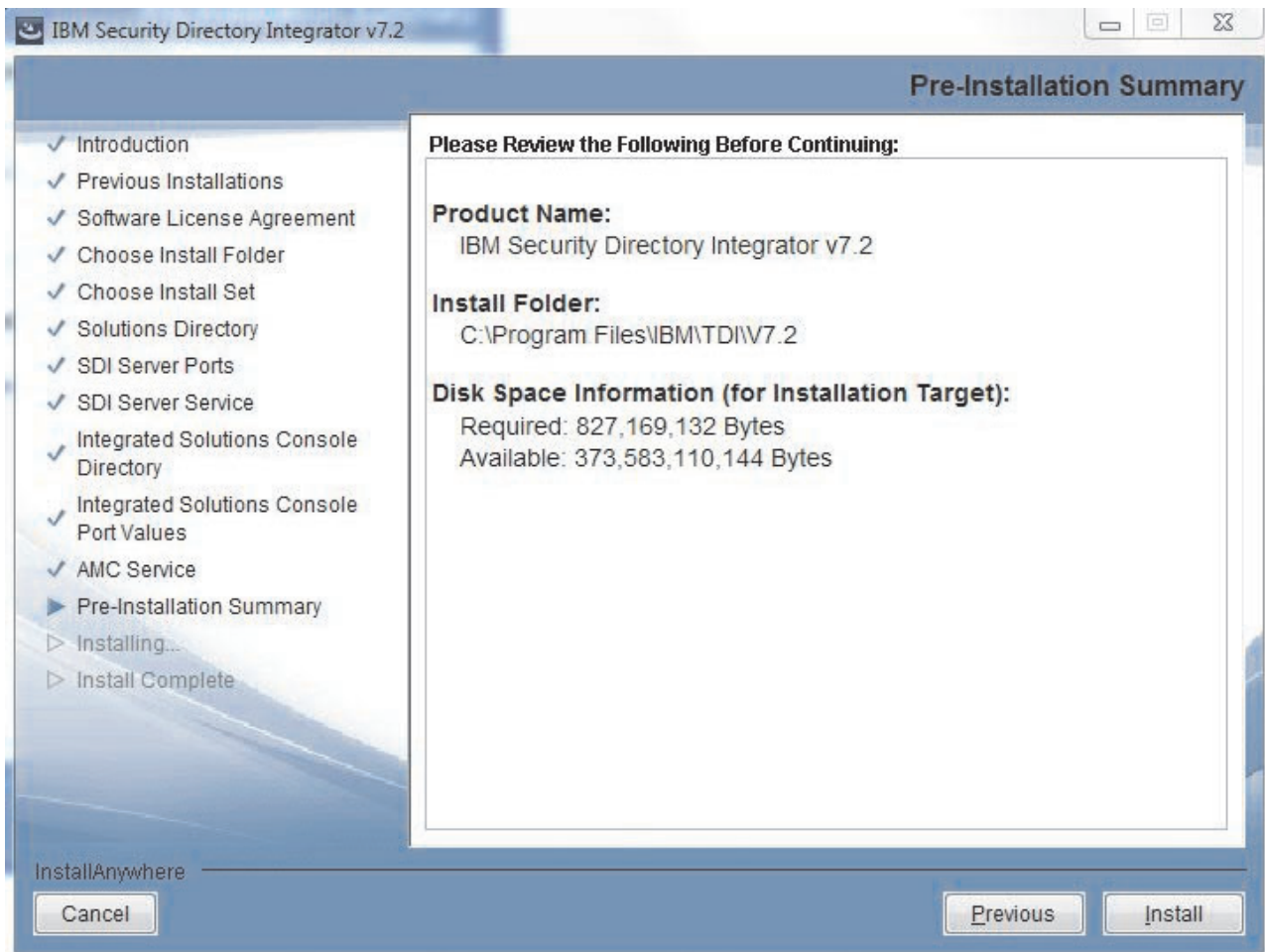
Das Installationsprogramm versucht nach Möglichkeit, einen gültigen Standardwert für den Servicenamen bereitzustellen (Details zu diesem Prozess können Sie den Informationen zur Registrierung von AMC als Windows-Dienst oder als UNIX-Prozess entnehmen). Falls das Installationsprogramm keinen gültigen Servicenamen bestimmen kann, ist das Feld leer. Sie können erst dann mit der Installation fortfahren, nachdem Sie einen gültigen Servicenamen eingegeben haben.

Anmerkung: Wenn Sie die Installation auf einem UNIX-System durchführen, stellen Sie sicher, dass der Servicenamen die maximale Länge von vier Zeichen nicht überschreitet. Wenn er vier Zeichen überschreitet, führt diese Einschränkung zu einem Fehler und die Installation kann nicht fortgesetzt werden.

Die AMC-Komponente ist veraltet und wird in einer zukünftigen Version von IBM Security Directory Integrator entfernt.

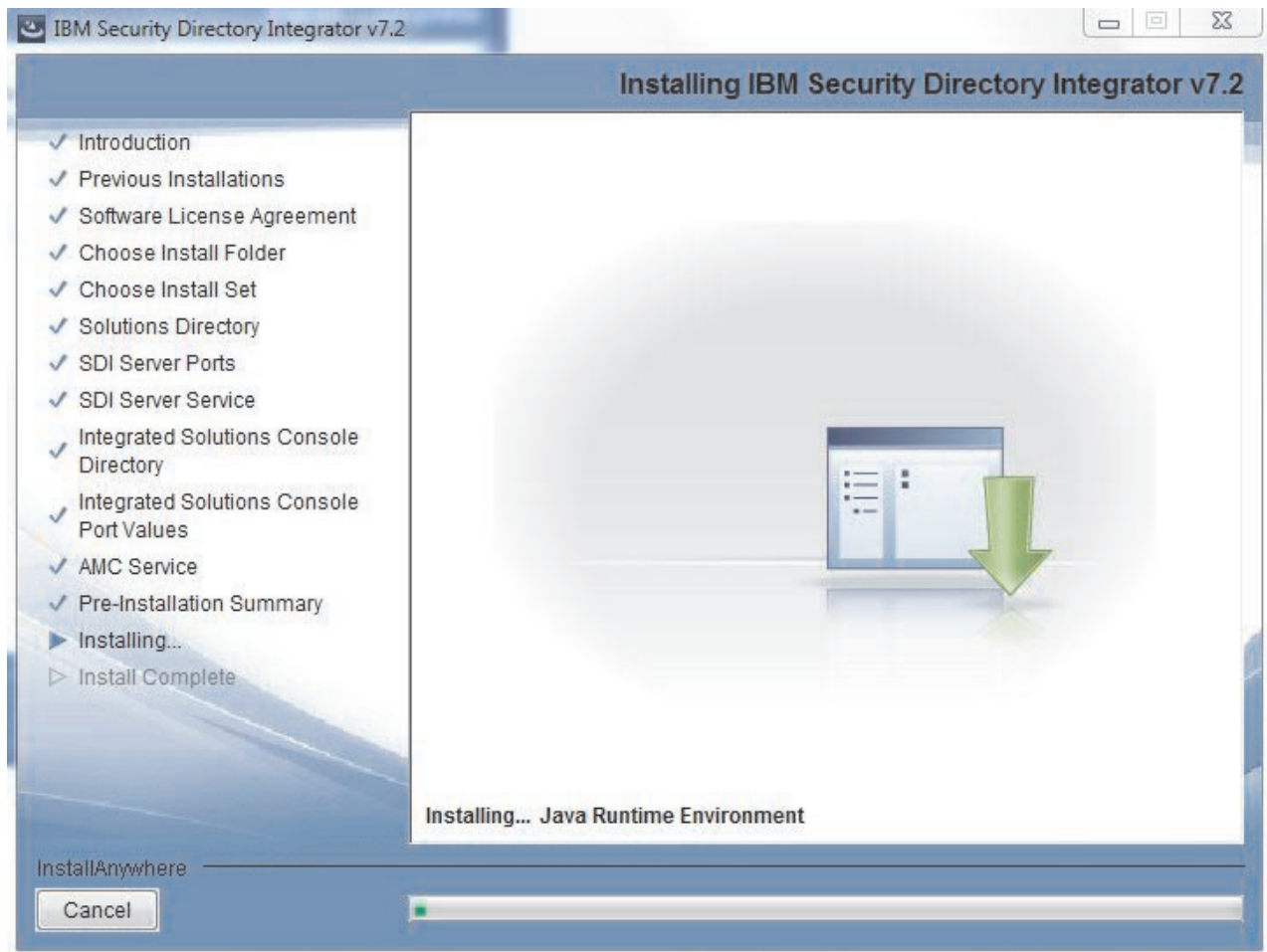
Anzeige mit Zusammenfassung der Installationsvorbereitung

In dieser Zusammenfassungsanzeige erhalten Sie einen Überblick darüber, welche Komponenten installiert werden und an welchen Positionen die Installation erfolgt.



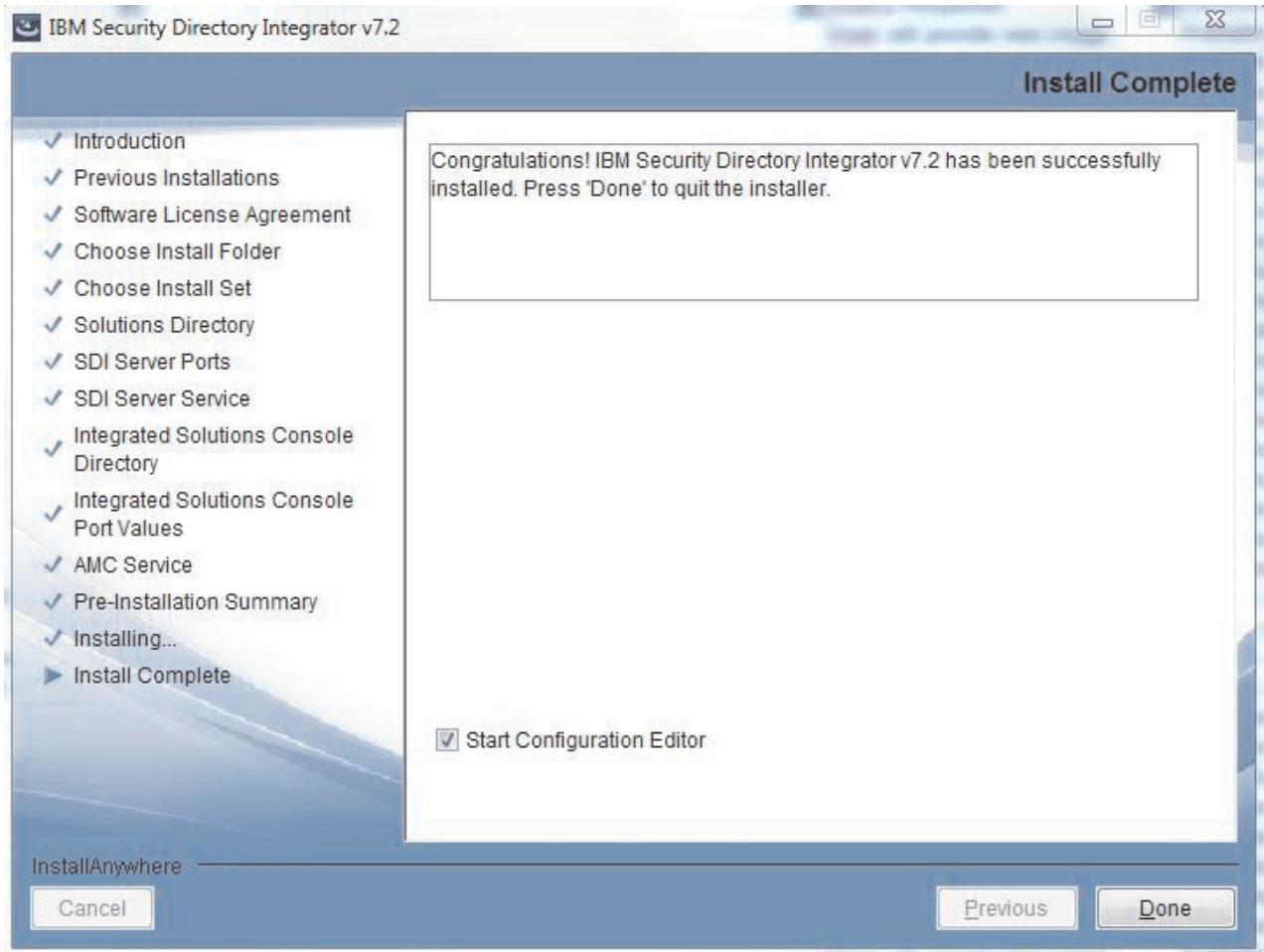
Anzeige für Installationsfortschritt

Diese Anzeige wird ausgegeben, während die eigentliche Installation stattfindet. Hierbei handelt es sich um eine von InstallAnywhere bereitgestellte Fortschrittsanzeige. Während sie angezeigt wird, werden alle Komponenten installiert.



Anzeige für Installationsabschluss

In dieser Anzeige ist angegeben, dass die Installation erfolgreich abgeschlossen wurde. Sobald Sie die Schaltfläche "Fertig" auswählen, ist die Installation beendet. Die Option **Konfigurationseditor starten** ist standardmäßig ausgewählt.

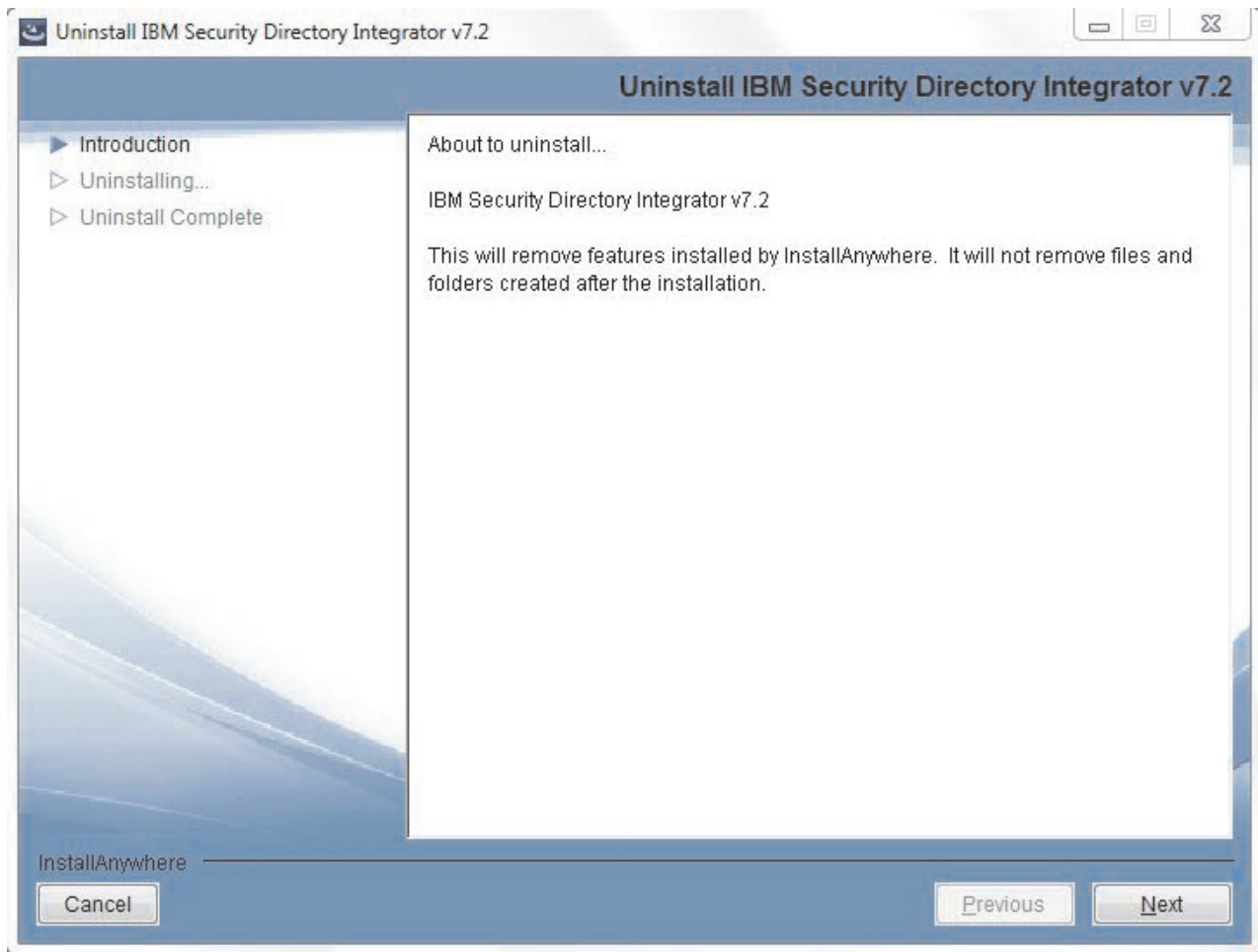


Anzeigenfolge bei der Deinstallation

Mit den hier aufgeführten Anweisungen können Sie die Anzeigenfolge deinstallieren.

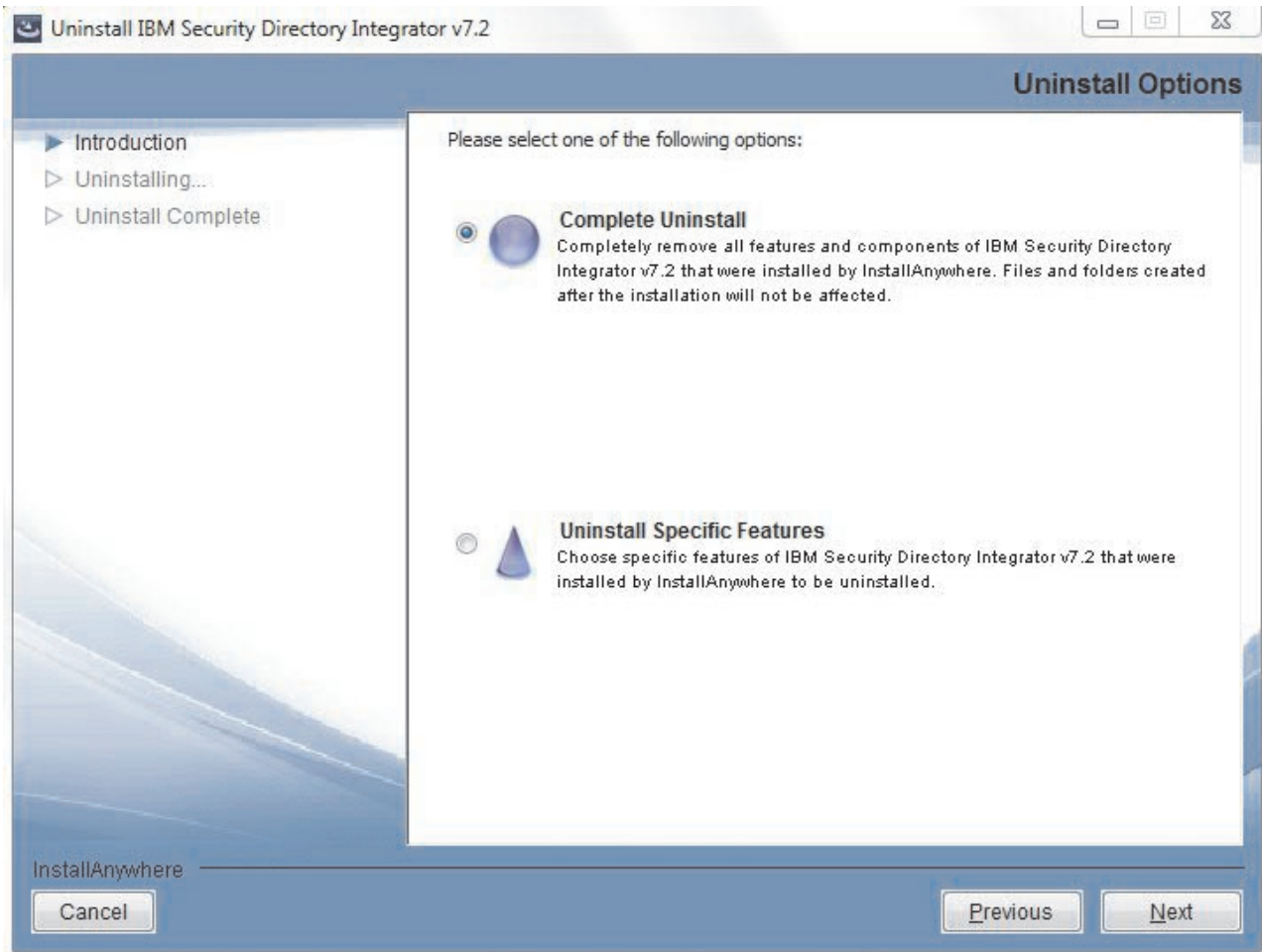
Eingangsanzeige der Deinstallation

Hierbei handelt es sich um eine InstallAnywhere-Anzeige mit Standardinhalt.

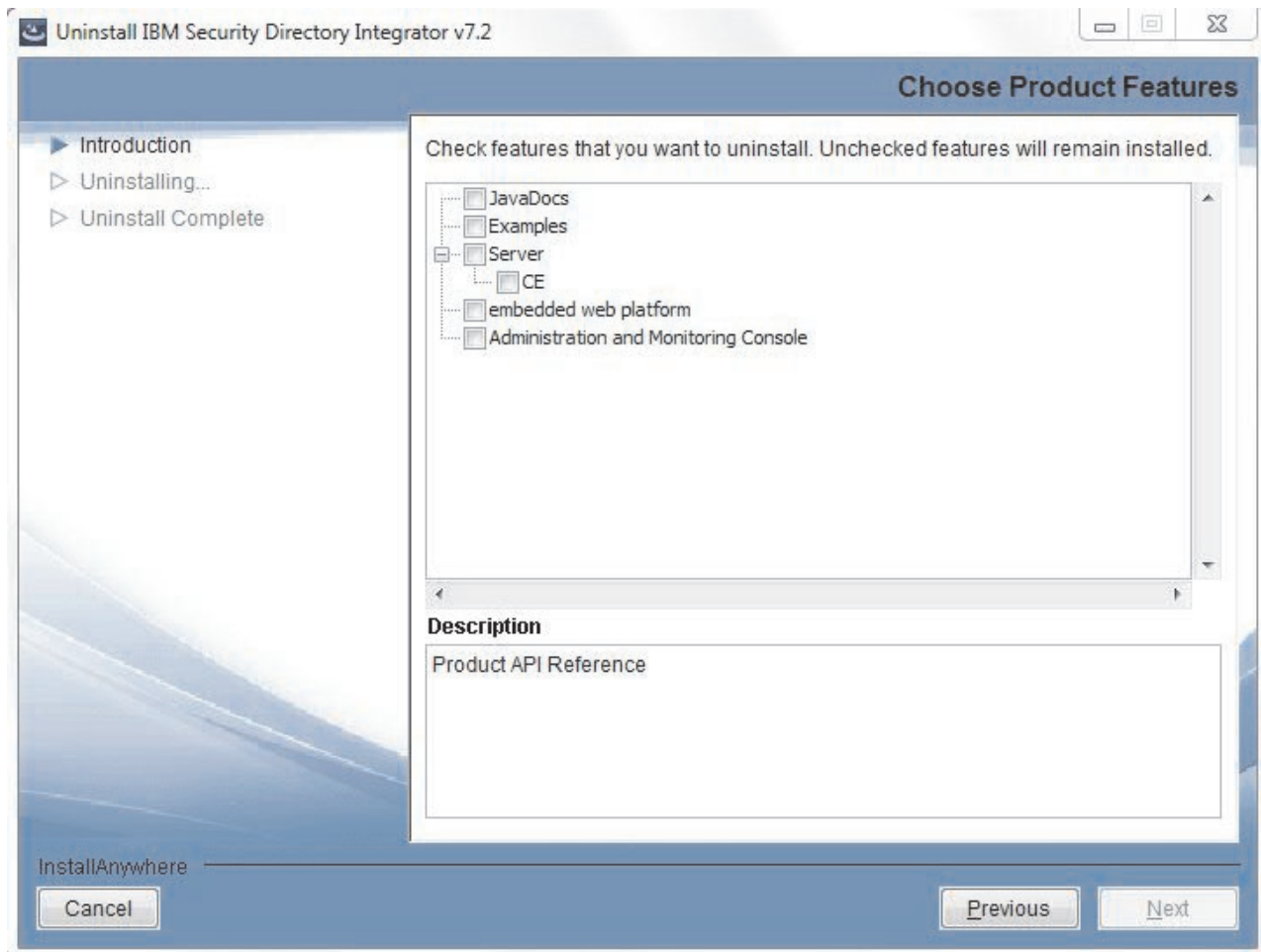


Anzeige für Produktkomponentenauswahl

In dieser Anzeige können Sie auswählen, ob das gesamte Produkt oder nur bestimmte Komponenten deinstalliert werden sollen.

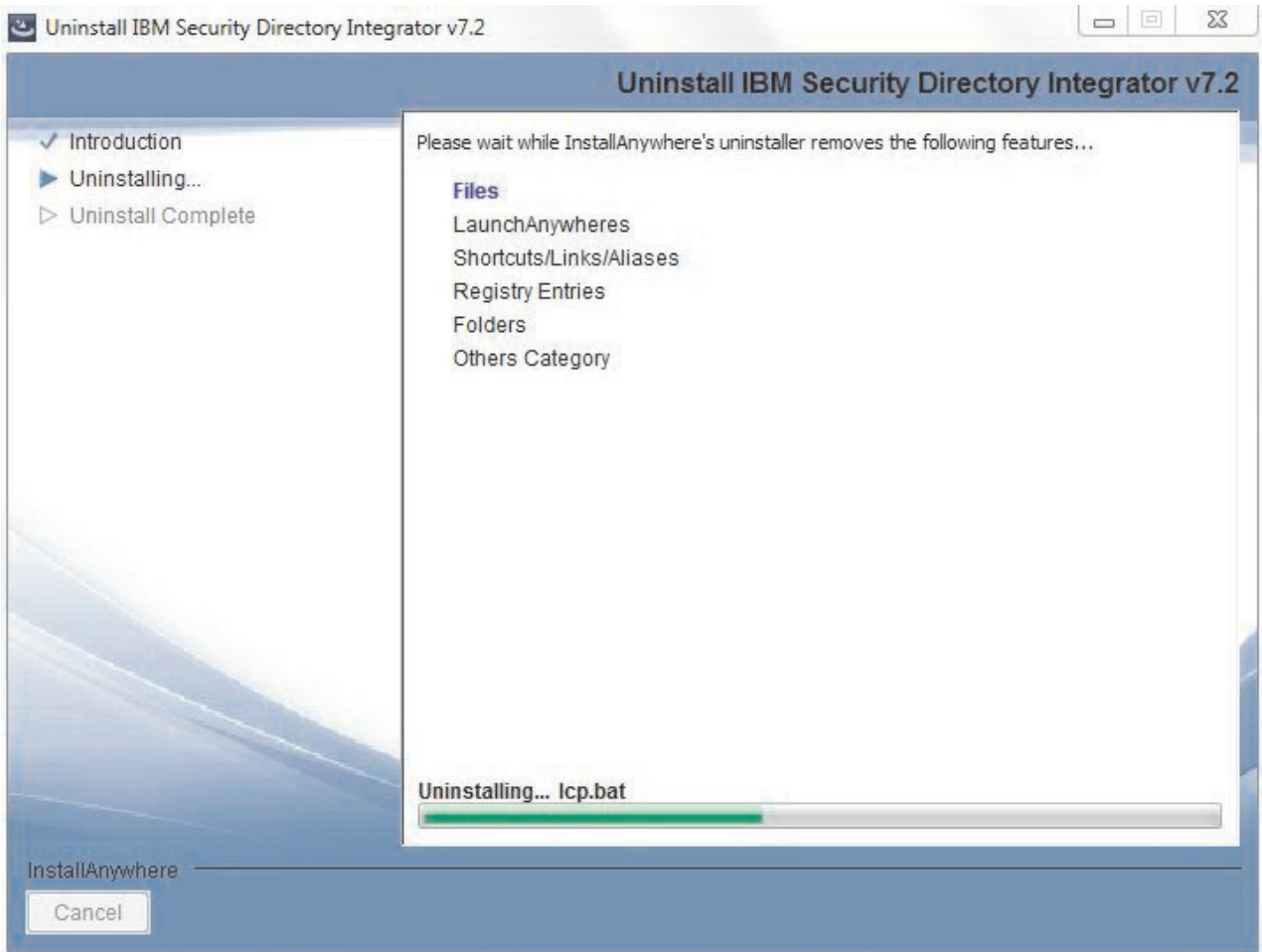


Falls die Option **Bestimmte Komponenten deinstallieren** ausgewählt wird, wird die folgende Anzeige ebenfalls aufgerufen:



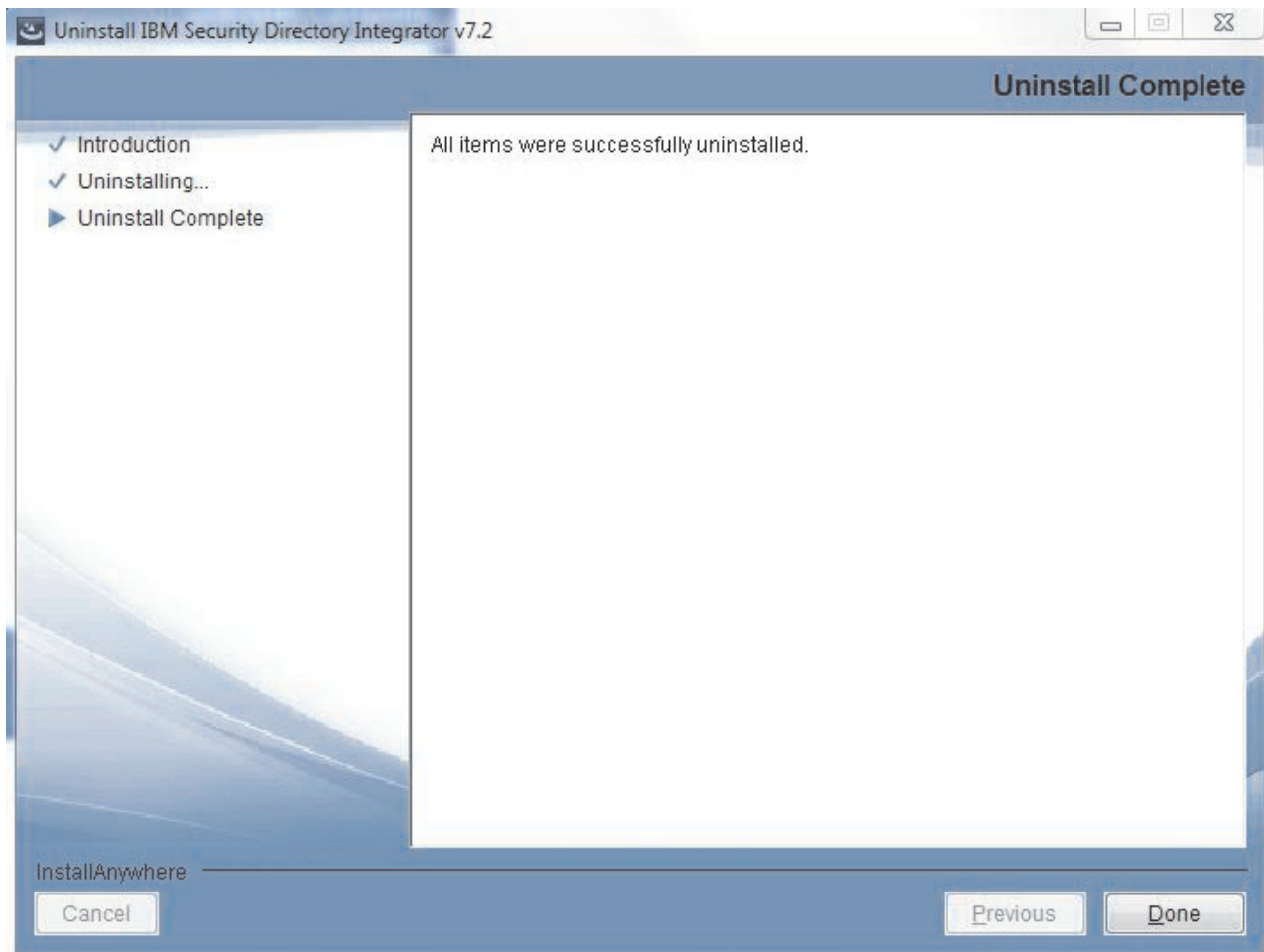
Anzeige für den Deinstallationsfortschritt

Diese Anzeige wird während der Deinstallation ausgegeben.



Anzeige für den Deinstallationsabschluss

In dieser Anzeige ist angegeben, dass die Deinstallation erfolgreich abgeschlossen wurde. Sobald Sie die Schaltfläche "Fertig" auswählen, wird das Deinstallationsprogramm beendet.



Anzeigenfolge beim Hinzufügen von Komponenten

Nachstehend erhalten Sie Informationen zur Anzeigenfolge beim Hinzufügen von Komponenten.

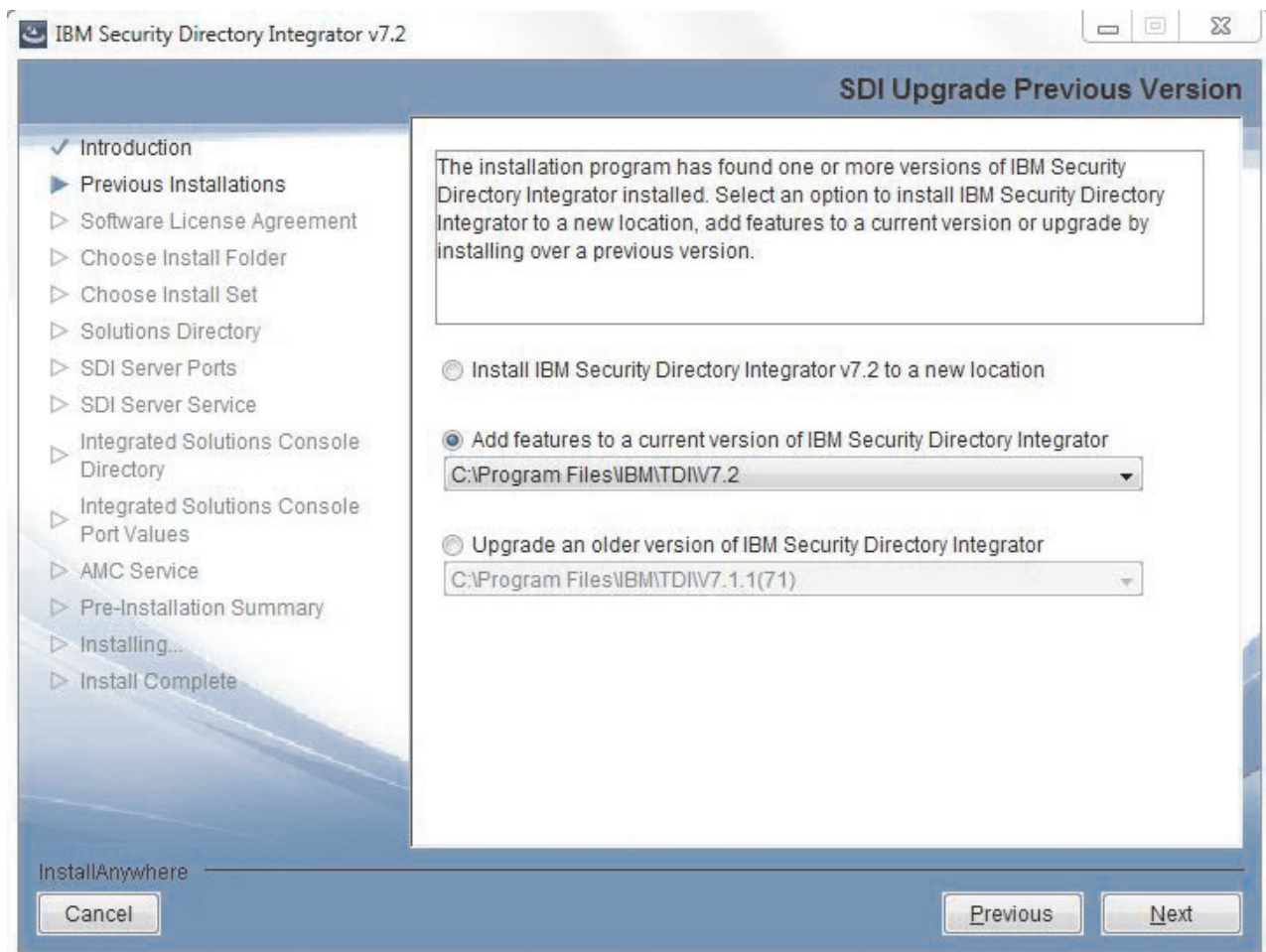
Die Anzeigenfolge beim Hinzufügen von Komponenten hat Ähnlichkeit mit der Anzeigenfolge bei einer Neuinstallation. Im Folgenden werden nur die für diese Folge spezifischen Anzeigen dargestellt.

Anzeige für Initialisierungsvorbereitung

Eingangsanzeige

Upgradeanzeige

Diese Anzeige wird nach der Eingangsanzeige und der Anzeige mit den Informationen zu vorhergehenden IBM Security Directory Integrator-Installationen ausgegeben, wenn auf der Einheit bereits eine IBM Security Directory Integrator-Instanz installiert ist.



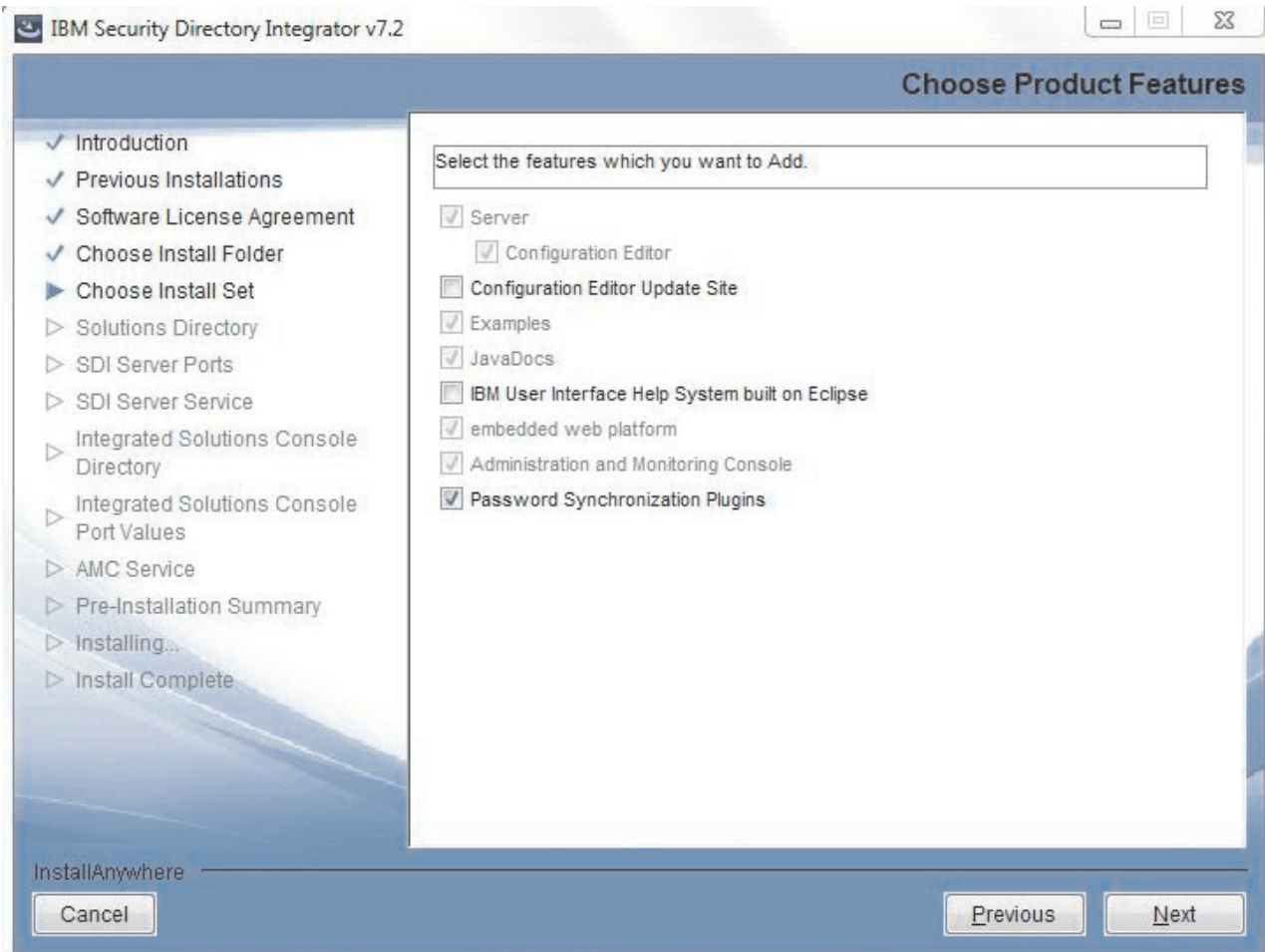
Die Schaltfläche zum Hinzufügen von Komponenten kann nicht ausgewählt werden, wenn keine Instanz von IBM Security Directory Integrator Version 7.2 verfügbar ist.

Wenn keine Vorversionen von IBM Security Directory Integrator verfügbar sind, kann die Schaltfläche **Upgrade** nicht ausgewählt werden.

Die Dropdown-Liste für IBM Security Directory Integrator wird beim Auswählen der Schaltfläche zum Hinzufügen von Komponenten aktiviert.

Anzeige für Komponentenauswahl

In der Anzeigenfolge für das Hinzufügen von Komponenten wird als Nächstes die Anzeige für die Komponentenauswahl ausgegeben, in der die bereits installierten Komponenten ausgewählt und abgeblendet sind.



An dieser Stelle können Sie alle gewünschten weiteren Komponenten hinzufügen.

Das Entfernen von Komponenten ist nicht zulässig.

Im Anschluss entspricht die Anzeigenfolge dem Ablauf bei der Neuinstallation. Anzeigen, die sich auf bereits installierte Komponenten beziehen, werden jedoch übersprungen.

Falls Sie den Konfigurationseditor auswählen, wird automatisch ebenfalls der Server ausgewählt. Wenn beide Komponenten ausgewählt sind und Sie den Server abwählen, wird analog auch der Konfigurationseditor abgewählt.

Anzeige für Security Directory Integrator-Lösungsverzeichnis

Anzeige für Registrierung des Servers als Systemservice

Anzeige für Security Directory Integrator-AMC-Implementierung

Anzeige für die Registrierung von AMC als Service

Anzeige mit Zusammenfassung der Installationsvorbereitung

Anzeige für Installationsfortschritt

Anzeige für Installationsabschluss

Anzeigenfolge bei der Migration

Nachstehend erhalten Sie Informationen zur Anzeigenfolge bei der Migration.

Die Anzeigenfolge bei der Migration hat Ähnlichkeit mit der Anzeigenfolge bei einer Neuinstallation. Im Folgenden werden nur die für diese Folge spezifischen Anzeigen dargestellt.

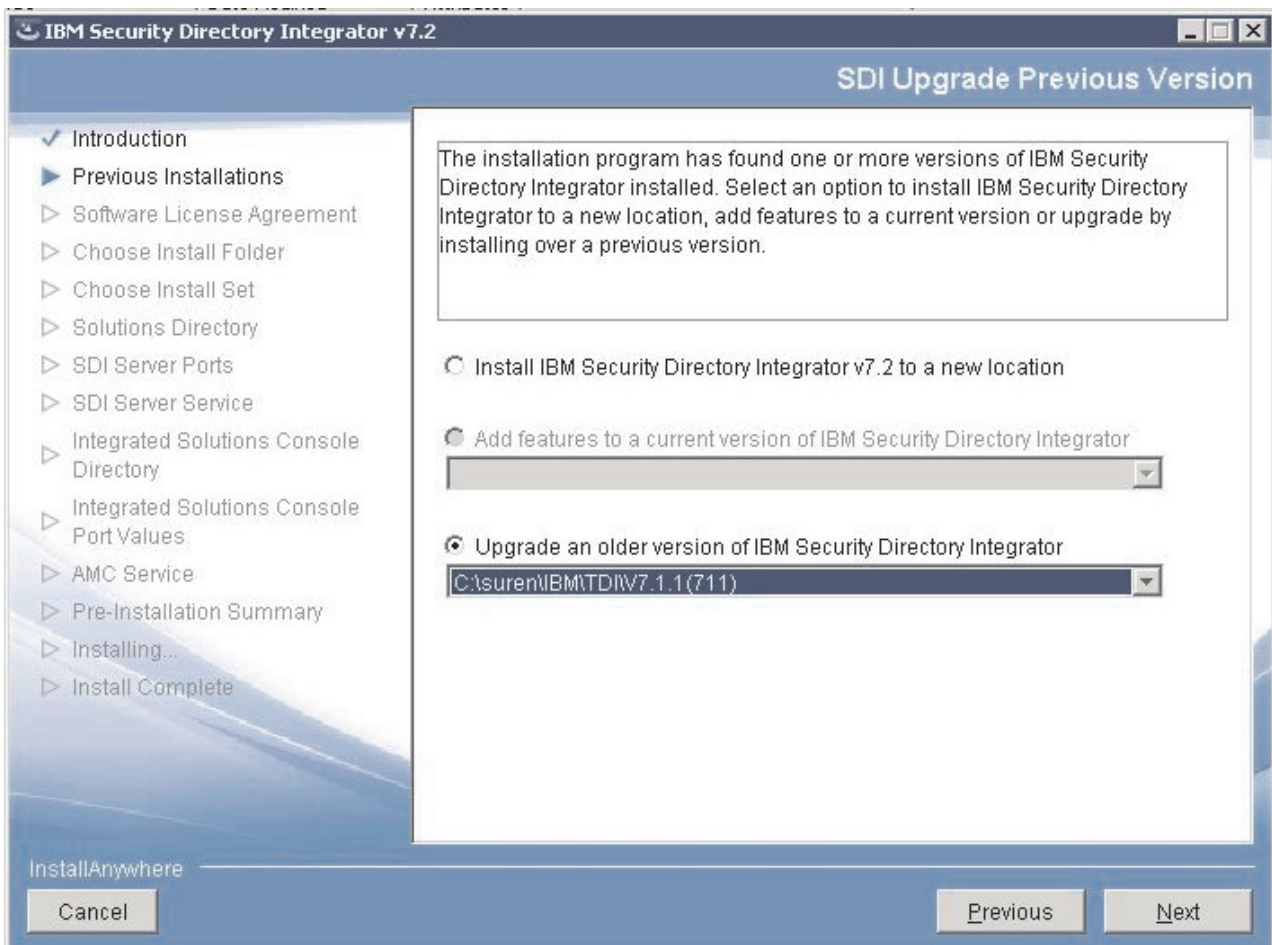
Anzeige für Initialisierungsvorbereitung

Eingangsanzeige

Upgradeanzeige

Anmerkung: Wenn Sie die Durchführung eines Upgrades von IBM Security Directory Integrator Version 7.1.1 auf Version 7.2 auf einem Windows 2012-Betriebssystem planen, müssen Sie vor dem Start des Installationsprogramms für Version 7.2 sicherstellen, dass der **Windows 7-Kompatibilitätsmodus** für die Datei "uninstaller.exe" von Version 7.1.1 aktiviert ist. Weitere Informationen finden Sie in den technischen Hinweisen unter <http://www-01.ibm.com/support/docview.wss?uid=swg21634336>.

Die folgende Anzeige wird nach der Eingangsanzeige und der Anzeige mit den Informationen zu vorhergehenden SDI-Installationen ausgegeben, wenn auf der Einheit bereits eine IBM Security Directory Integrator-Instanz installiert ist:



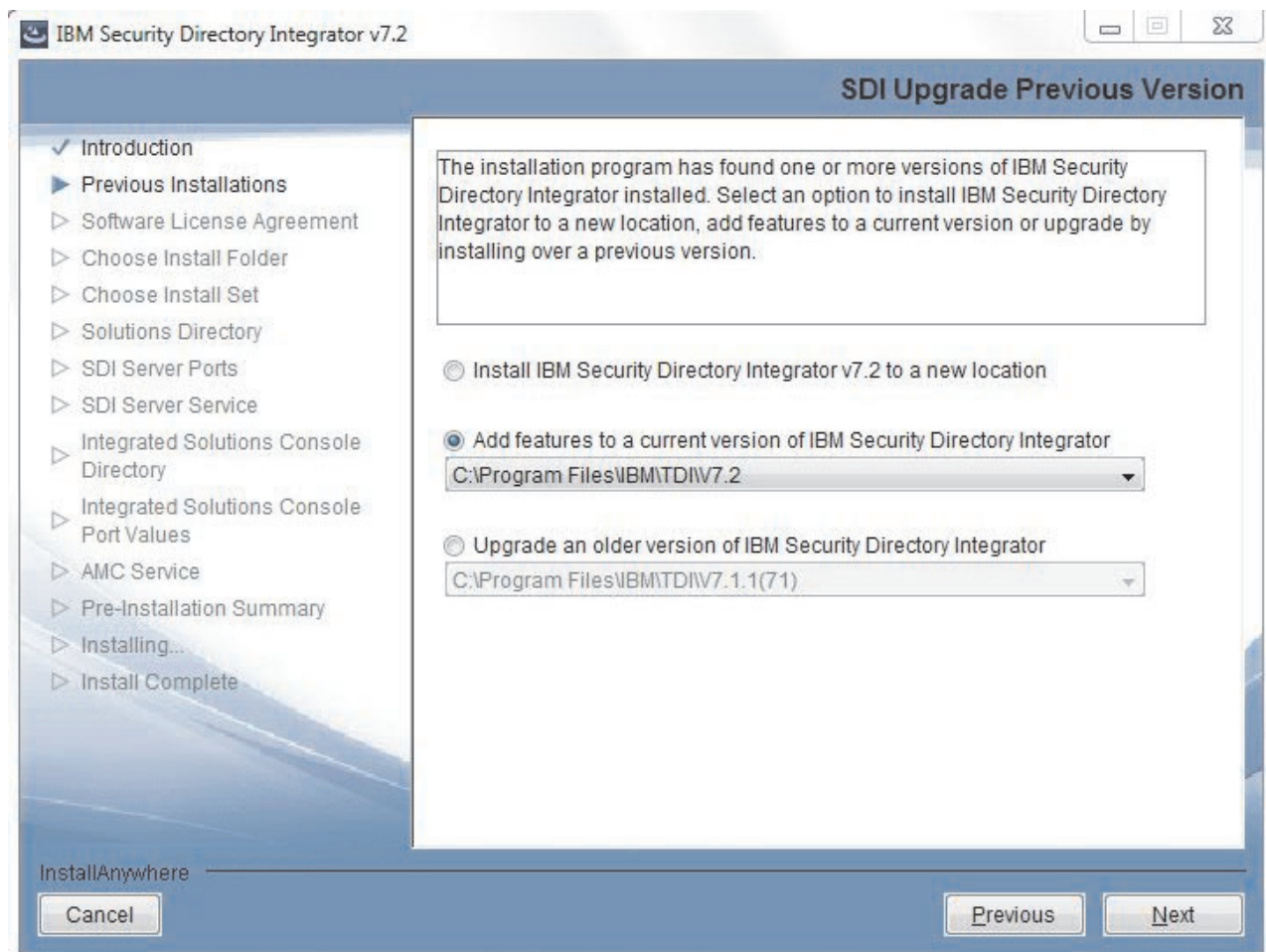
Die Schaltfläche zum Hinzufügen von Komponenten kann nicht ausgewählt werden, falls keine Instanz von IBM Security Directory Integrator Version 7.2 verfügbar ist.

Wenn keine Vorversionen von IBM Security Directory Integrator verfügbar sind, kann die Schaltfläche "Upgrade" nicht ausgewählt werden.

Die Dropdown-Liste für die Vorversionen von IBM Security Directory Integrator wird aktiviert, wenn das Feld "Upgrade" ausgewählt wird.

Anzeigenfolge bei Upgrade von IBM Security Directory Integrator 6.1.x, 7.0, 7.1.x

Anmerkung: Ein direktes Upgrade von IBM Security Directory Integrator von Version 6.x oder einer älteren Version auf Version 7.2 wird nicht unterstützt. Sie müssen zunächst ein Upgrade von Version 6.x auf Version 7.1.1 und anschließend von Version 7.1.1 auf Version 7.2 durchführen. Falls Sie das Upgrade von IBM Security Directory Integrator Version 7.1.1 auswählen, wird als Nächstes die Lizenzanzeige aufgerufen. Nachdem Sie die Lizenz akzeptiert haben, wird die Anzeige für die Komponentenauswahl ausgegeben:



In dieser Anzeige sind diejenigen Komponenten als ausgewählt angezeigt, die bereits in der Vorversion installiert waren. Außerdem können Sie neue Komponenten auswählen. Eine Inaktivierung von zuvor installierten Komponenten ist nicht zulässig.

Falls der Server zuvor installiert war, wird die Anzeige für das SDI-Lösungsverzeichnis übersprungen. Das Installationsprogramm verwendet den Wert aus der vorhergehenden Installation.

Falls AMC zuvor installiert war, wird die Anzeige für die Auswahl von ISC bei einer Migration von IBM Security Directory Integrator 6.x.x ausgegeben.

Im weiteren Verlauf entspricht die Anzeigenfolge dem Ablauf bei einer Neuinstallation. Bei einer Migration von IBM Security Directory Integrator 7.0 gibt es nur eine einzige neue Funktion, nämlich **Server als Service registrieren**. Diese Anzeige wird erst nach der Anzeige zur Komponentenauswahl ausgegeben.

Falls Sie den Konfigurationseditor auswählen, wird automatisch ebenfalls der Server ausgewählt. Wenn beide Komponenten ausgewählt sind und Sie den Server abwählen, wird analog auch der Konfigurationseditor abgewählt.

Installation über die Befehlszeile ausführen

Nachstehend finden Sie eine Liste der Befehle, die Sie bei der Installation verwenden können.

Das Installationsprogramm von IBM Security Directory Integrator unterstützt die folgenden Befehlszeilenoptionen:

-i Diese Option legt den Schnittstellenmodus für das Installationsprogramm fest ("silent", "console" oder "gui").

```
install_sdiv72_win_x86.exe -f name_der_antwortdatei -i silent
```

```
install_sdiv72_win_x86.exe -i console
```

-f Diese Option legt die Position einer Antwortdatei (Datei `installer.properties`) fest, die das Installationsprogramm verwenden soll.

```
install_sdiv72_win_x86.exe -f installer.properties
```

Der Pfad kann ein absoluter oder ein relativer Wert sein. (Relative Pfade beziehen sich auf die Position des Installationsprogramms.)

-r Diese Option erstellt eine Antwortdatei.

```
install_sdiv72_win_x86.exe -r myinstaller.properties
```

Anmerkung: Das Installationsprogramm von IBM Security Directory Integrator erstellt die Antwortdatei `tdi_respfile72.txt` im Verzeichnis mit den temporären Dateien des Systems auch dann, wenn die Option `-r` nicht angegeben ist. Beispiel:

- Auf der Windows-Plattform wird die Antwortdatei im Verzeichnis `C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp` erstellt.
- Auf der Nicht-Windows-Plattform wird die Antwortdatei im Verzeichnis `/tmp` erstellt.

Das Verzeichnis `tdi-installationsverzeichnis/examples/install` enthält Beispielantwortdateien für verschiedene Installations- oder Deinstallations-szenarios.

-D Diese Option übergibt angepasste Befehlszeilenargumente.

```
install_sdiv72_win_x86.exe -Dmyvar=myvalue
```

- l** Diese Option verwendet den angegebenen Sprachencode (und optionalen Landescode), um die Ländereinstellung für das InstallAnywhere-Installationsprogramm festzulegen.

```
install_sdiv72_win_x86.exe -l en
```

```
install_sdiv72_win_x86.exe -l pt_BR
```

Der erforderliche Sprachencode ist ein aus zwei Buchstaben (normalerweise in Kleinschreibung) bestehender Code, der im ISO-639-Standard definiert ist. InstallAnywhere akzeptiert sowohl alte Sprachencodes (iw, ji und in) als auch neue Sprachencodes (he, yi und id).

Der optionale Landescode ist ein aus zwei Buchstaben (normalerweise in Großschreibung) bestehender Code, der im ISO-3166-Standard definiert ist.

Ländereinstellungsoptionen werden nur dann berücksichtigt, wenn das Installationsprogramm Lokalisierungen für die von Ihnen angegebene Ländereinstellung bietet.

- ?** Diese Option zeigt den Hilfetext für das InstallAnywhere-Installationsprogramm an.

Unter Windows kann die Option `-help` nur über das Konsolenstartprogramm verwendet werden. Achten Sie darauf, für LaunchAnywhere auf der Registerkarte Windows der Untertask **Project > Platforms** die Einstellung **Console** festzulegen. (Damit eine installierte LaunchAnywhere-Instanz diese Informationen bereitstellt, müssen Sie darauf achten, dass als Aktion explizit das Konsolenstartprogramm festgelegt ist.)

Die folgende Befehlszeilenoption ist ausschließlich beim Installationsassistenten von IBM Security Directory Integrator verfügbar:

LAX_VM

Der Parameter LAX_VM wird zum Booten des Installationsprogramms von der Java Virtual Machine verwendet, die auf dem System installiert ist.

Sie müssen den absoluten Pfad der ausführbaren Java-Datei angeben, die sich im Java-bin-Verzeichnis befindet. Beispiel:

```
install_sdiv72_win_x86.exe LAX_VM "java-verzeichnis/jre/bin/java.exe"
```

Verwenden Sie nur die Leerzeichen zwischen den Argumenten.

Anmerkung: Stellen Sie sicher, dass Sie den absoluten Pfad von IBM JRE 7.0.4 und höher als Parameterwert verwenden. Das Installationsprogramm von IBM Security Directory Integrator arbeitet mit anderen JREs möglicherweise nicht ordnungsgemäß.

-D\$TDI_BACKUP\$="true"

Dieser Parameter sollte nur bei einer Deinstallation übergeben werden. Er wird im Hinblick auf künftige Migrationen bereitgestellt. Beispiel:

```
tdi-installationsverzeichnis\uninst\uninstaller.exe -D$TDI_BACKUP$="true"
```

Hierdurch wird das Deinstallationsprogramm angewiesen, das Script `tdi-installationsverzeichnis/bin/tdiBackup.bat (.sh)` auszuführen, das seinerseits die Erstellung eines Verzeichnisses `tdi-installationsverzeichnis/backup_tdi` bewirkt. In diesem Verzeichnis wird eine Reihe von Dateien gespeichert, die für Ihre Installation spezifisch sind. Hierzu gehören Ihre globale Eigenschaftendatei, globale Zertifikate und dergleichen.

Anmerkung: Auf Nicht-Windows-Systemen muss vor dem Dollarzeichen (\$) ein umgekehrter Schrägstrich (\) stehen. Beispiel:

```
tdi-installationsverzeichnis\_uninst\uninstaller -D\%TDI_BACKUP\%="true"
```

-D\$TDI_SKIP_VERSION_CHECK\$="true"

Dieser Parameter bewirkt, dass das Installationsprogramm die Prüfung auf Vorversionen überspringt. Dies inaktiviert im Wesentlichen die Migration ausgehend von Vorgängerreleases.

Wenn bei einer unbeaufsichtigten Installation diese Option zum Überspringen ausgewählt ist und das Installationsverzeichnis mit dem Verzeichnis einer früheren Installation von IBM Security Directory Integrator identisch ist, wird das Installationsprogramm gestoppt.

Anmerkung: Auf Nicht-Windows-Systemen muss vor dem Dollarzeichen (\$) ein umgekehrter Schrägstrich (\) stehen. Beispiel:

```
./install_sdiv72_linux_x86_64.bin -D\%TDI_SKIP_VERSION_CHECK\%="true"
```

-D\$TDI_NOSHORTCUTS\$="true"

Mit diesem Parameter wird verhindert, dass das Installationsprogramm Verknüpfungen zum Deinstallationsprogramm, zum Konfigurationseditor oder zu AMC erstellt.

Anmerkung: Auf Nicht-Windows-Systemen muss vor dem Dollarzeichen (\$) ein umgekehrter Schrägstrich (\) stehen. Beispiel:

```
./install_sdiv72_linux_x86_64.bin -D\%TDI_NOSHORTCUTS\%="true"
```

Speicherbedarf für temporäre Dateien während der Installation

Mit den hier aufgeführten Anweisungen können Sie den Speicher für temporäre Dateien optimal nutzen.

Während der Installation verwendet das Installationsprogramm einen beträchtlichen Umfang des Speicherbereichs für temporäre Dateien, um Dateien zwischenspeichern. Falls Ihr System in dieser Hinsicht Einschränkungen unterliegt, kann es sein, dass bei der Installation Fehler auftreten.

UNIX- und Linux-Systeme verwenden normalerweise das Verzeichnis /tmp oder /var/tmp als Speicher für temporäre Dateien. Unter Windows befindet sich der Speicherbereich für temporäre Dateien hingegen an der Position, auf die die Umgebungsvariable TEMP zeigt.

InstallAnywhere-Installationsprogramme können angewiesen werden, die Verwendung von temporären Dateien umzuadressieren, indem vor dem Start des Installationsprogramms die Umgebungsvariable IATEMPDIR festgelegt wird. Beispiel für UNIX:

```
export IATEMPDIR=/opt/IBM/TDI/temp
```

Starten Sie anschließend die Installationsprogramme im Konsolenmodus aus der Sitzung heraus, in der Sie die Variable IATEMPDIR festgelegt haben.

Unbeaufsichtigte Installation ausführen

Mit den hier aufgeführten Anweisungen können Sie eine unbeaufsichtigte Installation ausführen.

Um eine unbeaufsichtigte Installation ausführen zu können, müssen Sie zuerst eine Antwortdatei generieren. Zur Generierung dieser Datei führen Sie eine beaufsichtigte Installation aus, bei der Sie die Option `-r` angeben. Beispiel:

```
install_sdiv72_win_x86.exe -r name_der_antwortdatei
```

Die Antwortdatei wird in dem Verzeichnis erstellt, das Sie während der Installation angeben.

Anmerkung: Das Verzeichnis `tdi-installationsverzeichnis/examples/install` enthält eine Reihe von Beispielantwortdateien für verschiedene Installations- und Deinstallationszenarios.

Sobald die Antwortdatei erstellt wurde, können Sie mit dem folgenden Befehl eine unbeaufsichtigte Installation ausführen:

```
install_sdiv72_win_x86.exe -i silent -f name_der_antwortdatei
```

Anmerkung: Bei den Beispielen im vorliegenden Dokument wird die ausführbare Datei für die Installation auf der Windows-Plattform verwendet. Unter „Geeignetes Installationsprogramm starten“ auf Seite 10 finden Sie eine Liste mit den Namen der ausführbaren Dateien für die verschiedenen unterstützten Plattformen.

Einschränkung für Servicenamen auf UNIX-Systemen

Beachten Sie die Einschränkungen, wenn Sie auf UNIX-Systemen einen Service benennen.

Stellen Sie bei der unbeaufsichtigten Installation von IBM Security Directory Integrator auf UNIX-Systemen sicher, dass der Servicenamen für IBM Security Directory Integrator und AMC die maximale Länge von vier Zeichen nicht überschreitet. Dieser Wert wird in der Antwortdatei `examples\install\TDICustomInstallRsp_Unix.txt` angegeben.

Die unbeaufsichtigte Installation schlägt fehl, wenn Sie einen Namen angeben, der länger als vier Zeichen ist. Beispiel:

```
TDI_SERVER_SERVICENAME=tdisrv_silent  
TDI_AMC_SERVICENAME=tdiamc_silent
```

In den Protokolldateien `/tmp/sdiv72install.log` und `/tmp/sdiv72debug.log` wird der folgende Fehler gemeldet:

```
tdisrv_silent muss 4 oder weniger Zeichen lang sein.  
Es ist bereits ein Service mit diesem Namen vorhanden, oder es wurde ein ungültiger Name angegeben.  
Auf UNIX-Plattformen sind Namen mit maximal 4 Zeichen zulässig.
```

Wenn dieser Fehler auftritt, müssen Sie die Antwortdatei bearbeiten und die Werte der Eigenschaften `TDI_SERVER_SERVICENAME` und `TDI_AMC_SERVICENAME` ändern, sodass sie vier oder weniger Zeichen enthalten.

Schritte für den Installationsabschluss

Führen Sie die hier aufgeführten Schritte aus, nachdem die Installation erfolgt ist.

Aktualisierungsseite des Konfigurationseditors

Nachstehend erhalten Sie Informationen zur manuellen Implementierung von Eclipse.

Falls die Komponente "Aktualisierungsseite des Konfigurationseditors" installiert wurde, müssen Sie jetzt die manuelle Implementierung in Eclipse vornehmen. Wei-

tere Informationen enthält der Abschnitt „Installation oder Aktualisierung mit Eclipse-Aktualisierungsmanager vornehmen“ auf Seite 52.

Plug-ins

Nachstehend erhalten Sie Informationen zum Zugriff auf die Dokumentation zu Plug-ins zur Kennwortsynchronisation.

Falls eines der Plug-ins zur Kennwortsynchronisation installiert wurde, finden Sie im IBM Knowledge Center for IBM Security Directory Integrator im Abschnitt *Plug-ins für den Kennwortabgleich* Informationen dazu, wie Sie den Plug-in-Code implementieren.

Administration and Monitoring Console (AMC)

Nachstehend erhalten Sie allgemeine Informationen sowie Informationen zur Webplattform- und verzögerten Implementierung von Administration and Monitoring Console.

Allgemeine Informationen

- Weitere Informationen zu AMC enthält „Administration and Monitoring Console (AMC)“.
- Wenn Sie zur Anmeldung bei der Konsole bereit sind, rufen Sie die Adresse <http://hostname:port/ibm/console> auf. Weitere Informationen enthält der Abschnitt „Bei der Konsole an- und abmelden“ auf Seite 289.
- Weitere Informationen zum Hinzufügen von Benutzern und Benutzerrollen finden Sie im Abschnitt „AMC in ISC“ auf Seite 273 unter „Konsolbenutzerberechtigung“.

Implementierung der im Produktpaket enthaltenen integrierten Webplattform

- Falls Sie AMC mit der im Produktpaket enthaltenen integrierten Webplattform installiert haben und mit der Verwendung von AMC beginnen wollen, müssen Sie die Befehle zum Starten von AMC und Action Manager ausführen, bevor Sie sich bei der ISC-Konsole anmelden können. Weitere Informationen enthält der Abschnitt „AMC und Action Manager starten sowie Anmeldung durchführen“ auf Seite 269.

Anmerkung: Unter Windows wird im Startmenü unter "Programmdateien" eine Verknüpfung zur Datei "launchAMC.html" erstellt.

- Der Benutzer, der IBM Security Directory Integrator installiert, ist standardmäßig der einzige Benutzer mit Anmeldezugriff auf die Konsole.

Angepasste oder verzögerte Implementierung

- Falls Sie eine angepasste ISC SE-Instanz oder IBM Dashboard Application Services Hub für die Implementierung von AMC verwenden wollen und nun mit der Implementierung beginnen können, lesen Sie die Informationen unter „AMC in angepasster ISC SE-Instanz oder IBM Dashboard Application Services Hub implementieren“ auf Seite 51. Bei einer solchen Implementierung von AMC ordnet das Installationsprogramm dem aktuellen Benutzer nicht automatisch die Rolle "SDI AMC Admin" (SDI-AMC-Administrator) zu. Diese Berechtigung muss durch einen Administrator der ISC-Konsole manuell erteilt werden. In der Regel erfolgt dies über die Anzeige **Benutzer und Gruppen -> Rollen für Benutzer mit Verwaltungsaufgaben** der IBM Dashboard Application Services Hub-Konsole. Alternativ kann diese Rolle auch mit dem Befehl `setAMCRoles` zugeordnet werden.

- Falls Sie ausgewählt haben, die Implementierung von AMC in einer ISC zu verzögern, und dies nun vornehmen möchten, lesen Sie die Informationen im Abschnitt „AMC in angepasster ISC SE-Instanz oder IBM Dashboard Application Services Hub implementieren“ auf Seite 51.

Anmerkung: Wenn Sie die Implementierung in einer angepassten ISE SE- oder ISC AE-Instanz ausgeführt haben, müssen Sie nach dem Starten der ISC SE-/ISC AE-Instanz, in der AMC installiert wurde, mindestens sicherstellen, dass Action Manager gestartet wurde.

Dokumentation

Sie können auf die Dokumentation online zugreifen oder sie manuell implementieren. Nachstehend erhalten Sie Informationen dazu.

IBM Security Directory Integrator verwendet das IBM Knowledge Center als Dokumentationssystem. Dies bedeutet, dass Ihnen die Dokumentation zu IBM Security Directory Integrator online zur Verfügung gestellt wird, nachdem Sie eine Standardinstallation ausgeführt haben. Die Bereitstellung erfolgt im World Wide Web durch IBM. Sie können jedoch die Dokumentation auch lokal implementieren. Weitere Informationen können Sie dem Abschnitt „Lokale Hilfedateien installieren“ entnehmen.

Wenn Sie zuvor noch nicht mit IBM Security Directory Integrator gearbeitet haben, sollten Sie das Handbuch *Erste Schritte* lesen und durcharbeiten, um sich mit den verwendeten Konzepten vertraut zu machen.

Haben Sie Vorversionen von IBM Security Directory Integrator verwendet, werden Ihnen in Abschnitt 3 des Handbuchs *Directory Integrator - Konfiguration* nützliche Kenntnisse über das neue IDE-Framework und -Layout vermittelt. Außerdem wird dort erläutert, wie Sie Ihre vorhandenen Konfigurationen importieren und öffnen können. Zudem erfahren Sie, wie der Server auch weiterhin zur Laufzeit das Konfigurationsmodell verwendet.

Migration

Unter dem hier aufgeführten Link erhalten Sie weitere Informationen zur Migration.

Falls Sie eine frühere Version von IBM Security Directory Integrator installiert haben, müssen Sie höchstwahrscheinlich bestimmte Aspekte der vorherigen Implementierung migrieren. Weitere Informationen zu den in diesem Fall erforderlichen Maßnahmen enthält Kapitel 5, „Migration“, auf Seite 67.

Lokale Hilfedateien installieren

Mit den hier aufgeführten Anweisungen können Sie die Dokumentation lokal installieren.

Anmerkung: Diese Komponente ist veraltet und wird in einer zukünftigen Version von IBM Security Directory Integrator entfernt.

Das Installationsprogramm von IBM Security Directory Integrator enthält neben der Java-API-Dokumentation, die durch Auswahl der Anzeige **Hilfe** -> **Willkommen** und des Links **JavaDocs** im Konfigurationseditor angezeigt werden kann, keine Benutzerdokumentation. IBM stellt die Benutzerdokumentation online im IBM Knowledge Center for IBM Security Directory Integrator zur Verfügung.

IBM Security Directory Integrator ist mit Code¹ ausgestattet, der Ihnen die kontextabhängige Onlinehilfe zur Verfügung stellt und den Sie aus dem Konfigurationseditor heraus starten können. Dieser Code wird standardmäßig in die Dokumentation aus der online verfügbaren Produktdokumentationswebsite (siehe oben) aufgelöst. Sie können die Dokumentation jedoch lokal installieren, damit Sie nicht vom Internet abhängig sind, wenn Sie sie lesen wollen.

Zur lokalen Installation der Dokumentation müssen Sie die folgenden Schritte ausführen:

- Der Code für die Verarbeitung der Dokumentationsdateien (also die Komponente "IBM Hilfesystem der Benutzerschnittstelle auf der Basis der Eclipse-Technologie (Built on Eclipse)") wird nicht standardmäßig installiert. Um das Hilfesystem zu installieren, müssen Sie eine benutzerdefinierte Installation ausführen und die Komponente des Hilfesystems in Ihrer vorhandenen IBM Security Directory Integrator-Installation installieren.
- Alle Handbücher sind zusammen in einem einzigen komprimierten Verzeichnis gespeichert. Wenn dieses Verzeichnis dekomprimiert wird, enthält es ein *Eclipse-Dokument-Plug-in*.
- Alle Handbücher können in ihrer komprimierten Form aus dem IBM Knowledge Center for IBM Security Directory Integrator heruntergeladen werden. Klicken Sie auf der Begrüßungsseite des aktuellen Release auf den Link *Information Center-Plug-ins* unter der Spalte *More information*.
- Das gesamte Dokumentationspaket (*di_plug-ins-7.2.0.1.zip*) muss an der korrekten Position dekomprimiert werden, also im Ordner *tdi-installationsverzeichnis/ibm_help/eclipse/plugins* (alternativ kann das Paket auch an einer anderen Position dekomprimiert und anschließend in den korrekten Ordner verschoben werden). Das Paket enthält die eigentliche Dokumentation zu IBM Security Directory Integrator unter *com.ibm.IBMDI.doc_7.2.0.1* sowie eine Reihe anderer Verzeichnisse, deren Namen mit ".doc" enden. Alle diese Verzeichnisse müssen sich auf derselben vorgenannten Ebene *plugins* befinden.
- Die Position der Dokumentation, auf die der Konfigurationseditor zuzugreifen versucht, ist in der Datei *global.properties* festgelegt, die sich im Ordner "etc" des Installationsverzeichnisses von IBM Security Directory Integrator befindet, oder in der Datei *solutions.properties* im Lösungsverzeichnis. Diese Position zeigt standardmäßig auf die online verfügbare Produktdokumentation. Wenn Sie die folgenden Zeilen ändern, wird bei der nächsten Ausführung des Konfigurationseditors und beim Starten der Hilfe das lokale Hilfesystem verwendet.

```
com.ibm.di.helpHost=publib.boulder.ibm.com  
com.ibm.di.helpPort=80
```

in:

```
com.ibm.di.helpHost=localhost  
com.ibm.di.helpPort=9999
```

- Die Position des Dokumentationservers, auf den AMC zuzugreifen versucht, ist in der Datei *web.xml* für AMC festgelegt. Öffnen Sie die Datei "web.xml", die sich im Ordner WEB-INF der Webanwendung "tdiamc" befindet, und listen Sie die IP-Adresse (oder den Hostnamen) und den Port des Hilfeservers für beide Vorkommen der folgenden Attribute auf: *InfocenterHostName* und *InfocenterPort*.

Nachdem Sie die Dokumentation wie oben skizziert im Verzeichnis *plugins* installiert haben, können Sie die Dokumentation auf diesem Computer auch für andere IBM Security Directory Integrator-Installationen in Ihrer Umgebung bereitstellen.

1. Das Hilfesystem basiert auf Eclipse-Technologie (Powered by Eclipse™). Weitere Informationen finden Sie unter der Adresse <http://www.eclipse.org>.

Das Verzeichnis *tdi-installationsverzeichnis/ibm_help* enthält eine Reihe von Dateien ".bat" (Windows) oder ".sh" (UNIX/Linux), die dies ermöglichen.

Lokale Task für Dokumentation starten und stoppen

Zunächst muss eine lokale Task für die Dokumentation gestartet werden.

Script "IC_start.bat" bzw. "IC_start.sh"

Wenn Sie dieses Script ausführen, startet es ein Information Center über die Adresse "*http://ihre_ip-adresse:9999*".

Durch eine Bearbeitung dieser Datei können Sie den Standardwert 9999 für die Portnummer ändern. Wenn Sie den Port beispielsweise in 80 ändern wollen, ändern Sie die Angabe "-port 9999" in "-port 80". Bei den Clients, die versuchen, auf dieses Information Center zuzugreifen, muss die Portnummer mit einer anderen Eigenschaft in der Datei *global.properties* oder *solution.properties* übereinstimmen, nämlich mit der Eigenschaft *com.ibm.di.helpPort*, deren Standardwert 80 lautet. Auch die Eigenschaft *com.ibm.di.helpHost* sollte einen Wert wie etwa *ip-adresse_für_information_center* aufweisen. Hierbei steht *ip-adresse_für_information_center* für die Adresse des lokalen Information Centers. Damit AMC dieses Information Center finden kann, müssen Sie zusätzlich die Attribute der Parameter *InfoCenterHostname* und *InfoCenterPort* in der AMC-Konfigurationsdatei *web.xml* in die obigen Werte ändern.

Script "IC_stop.bat" bzw. "IC_stop.sh"

Dieses Script stoppt das Hilfesystem, ein Java-Programm, das das lokale Information Center bereitstellt.

Script "help_start.bat" bzw. "help_start.sh"

Dieses Script hat eine ähnliche Funktionsweise wie das Script "IC_start", verwendet jedoch für die Portnummer einen Zufallswert und startet außerdem einen lokalen Browser, in dem die Startseite angezeigt wird. Da es sich bei der Portnummer um einen Zufallswert handelt, ist dieses Script ausschließlich zur Verwendung auf dem lokalen Computer geeignet.

Script "help_stop.bat" bzw. "help_stop.sh"

Dieses Script stoppt die lokale Java-Task, die durch das Script "help_start" gestartet wurde.

AMC in angepasster ISC SE-Instanz oder IBM Dashboard Application Services Hub implementieren

Mit den hier aufgeführten Anweisungen können Sie AMC nach einer Verzögerung implementieren.

Falls Sie ausgewählt haben, die Implementierung von AMC in ISC zu verzögern, und dies nun vornehmen möchten, führen Sie die folgenden Schritte aus:

- Führen Sie die folgenden Scripts aus:

```
tdi-installationsverzeichnis/bin/setISCHome.bat(sh) isc-position  
tdi-installationsverzeichnis/bin/amc/install.bat(sh)  
tdi-installationsverzeichnis/bin/amc/setAMCRoles.bat(sh) benutzername
```

Anmerkung:

1. Der Aufruf des Scripts *setAMCRoles* ist für ISC SE und ISC AE optional. Wenn es ausgeführt wird, sollte für *benutzername* ein bereits in der ISC/IBM

WebSphere Application Server-Umgebung vorhandener Benutzername angegeben werden. Alternativ können Sie auch die ISC-Konsole verwenden (insbesondere die Anzeige für die Berechtigung des Konsolbenutzers), um eine der mit AMC bereitgestellten Rollen ("SDI AMC Admin" und "SDI AMC User") einem Benutzer manuell zuzuordnen.

2. Weitere Informationen zu AMC-Rollen enthält der Abschnitt „AMC in ISC“ auf Seite 273.
- Ändern Sie die Datei `amc.properties` so, dass in den Zeilen, die `"am.api.port"` und `"amc.help.port"` angeben, geeignete Portwerte verwendet werden. Für ISC SE befindet sich diese Datei im Verzeichnis `isc-position/runtime/isc/eclipse/plugins/AMC_7.2.0.0/`. Für IBM Dashboard Application Services Hub befindet sich diese Datei im Verzeichnis `isc-position/systemApps/isclite.ear/tdiamc.war`.

Falls Sie eine angepasste IBM Dashboard Application Services Hub-Instanz für die Implementierung von AMC verwenden wollen und nun mit der Implementierung beginnen können, führen Sie den folgenden Schritt aus:

- Führen Sie das folgende Script aus:

```
tdi-installationsverzeichnis/bin/amc/setAMCRoles.bat(sh) benutzername
```

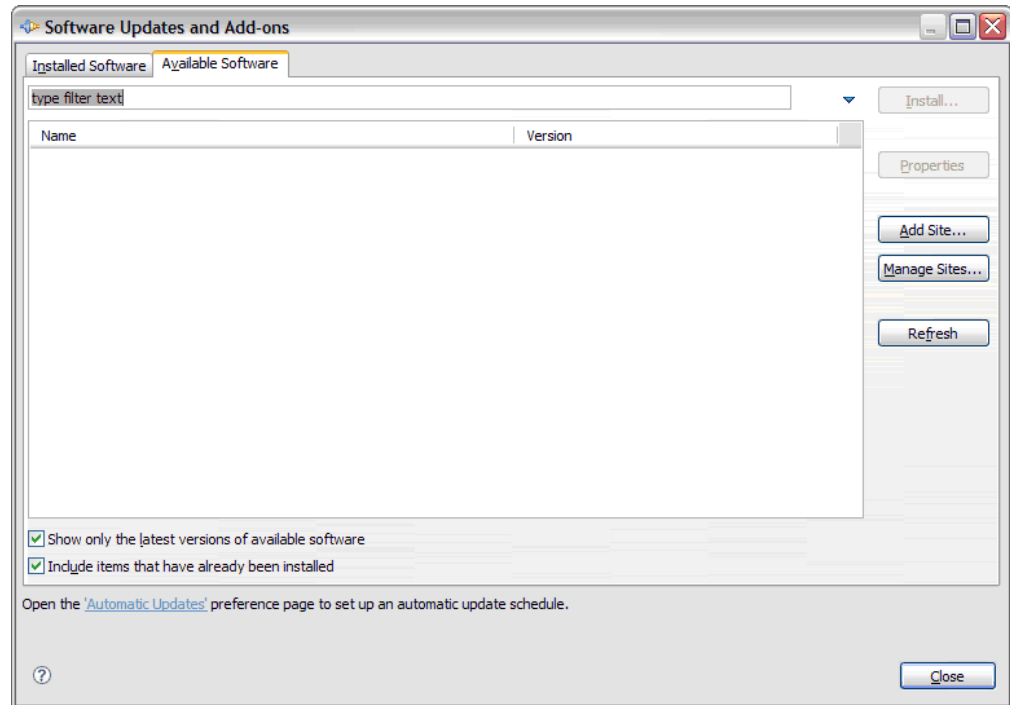
Anmerkung:

1. Der Aufruf des Scripts `setAMCRoles` ist für ISC SE und ISC AE optional. Wenn es ausgeführt wird, sollte für *benutzername* ein bereits in der ISC/IBM WebSphere Application Server-Umgebung vorhandener Benutzername angegeben werden. Alternativ können Sie auch die ISC-Konsole verwenden (insbesondere die Anzeige für die Berechtigung des Konsolbenutzers), um eine der mit AMC bereitgestellten Rollen ("SDI AMC Admin" und "SDI AMC User") einem Benutzer manuell zuzuordnen.
2. Weitere Informationen zu AMC-Rollen enthält der Abschnitt „AMC in ISC“ auf Seite 273.

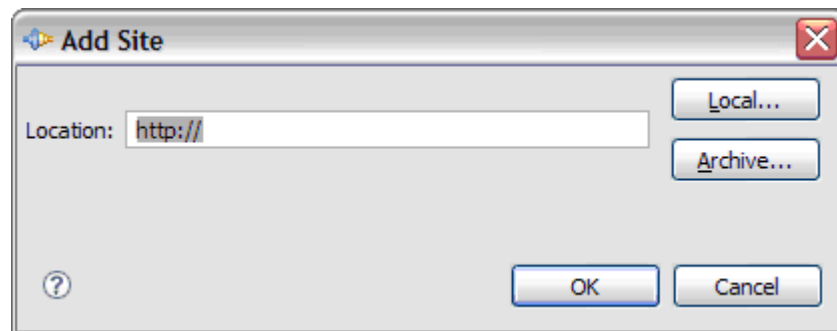
Installation oder Aktualisierung mit Eclipse-Aktualisierungsmanager vornehmen

Mit den hier aufgeführten Anweisungen können Sie den Eclipse-Aktualisierungsmanager verwenden.

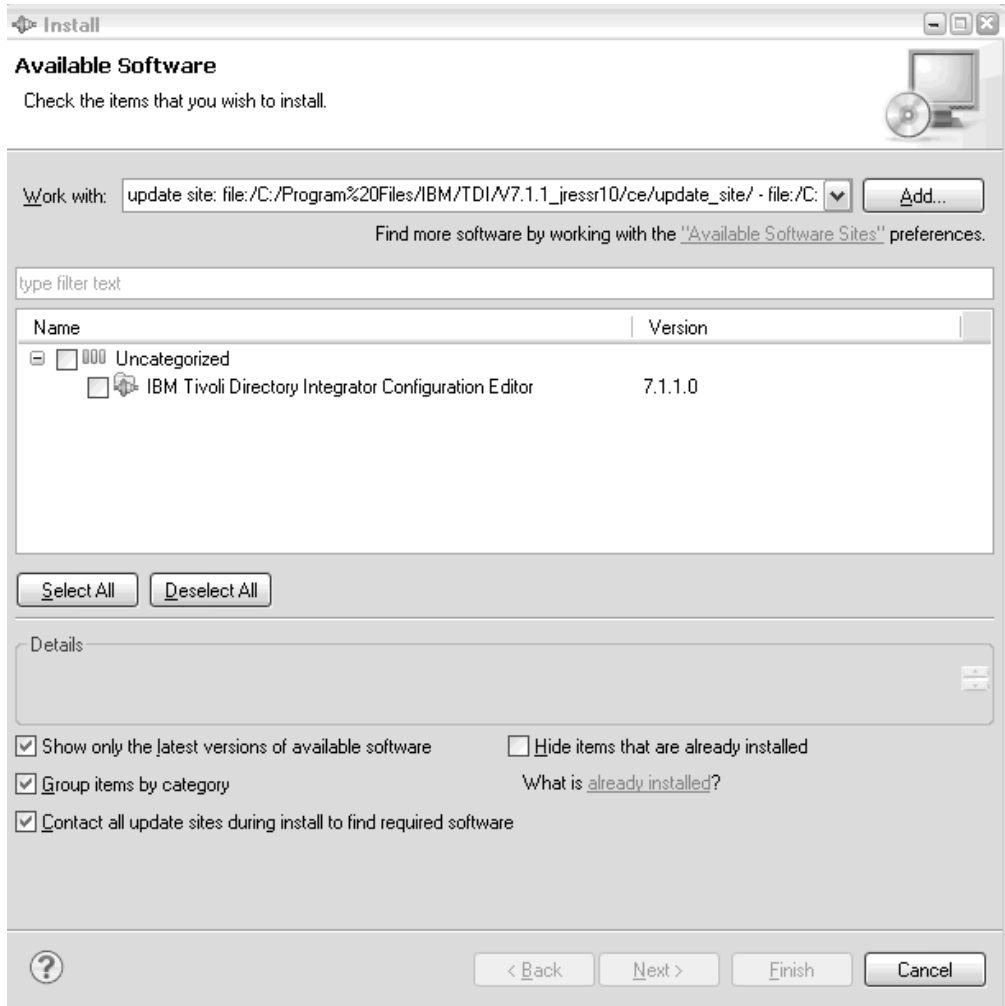
Die Rich-Client-Plattform von IBM Security Directory Integrator enthält eine vollständige Laufzeitumgebung für die Ausführung des Konfigurationseditors von IBM Security Directory Integrator. Es ist jedoch möglich, das Eclipse-Plug-in von IBM Security Directory Integrator in einer vorhandenen Eclipse-Installation zu installieren. Hierzu wird der Eclipse-Aktualisierungsmanager (Eclipse-Update-Manager) verwendet. Öffnen Sie in Eclipse den Eclipse-Aktualisierungsmanager über das Menü **Hilfe**.



Bevor das IBM Security Directory Integrator-Plug-in installiert wird, ist es sinnvoll, eine neue Aktualisierungssite hinzuzufügen. Wählen Sie die Schaltfläche **Site hinzufügen...** aus und geben Sie die Position der Aktualisierungssite an.



Wählen Sie je nach Position der Aktualisierungssite die geeignete Aktion aus. Im vorliegenden Beispiel wird ein Verzeichnis des lokalen Dateisystems verwendet. Nach Auswahl der Schaltfläche **Lokal** werden Sie aufgefordert, ein Verzeichnis auszuwählen, das dann im Eingabefeld für die Position eingefügt wird. Sobald Sie die Schaltfläche **OK** auswählen, sollten die neue Aktualisierungssite und die Aktualisierungen verfügbar sein:



Wählen Sie die Plug-ins aus, die installiert werden sollen, und wählen Sie **Installieren** aus. Während der Software-Update-Manager Ihre Installation aktualisiert, werden Sie möglicherweise aufgefordert, die Installation zu bestätigen. Außerdem werden Sie nach der Installation normalerweise aufgefordert, die Workbench erneut zu starten. Nach Abschluss der Installation sollte IBM Security Directory Integrator auf der Registerkarte **Installierte Software** angezeigt werden.

Schritte für den Installationsabschluss

Mit den hier aufgeführten Anweisungen können Sie am Installationsabschluss bestimmte Schritte ausführen.

Wenn der Konfigurationseditor wie bei der zuvor beschriebenen Prozedur in einer anderen Eclipse-Installation als Plug-in installiert wird, müssen eine Reihe spezieller Eigenschaften definiert werden, damit das *SDI-Ladeprogramm* enthalten ist. Das Ladeprogramm für IBM Security Directory Integrator ist ein Fragment von `org.eclipse.osgi`, mit dem das Laden von Klassen für den Konfigurationseditor bereitgestellt wird.

```
# TDI class loader
osgi.framework.extensions=com.ibm.tdi.loader
osgi.hook.configurators.include=com.ibm.tdi.loader.TDIClassLoaderHook
TDI_HOME_DIR=c:/Program Files/IBM/TDI/V7.2
```

Beachten Sie, dass die Eigenschaft TDI_HOME_DIR auf eine vorhandene IBM Security Directory Integrator-Serverinstallation zeigen muss, da der Konfigurationseditor in der Lage sein muss, viele Java-Klassen von IBM Security Directory Integrator-Komponenten abzufragen, damit eine ordnungsgemäße Funktionsweise erreicht wird. Diese Installation wird zudem verwendet, um den lokalen Entwicklungsserver zu erstellen, den der Konfigurationseditor verwendet. Das obige Fragment zeigt den Installationsstandardwert für Windows. Aktualisieren Sie diesen Wert Ihrer Umgebung entsprechend.

Es gibt mehrere Verfahren, um diese Eigenschaften festzulegen. Eines besteht darin, die Datei `configuration/config.ini` der Eclipse-Installation zu aktualisieren.

Anmerkung: Nachdem Sie den Konfigurationseditor in Eclipse installiert und konfiguriert haben, kann es sein, dass Abhängigkeitsprobleme auftreten. Zur Lösung solcher Probleme können Sie einen technischen Hinweis (Technote) hinzuziehen, der zu diesem Thema veröffentlicht wurde.

Deinstallation

Sie können IBM Security Directory Integrator vollständig deinstallieren oder nur bestimmte Komponenten deinstallieren.

Deinstallationsprogramm starten

Zum Starten des Deinstallationsprogramms müssen bestimmte Schritte ausgeführt werden. Nachstehend erhalten Sie Informationen dazu.

Informationen zu diesem Vorgang

Zur Deinstallation von IBM Security Directory Integrator müssen Sie zuerst das Deinstallationsprogramm starten:

Anmerkung: Stoppen Sie vor der Deinstallation alle Komponenten, die Sie entfernen wollen, beispielsweise eine Instanz der IBM Security Directory Integrator-Laufzeit, einen aktiven AMC-Service oder ein Plug-in der Komponente "Password Synchronization Plug-ins". Werden aktive Komponenten nicht gestoppt, kann dies dazu führen, dass einige Dateien nicht entfernt werden und nach der Deinstallation weiter vorhanden sind. Unter Windows ist möglicherweise ein Neustart erforderlich. Außerdem ist der IBM Security Directory Integrator-Webverwaltungsservice (AMC) eventuell weiter in der Liste "Dienste" vorhanden und muss manuell gelöscht werden.

Vorgehensweise

1. Navigieren Sie zum Verzeichnis `_uninst` von IBM Security Directory Integrator.
Beispiel: `installationspfad/_uninst`
2. Starten Sie das Deinstallationsprogramm, indem Sie die ausführbare Datei für die Deinstallation ausführen.
Bei Windows-Plattformen heißt die ausführbare Datei für die Deinstallation `uninstaller.exe`. Bei allen anderen Plattformen hat die ausführbare Datei für die Deinstallation den Namen "uninstaller.bin".
3. Führen Sie nun die Aktionen aus, die im Abschnitt „Anzeigenfolge bei der Deinstallation“ auf Seite 34 beschrieben sind.

Ergebnisse

Achtung: Während einer Deinstallation werden eine Reihe von Verzeichnissen auf dem Computer geleert und entfernt. Es handelt sich um die folgenden Elemente:

- *tdi-installationsverzeichnis/lwi* - Es besteht die Möglichkeit, dass hier einige Dateien übrig bleiben oder dass Dateien durch die integrierte Webplattform erstellt werden, die vom Installationsprogramm nicht abgelegt werden. Dieses Verzeichnis wird bei einer Deinstallation gelöscht.
- *tdi-installationsverzeichnis/ce/eclipsece/features/com.ibm.tdi.*.jar*
- *tdi-installationsverzeichnis/ce/eclipsece/plugins/com.ibm.tdi.*.jar*
- *tdi-installationsverzeichnis/ce/eclipsece/configuration*
- *tdi-installationsverzeichnis/ce/update_site/features/com.ibm.tdi.*.jar* - Falls Komponenten hinzugefügt wurden, die mit dem Muster dieses Platzhalters übereinstimmen, werden sie gelöscht.
- *tdi-installationsverzeichnis/ce/update_site/plugins/com.ibm.tdi.*.jar* - Dito.
- *tdi-installationsverzeichnis/maintenance/BACKUP* - Dieses Verzeichnis wird möglicherweise von Update Installer erstellt.
- *tdi-installationsverzeichnis/_uninst/**, *tdi-installationsverzeichnis/amc/**, *tdi-installationsverzeichnis/osgi/**, *tdi-installationsverzeichnis/SCIM/** und *tdi-installationsverzeichnis/LDAPSync/**. Diese Verzeichnisse werden unabhängig von den Änderungen, die der Benutzer vorgenommen hat, gelöscht.

Alle Elemente, die Sie möglicherweise selbst in diesen Verzeichnissen gespeichert haben und die mit einem dieser Kriterien übereinstimmen, werden bei einer Deinstallation ebenfalls entfernt.

Unbeaufsichtigte Deinstallation ausführen

Mit den hier aufgeführten Anweisungen können Sie eine unbeaufsichtigte Deinstallation ausführen.

Um eine unbeaufsichtigte Deinstallation ausführen zu können, müssen Sie zuerst eine Antwortdatei generieren. Zur Generierung dieser Datei müssen Sie eine vollständige Deinstallation über die grafische Benutzerschnittstelle oder die Konsole ausführen und hierbei die Option `-options-record` angeben. Beispiel:
`tdi-installationsverzeichnis/_uninst/uninstaller.exe -r name_der_deinstallationsantwortdatei`. Die Antwortdatei wird in dem Verzeichnis erstellt, das Sie bei der Deinstallation angeben.

Anmerkung: Das Verzeichnis `tdi-installationsverzeichnis/examples/install` enthält eine Reihe von Beispielantwortdateien für verschiedene Installations- oder Deinstallationsszenarios.

Sobald die Antwortdatei erstellt wurde, können Sie mit dem folgenden Befehl eine unbeaufsichtigte Deinstallation ausführen: `tdi-installationsverzeichnis/_uninst/uninstaller.exe -f name_der_deinstallationsantwortdatei`.

Standardinstallationspositionen

IBM Security Directory Integrator wird an den folgenden Standardpositionen installiert:

Windows-Plattformen

C:\Program Files\IBM\TDI\V7.2

Linux- und UNIX-Plattformen

/opt/IBM/TDI/V7.2

Standardlösungsverzeichnis

Verwenden Sie die Datei *tdi-installationsverzeichnis\bin\setDefaultSoldir.bat*(sh) zum Festlegen eines neuen Standardlösungsverzeichnisses.

Wenn Sie beispielsweise den Befehl

```
setDefaultSoldir.bat C:\mysoldir
```

ausführen, wird `TDI_SOLDIR=C:\mysoldir` in der Datei *defaultSoldir.bat*(sh) festgelegt.

Der Wert in der Datei *defaultSoldir.bat*(sh) wird zur Angabe des Standardverzeichnisses verwendet, wenn Sie das Lösungsverzeichnis während der Installation von IBM Security Directory Integrator auswählen.

Das Wert des Lösungsverzeichnisses im Script *defaultSoldir.bat*(sh) wird vom TDI-Installationsprogramm festgelegt.

Der Standardwert wird basierend auf der Option geändert, die Sie auf der Seite für das Lösungsverzeichnis des Installationsprogramms angeben.

Die Datei *defaultSoldir.bat*(sh) wird vom IBM Security Directory Integrator-Server zum Abrufen der Position des Standardlösungsverzeichnisses verwendet.

Kapitel 3. Update Installer

Verwenden Sie IBM Security Directory Integrator Update Installer (**applyUpdates-.bat (sh)**) für die Installation von Fixpacks auf einer vorhandenen IBM Security Directory Integrator-Installation.

Das normale Installationsprogramm legt eine Datei namens `.registry` im Installationsverzeichnis ab, die die aktuelle Version der installierten Komponenten angibt. Im Verzeichnis "bin" der Installation wird ein Script namens `tdiSetBackupDir.bat` bzw. `tdiSetBackupDir.sh` erstellt, das die Position des Sicherungsverzeichnisses festlegt. Standardmäßig ist dies ein Verzeichnis namens "BACKUP" im Verzeichnis "maintenance". Durch eine Ausführung des Scripts `tdiSetBackupDir` können Sie die Sicherungsposition ändern. Heißt beispielsweise ein Fix "ifix1", befinden sich die Sicherungsdateien und -verzeichnisse in diesem Szenario unter `installationsverzeichnis/maintenance/BACKUP/ifix1`. Update Installer erbt den Namen des Sicherungsverzeichnisses während der Pflegeprozedur. Der Benutzer, der diese Prozedur für IBM Security Directory Integrator ausführt, muss über Schreibberechtigungen für die Installations- und Sicherungsverzeichnisse verfügen. Außerdem müssen Sie bedenken, dass das Deinstallationsprogramm bei einer vollständigen Deinstallation versucht, das Standardsicherungsverzeichnis zu löschen.

Das normale Installationsprogramm übernimmt auch die Pflege der Datei `.registry` beim Deinstallieren und Hinzufügen von Komponenten.

- Bei einer vollständigen Deinstallation wird die Datei `.registry` zusammen mit den anderen Dateien gelöscht.
- Während einer Teilinstallation werden nur die Komponenten, die deinstalliert werden, aus der Datei `.registry` gelöscht.
- Beim Hinzufügen von Komponenten wird die Datei `.registry` so aktualisiert, dass Sie die neu installierten Komponenten enthält.

Nach dem Hinzufügen einer Komponente müssen Sie unverzüglich alle gegenwärtig angewendeten Fixes installieren.

Update Installer besteht aus mehreren Java-Dateien. Um die ausführbare Java-Datei nicht angeben zu müssen, wird im Verzeichnis "bin" ein Wrapper-Script namens **applyUpdates.bat (sh)** erstellt. Dieses Script ermittelt mit vorhandenen Scripts die richtige JRE für die Verwendung und ruft den zugrunde liegenden Code auf. Die Syntax des Scripts lautet folgendermaßen:

```
applyUpdates -update fixdatei.zip [-clean [-silent]]
applyUpdates -rollback
applyUpdates -queryreg
applyUpdates -queryfix fix_file.zip
applyUpdate -enroll lizenzdatei.zip
applyUpdates -?
```

Die einzelnen Optionen sind nachstehend beschrieben:

-update

Diese Option wird verwendet, um ein Fixpack anzuwenden.

Der Name der komprimierten Datei mit dem Fixpack lautet `fix_file.zip`. Hierbei kann es sich um einen relativen oder einen absoluten Pfad handeln.

Die Option **-clean**, die nur für ein Fixpack verfügbar ist, löscht vor dem Anwenden des aktuellen Fixpacks alle gesicherten Dateien. Sie werden aufgefordert, das Löschen der alten Daten zu bestätigen. Die Option **-silent** unterdrückt die Bestätigungsaufforderung.

Wenn ein Fixpack erneut angewendet wird, beispielsweise nach dem Hinzufügen von neuen Funktionen, für die das Fixpack benötigt wird, wird die Option **-clean** ignoriert.

Wenn Sie die Option **-clean** zum Bereinigen der Sicherungsverzeichnisse verwenden, kann ein Rollback nur auf einer einzelnen Ebene durchgeführt werden.

-rollback

Mit dieser Option wird IBM Security Directory Integrator auf den Zustand zurückgesetzt, der vor dem Anwenden des neuesten Fix vorlag. Diese Daten werden standardmäßig im Verzeichnis *tdi-installationsverzeichnis/maintenance/BACKUP/FP##* gespeichert.

-queryreg

Diese Option zeigt die Komponenten an, die in der aktuellen Installation vorhanden sind, sowie alle angewendeten Fixes.

Es folgt ein Beispiel für eine Ausgabe:

```
Informationen aus der .registry-Datei in: C:\Program Files\IBM\TDI\V7.2
Edition: Identity
Version: 7.2.0.1
```

```
Anwendete Fixes
=====
SDI-7.2-FP0001(7.2.0.0)
```

```
Installierte Komponenten
=====
BASE
SERVER
CE
CE UPDATE
JAVADOCS
EXAMPLES
IEHS
EMBEDDED WEB PLATFORM
AMC
      Verzögert: falsch (false)
PLUGINS
```

-queryfix

Diese Option zeigt Informationen zum Fix aus der Datei *fix_file.zip* an.

Es folgt ein Beispiel für eine Ausgabe:

```
Informationen aus der Fixdatei: C:\fixes\SDI-7.2-FP0001.zip
Name: fixpack1

Erforderliche Mindestversion für die Anwendung des Fix: 7.2.0
Zulässige Maximalversion für die Anwendung des Fix: 7.2.0.1

Voraussetzungen
=====
Keine

Betroffene Komponenten
=====
BASE
CE
EXAMPLES
```

Die komprimierte Datei mit dem Fix *fix_file.zip* enthält eine Manifestdatei mit der Erweiterung *.manifest*, in der Informationen zum Anwenden des Fix gespeichert sind.

-enroll

Diese Option wird verwendet, um eine leere Datei, eine Testlizenz oder eine Volllizenz zu registrieren. Sie können diese Option auch für ein Upgrade des Produkts von der Test- auf eine Vollversion verwenden. Diese Option wird jedoch von allen Installationsprogrammen verwendet. Die zu registrierende Lizenz ist in der komprimierten Datei enthalten und wird als Argument übergeben.

Führen Sie den folgenden Befehl aus, um für das Produkt ein Upgrade von der Test- auf die Vollversion durchzuführen:

```
applyupdates -enroll lizenzdatei.zip
```

Anmerkung: Die Datei *lizenzdatei.zip* kann vom IBM Sales oder Support Team bereitgestellt werden.

Führen Sie die folgenden Schritte aus, wenn der IBM Security Directory Integrator-Server aufgrund einer bereits registrierten Lizenz und aufgrund eines Hardwarefehlers nicht gestartet wird:

1. Führen Sie für die Datei *tdi-ausgangsverzeichnis.registry* ein Backup durch.
2. Entfernen Sie aus der Datei *.registry* den im Tag LICENSE definierten Wert. Beispiel: `<LICENSE> Full </LICENSE>`
3. Entfernen Sie aus der Datei *tdi-ausgangsverzeichnis\license* die vorhandene knotengebundene Datei.
4. Führen Sie den Befehl **applyUpdates** aus.

```
tdi-ausgangsverzeichnis\bin\applyUpdates -enroll lumfile.zip
```

In Abhängigkeit von der Anzahl der Lizenzen in der komprimierten Datei werden in der Ergebnismeldung die angewendeten Lizenzen wie im folgenden Beispiel dargestellt angezeigt.

```
./applyUpdates.sh -enroll /tmp/TDI_LUM_FULL.zip  
CTGDK0059I Die Testlizenz wurde erfolgreich registriert.  
CTGDK0062I Die Volllizenz wurde erfolgreich registriert.
```

Anmerkung: Die Datei *lum_file.zip* wird nach der Ausführung des Befehls gelöscht.

-? Diese Option stellt Informationen zur Verwendung bereit.

Datei ".registry"

Mit der Datei *.registry* können Sie die Version aller gegenwärtig auf dem System in einem bestimmten Installationsverzeichnis installierten IBM Security Directory Integrator-Komponenten ermitteln.

Die Datei *.registry* befindet sich im Installationsverzeichnis. Diese Datei wird anfänglich vom Installationsprogramm erstellt und basiert auf den zum Installationszeitpunkt ausgewählten Optionen.

Wenn ein Fix installiert wird, werden die gesicherten Dateien innerhalb des Sicherungsverzeichnisses in einem Verzeichnis mit dem Namen des Fix gespeichert. Wurde das Fixpack erfolgreich installiert, erhält die Datei *.registry* zusätzliche Einträge, die die Änderungen darstellen, die durch ein Fix an den Komponenten vorgenommen wurden. Im Abschnitt FIXES der Datei *.registry* sind die Fixes angegeben, die angewendet wurden. Für jede Komponente sind Einträge enthalten, in denen angegeben ist, von welchen angelegten Fixes sie geändert wurde. Wenn

das Fixpack jedoch nicht installiert werden konnte, wird die Datei `.registry` nicht aktualisiert und enthält dieselben Einträge wie vor der Anwendung des Fixpacks.

Update Installer erkennt die folgenden Komponenten:

- BASE
- SERVER
- Konfigurationseditor
- CE_UPDATE
- JAVADOCS
- EXAMPLES
- IEHS
- Integrierte Webplattform
- Administration and Monitoring Console (AMC)
- PLUGINS: Für die Plug-in-Komponente sind möglicherweise einige Schritte erforderlich, die nicht durch Update Installer ausgeführt werden können und manuell durchgeführt werden müssen. Wenn in den Dateien des Typs `pwsync.props` Änderungen vorgenommen werden müssen, muss dies manuell geschehen. Befolgen Sie die Anweisungen in der Datei `manual_readme.txt` im Fixpack. Sie müssen diese Schritte nach der Installation des Fixpacks durchführen, jedoch vor allen Schritten nach Installationsabschluss, die in den folgenden Abschnitten aufgeführt sind. Die Readme-Datei enthält eine Warnung, dass Update Installer nur Dateien enthält, die vom Installer für die Installation verwendet werden. Dateien, die Sie selbst kopiert haben, müssen manuell aktualisiert werden. Dies ist in den nachfolgenden Schritten für den Installationsabschluss beschrieben. Sie müssen diese Schritte wie angegeben vor und nach der Installation des Fixpacks durchführen. Diese Schritte sind nur dann erforderlich, wenn Sie die entsprechenden Kennwortsynchronisationsprogramme bei den Zielsystemen registriert haben.

Password Synchronizer für Windows

Installationsvorbereitung

Keine

Installationsabschluss

Diese Schritte sind nur erforderlich, wenn das Fixpack eine Aktualisierung für die DLL des Kennwortsynchronisationsprogramms enthält.

1. Entfernen Sie den Namen der DLL-Datei für das Kennwortsynchronisationsprogramm aus dem folgenden Registryschlüssel:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA\
Notification Packages
```

Die DLL-Datei heißt `tdipwflt` auf 32-Bit-Windows-Systemen und `tdipwflt_64` auf 64-Bit-Windows-Systemen.

2. Starten Sie Windows erneut, damit der Prozess für die lokale Sicherheitsberechtigung (Local Security Authority, LSA) die DLL-Datei für das Kennwortsynchronisationsprogramm entlädt.
3. Ersetzen Sie die DLL-Datei im Windows-Ordner `system32` durch die Datei aus der Installation des Kennwortsynchronisationsprogramms. Der Pfad der DLL-Datei nach der Installation lautet - je nach Windows-Version - entweder

installationsverzeichnis/pwd_plugins/windows/tdipwflt.dll oder *installationsverzeichnis/pwd_plugins/windows/tdipwflt_64.dll*.

4. Fügen Sie den Namen der DLL-Datei (ohne die Erweiterung .dll) im Registryschlüssel hinzu:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA\Notification Packages

5. Starten Sie Windows erneut, damit der Prozess "LSA" die neue DLL-Datei lädt.

Nach dem Neustart muss das Kennwortsynchronisationsprogramm normal unter Verwendung der aktualisierten Dateien ausgeführt werden.

Password Synchronizer für IBM Security Directory Server

Installationsvorbereitung

1. Stoppen Sie Directory Server.
2. Stoppen Sie den Proxy-Prozess für das Kennwortsynchronisationsprogramm. Verwenden Sie hierzu das Befehlszeilendienstprogramm **stopProxy**. Dieser Schritt ist notwendig, weil das Kennwortsynchronisationsprogramm für IBM Security Directory Server seinen Proxy nicht automatisch stoppt, wenn es beendet wird.

Installationsabschluss

Keine

Password Synchronizer für Sun Directory Server

Installationsvorbereitung

Stoppen Sie Directory Server.

Installationsabschluss

Keine

Password Synchronizer für PAM

Installationsvorbereitung

Vermeiden Sie nach Möglichkeit Kennwortänderungen, während die Aktualisierung stattfindet. Heben Sie andernfalls die Registrierung für das Kennwortsynchronisationsprogramm in der PAM-Konfigurationsdatei auf.

Installationsabschluss

Falls Sie vor der Aktualisierung die Registrierung des Kennwortsynchronisationsprogramms aufgehoben haben, nehmen Sie die Registrierung erneut vor.

Weitere Informationen finden Sie im Abschnitt zu *Plug-ins für die Kennwortsynchronisation* in der IBM Security Directory Integrator-Dokumentation.

Password Synchronizer für Domino

Installationsvorbereitung

Installationsabschluss

Befolgen Sie die Anweisungen für den Installationsabschluss im Abschnitt zum *Kennwortsynchronisationsprogramm für Domino HTTP* im Abschnitt zu *Plug-ins für die Kennwortsynchronisation* in der IBM Security Directory Integrator-Dokumentation.

Nehmen Sie anschließend eine Neukonfiguration des Domino-Plug-ins wie im Abschnitt zur *Bereitstellung eines einzelnen Domino-Servers* beschrieben vor.

Installation von Fixpacks

Befolgen Sie für die Installation von Fixpacks die Anweisungen in der Readme-Datei, die mit dem Fixpack bereitgestellt wird.

Falls eine Fixdatei einen Fix für eine Komponente enthält und diese Komponente auf dem System installiert ist, wird für die jeweilige Komponente eine Reihe von programmierten Aktionen ausgeführt.

Wenn außerhalb von Update Installer manuelle Schritte ausgeführt werden müssen, enthält die Readme-Datei für den Fix entsprechende Anweisungen.

Anmerkung: Vor der Aktualisierung eines Fixpacks müssen alle IBM Security Directory Integrator-Prozesse beendet werden.

Installationsabschluss: Wenn eine Ihrer vorhergehenden Installationen Federated Directory Server enthielt, müssen Sie die *.xml-Dateien aus dem Verzeichnis *sdi-lösungsverzeichnis/LDAPSync* in das Verzeichnis *sdi-lösungsverzeichnis/configs* kopieren.

Rollback durchführen

Während einer Rollback-Operation verwendet Update Installer Informationen, die zuvor während eines Fix abgelegt wurden, sowie gesicherte Dateien, um einen vorherigen Zustand wiederherzustellen.

Anmerkung: Vor dem Durchführen eines Rollbacks müssen alle IBM Security Directory Integrator-Prozesse beendet werden.

Die Rollback-Operation wird nicht für Dateien durchgeführt, für die Sie während der Fixpack-Installation eine manuelle Aktion ausgeführt haben.

Fehlerbehebung

Anhand der Update Installer-Protokolle können Sie mögliche Fehler beheben, die bei der Installation von Updates aufgetreten sind.

Update Installer erstellt eine Protokolldatei namens `updateinstaller.log` im Verzeichnis `installationsverzeichnis/logs`. Standardmäßig werden Nachrichten der Stufe INFO protokolliert. Durch eine Änderung der Datei `installationsverzeichnis/logsinstall_dir/etc/updateinstaller-log4j.properties` kann jedoch auch eine Protokollierung von Nachrichten der Stufe DEBUG festgelegt werden.

Kapitel 4. Unterstützte Plattformen

Unter dem hier aufgeführten Link erhalten Sie Informationen zu den unterstützten Betriebssystemen und Web-Browsern sowie zur Virtualisierungsunterstützung.

Lesen Sie den Abschnitt zu den „Softwarevoraussetzungen“ in der IBM Security Directory Integrator-Dokumentation unter http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDI.doc_7.2/sysreqs.html.

Kapitel 5. Migration

Nachstehend erhalten Sie Informationen zur Migration, deren Typen und Szenarios sowie zu verschiedenen migrierbaren Komponenten.

Im Kontext von IBM Security Directory Integrator kann der Begriff "Migration" verschiedene Bedeutungen haben:

- Vorbereitung relevanter Dateien (und deren Inhalts) für die Verwendung an einer neuen Position, auf derselben Maschine oder auf einer anderen Maschine oder
- Vorbereitung relevanter Dateien für die Verwendung mit einer neuen Version des Produkts

In der folgenden Tabelle sind die Migrationsszenarios zusammengefasst:

Tabelle 1. Migrationsszenarios

Quellen- & Zielversion identisch?	Quellen- & Zielinstallationspfad identisch?	Beschreibung des Szenarios
Nein	Nein	Dateien werden auf eine neue Version migriert, die an einer anderen Position installiert ist.
Nein	Ja	Dateien werden auf eine neue Version migriert, die an derselben Position installiert wird.
Ja	Nein	Dateien werden auf eine andere Installation derselben Version migriert.
Ja	Ja	Gesicherte Dateien werden an ihrer ursprünglichen Position wiederhergestellt.

Falls Sie sowohl auf eine neue Version als auch auf eine neue Position migrieren müssen, sollten Sie zuerst das Versionsupgrade durchführen, da im vorliegenden Handbuch die Positionsmigration nur für das aktuelle Release behandelt wird.

Das IBM Security Directory Integrator-Installationsprogramm kann Sie bei der Migration von IBM Security Directory Integrator 7.0, 7.1. und 7.1.1. auf IBM Security Directory Integrator Version 7.2 unterstützen.

Anmerkung: Ein direktes Upgrade von IBM Security Directory Integrator von Version 6.x oder einer älteren Version auf Version 7.2 wird nicht unterstützt. Sie müssen zunächst ein Upgrade von Version 6.x auf Version 7.1.1 und anschließend von Version 7.1.1 auf Version 7.2 durchführen.

Dateien auf andere Position migrieren

Vor der Auswahl der zu migrierenden Komponente bzw. Datei können Sie bestimmte Fragen lösen.

In diesem Abschnitt wird ausschließlich Version IBM Security Directory Integrator behandelt.

Welche Dateien müssen zur Verwendung an einer anderen Position nicht modifiziert werden?

Nachstehend finden Sie eine Liste der Dateitypen, die nicht modifiziert werden müssen.

- Benutzerkonfigurationen, Datendateien (.xml, .xsd, .xsl, .txt ...), Schlüsselspeicherdateien (.jks, ...), Zertifikatsdateien (.der, ...) usw.

Berücksichtigen Sie in diesem Zusammenhang auch die Auswirkungen, die im Abschnitt „Verschlüsselungsartefakte (Schlüssel, Zertifikate, Schlüsselspeicher, verschlüsselte Dateien) verwalten“ auf Seite 208 beschrieben sind.

- Scripts (Informationen zu Ausnahmen finden Sie unter „Welche Dateien sollten unter normalen Umständen nicht an einer anderen Position verwendet werden?“ auf Seite 69)
- Dateien ".bat", ".sh" und ".vbs"
- JAR-Dateien
- Native Binärdateien (.exe, .dll, .so usw.)
- Registrydatei der Server-API
- Server-Stashdatei
- Derby-Datenbanken:
Beispielsweise die Standarddatenbank des Systemspeichers (TDISysStore) und die AMC-Standarddatenbank (tdiamcdb).
Bitte beachten Sie, dass Sie die Datenbank in ihrer Gesamtheit (den ganzen Ordner) versetzen müssen. Auf keinen Fall sollten Sie die Dateien zweier Datenbanken mischen.
Bei komplexeren Szenarios verwenden Sie den JDBC-Connector, um Daten zwischen Datenbanken zu übertragen.
- Eigenschaftendateien von Action Manager (diese befinden sich im Ordner *tdi-installationsverzeichnis/bin/amc/ActionManager*)
- Konfigurationsdateien aus dem Ordner etc mit den folgenden Ausnahmen:
 - build.properties
 - global.properties
 - updateinstaller-log4j.properties
 - tdisrvctl-log4j.properties
- AMC-Konfigurationsdateien:
 - amc.properties
 - amcdbhandler.properties
 - amcdbschema.xml
 - idiamc.sth
 - Die AMC-Komponente ist veraltet und wird in einer zukünftigen Version von IBM Security Directory Integrator entfernt.

Welche Dateien müssen modifiziert werden, damit sie an einer anderen Position verwendet werden können?

Nachstehend finden Sie eine Liste der Dateitypen, die modifiziert werden müssen.

Positionsabhängige Dateien enthalten im Allgemeinen an einer oder mehreren Stellen den absoluten Pfad des Installationsordners. Diese Vorkommen müssen durch den neuen Verzeichnispfad ersetzt werden, damit die Datei für die neue Position relevant wird.

Die folgende Liste enthält die Dateien, bei denen eine Migration erforderlich ist, sowie Hinweise auf die Felder, die aktualisiert werden müssen. Diese Hinweise basieren auf dem Standardinhalt der entsprechenden Dateien. Falls Sie die Dateien modifiziert haben, kann es weitere Felder geben, die ebenfalls positionsabhängig sind und aktualisiert werden müssen.

- bin/amc/amcwinservice.ini:

Dies ist die Konfigurationsdatei für AMC, wenn AMC als Windows-Dienst registriert ist.

Aktualisieren Sie die Eigenschaften "WorkingDirectory", "StartCommand" und "StopCommand".

- global.properties/solution.properties:

Aktualisieren Sie die Eigenschaft "com.ibm.di.store.database".

Achten Sie auch auf die Eigenschaften "api.config.folder", "systemqueue.jmsdriver.param.mqe.file.ini" und "com.ibm.di.loader.userjars".

Falls Sie die Datei auf eine Installation migrieren, die einen anderen Verschlüsselungsschlüssel verwendet, lesen Sie die Angaben im Abschnitt „Verschlüsselungsartefakte (Schlüssel, Zertifikate, Schlüsselspeicher, verschlüsselte Dateien) verwalten“ auf Seite 208.

- etc/updateinstaller-log4j.properties:

Aktualisieren Sie die Eigenschaft "log4j.appender.Default.file".

- etc/tdisrvctl-log4j.properties:

Aktualisieren Sie die Eigenschaft "log4j.appender.Default.file".

- ibmdiservice.props:

Dies ist die Konfigurationsdatei für den Server, wenn dieser als Windows-Dienst registriert ist.

Aktualisieren Sie die Eigenschaften "path", "ibmdiroot" und "jvmRoot".

- pwsync.props:

Dies sind die Konfigurationsdateien der Kennwortsynchronisationsprogramme (Password Synchronizer).

Aktualisieren Sie die Eigenschaften "proxyStartExe", "logFile", "javaLogFile" und "mqe.file.ini".

Welche Dateien sollten unter normalen Umständen nicht an einer anderen Position verwendet werden?

Nachstehend finden Sie eine Liste der Dateitypen, die nach der Migration nicht verwendet werden sollten.

- Bestimmte Scripts

Diese Dateien dienen ausschließlich dazu, positionsspezifische Daten bereitzustellen.

Sie enthalten praktisch keine anderen Daten und wären daher an einer anderen Position nur von geringem Wert.

In dieser Hinsicht sind die folgenden Scripts aus dem Ordner *tdi-installationsverzeichnis/bin* relevant: "javaHome", "defaultSolDir", "backupDir", "tdiISCHome".

- Dateien ".reg"

Diese Dateien werden vom Kennwortsynchronisationsprogramm (Password Synchronizer) für Windows verwendet.

- IBM WebSphere MQ Everyplace-Warteschlangenmanagerdateien

Obwohl sich IBM WebSphere MQ Everyplace-Dateien nicht einfach auf eine andere Position migrieren lassen, können Sie Daten aus einer IBM WebSphere MQ Everyplace-Warteschlange an eine andere Warteschlange übertragen, indem Sie den JMS-Connector mit einem IBM WebSphere MQ Everyplace-JMS-Treiber verwenden.

- Arbeitsbereich des Konfigurationseditors

Um Directory Integrator-Projekte aus dem Arbeitsbereich des Konfigurationseditors wiederzuverwenden, exportieren Sie diese als Directory Integrator-Konfigurationen und importieren Sie sie in den neuen Arbeitsbereich.

- Datei "etc/build.properties"

Diese Datei enthält Zeit- und Versionsinformationen zum Release des Produkts.

Dateien mit verschlüsselten Daten migrieren

Unter dem hier aufgeführten Link erhalten Sie Informationen zum Migrieren von Dateien, die verschlüsselte Daten enthalten.

Informationen hierzu finden Sie unter „Verschlüsselungsartefakte (Schlüssel, Zertifikate, Schlüsselspeicher, verschlüsselte Dateien) verwalten“ auf Seite 208.

Dateien auf eine neuere Version migrieren

Sie können die Dateien auf drei Arten auf eine neuere Version migrieren. Nachstehend erhalten Sie Informationen dazu.

Migration mit Unterstützung durch das Installationsprogramm

Nachstehend erhalten Sie Informationen zur Funktionsweise des Installationsprogramms und dazu, welche Dateien automatisch bzw. manuell migriert werden.

Das Installationsprogramm migriert bestimmte Dateien bei einem Upgrade auf eine neuere Version automatisch. Bitte beachten Sie, dass das Installationsprogramm lediglich den Installationsordner von Directory Integrator berücksichtigt.

Alle anderen Lösungsordner als der Installationsordner müssen manuell migriert werden (bzw. unter Verwendung einiger der im Abschnitt „Migration mit Unterstützung durch ein Tool“ auf Seite 71 beschriebenen Tools).

Vom Installationsprogramm automatisch migrierte Dateien

- Migration von Version 6.0 auf Version 7.1
 - global.properties
 - Die Cloudscape-Datenbank (sofern sie für den Systemspeicher verwendet wird) wird auf Derby Version 10.5.3 migriert. Informationen hierzu finden Sie unter „Cloudscape-Datenbank auf Derby migrieren“ auf Seite 97.
 - pwsync.props (für jedes installierte Kennwort-Plug-in)
- Migration von Version 6.1.x auf Version 7.1
 - global.properties
 - AMC-Datenbank
 - amc.properties
 - am_config.properties
 - pwsync.props (für jedes installierte Kennwort-Plug-in)
- Migration von Version 7.0 auf Version 7.1
 - global.properties
 - pwsync.props (für jedes installierte Kennwort-Plug-in)
- Migration von Version 7.1 auf Version 7.1.1
 - global.properties
 - solution.properties (sofern im Standardlösungsverzeichnis vorhanden)
 - pwsync.props (für jedes installierte Kennwort-Plug-in)

- Migration von Version 7.1.1 auf Version 7.2
 - etc\reconnect.rules
 - etc\derby.properties
 - etc\jlog.properties
 - etc\log4j.properties
 - etc\tdisrvctl-log4j.properties
 - etc\tdimigbl-log4j.properties
 - etc\updateinstaller-log4j.properties
 - etc\it_registry.properties
 - etc\tp.xml
 - etc\activemq.xml
 - etc\global.properties
 - solution.properties (sofern im Standardlösungsverzeichnis vorhanden)
 - pwsync.props (für jedes installierte Kennwort-Plug-in)
 - Die AMC-Datenbank (mit den folgenden Tools kann ein Backup für AMC durchgeführt werden: bin/backupam.bat, bin/backupamc.bat und bin/backupamcdb.bat)
 - AMC_7.2.0.0\amc.properties
 - AMC_7.2.0.0\conf\amcdbhandler.properties
 - AMC_7.2.0.0\conf\logging.properties
 - bin\amc\ActionManager\am_config.properties
 - bin\amc\ActionManager\am_logging.properties

Manuell zu migrierende Dateien

Alle im Abschnitt „Manuelle Migration“ auf Seite 72 genannten Dateien, mit Ausnahme der im ersten Unterabschnitt (*„Eigenschaftendateien“*) aufgeführten Dateien.

Migration mit Unterstützung durch ein Tool

Sehen Sie die Liste der Tools durch, die bei einer durch das Installationsprogramm unterstützten Migration verwendet werden.

Die folgenden Tools werden vom Installationsprogramm bei einer durch das Installationsprogramm unterstützten Migration verwendet. Sie können diese Tools zur manuellen Migration einsetzen.

Migration von Eigenschaftendateien

- global.properties:
Verwenden Sie das Tool "tdimigbl" aus dem Ordner *tdi-installationsverzeichnis/bin* (siehe „Dateien "global.properties" und "solution.properties" mit dem Migrationstool migrieren“ auf Seite 99).
- amc.properties:
Verwenden Sie das Tool "tdimigamc" aus dem Ordner *tdi-installationsverzeichnis/bin/amc* (siehe „Befehlszeilendienstprogramme für AMC und Action Manager“ auf Seite 321). Die AMC-Komponente ist veraltet und wird in einer zukünftigen Version von IBM Security Directory Integrator entfernt.
- am_config.properties:

Verwenden Sie das Tool "tdimigam" aus dem Ordner *tdi-installationsverzeichnis/bin/amc* (siehe „Befehlszeilendienstprogramme für AMC und Action Manager“ auf Seite 321).

- *pwsync.props* (für jedes installierte Kennwort-Plug-in):
Verwenden Sie das Tool "migpwsync" aus dem Ordner *tdi-installationsverzeichnis/pwd_plugins/bin* (siehe „Eigenschaftendateien der Kennwort-Plug-ins mit dem Migrationstool migrieren“ auf Seite 100).

Migration der AMC-Datenbank

Verwenden Sie die Tools "backupamcdb"/"restreamcdb" aus dem Ordner *tdi-installationsverzeichnis/bin/amc* (siehe „Befehlszeilendienstprogramme für AMC und Action Manager“ auf Seite 321). Die AMC-Komponente ist veraltet und wird in einer zukünftigen Version von IBM Security Directory Integrator entfernt.

Migration des Cloudscape-Systemspeichers (nur bei Version 6.0)

Lesen Sie die detaillierteren Anweisungen im Abschnitt „Cloudscape-Datenbank auf Derby migrieren“ auf Seite 97.

Manuelle Migration

Nachstehend erhalten Sie Informationen zum Start der manuellen Migration. Außerdem finden Sie eine umfassende Liste mit den Eigenschaften, die in IBM Security Directory Integrator-Versionen geändert, hinzugefügt und gelöscht worden sind.

Kopieren Sie Ihre Konfigurationsdateien und alle anderen angepassten Dateien, einschließlich der Derby-Datenbanken, aus dem alten Installationsverzeichnis in das neue Installationsverzeichnis. IBM Security Directory Integrator unterstützt ein Lösungsverzeichnis. Es wird empfohlen, die Konfigurationsdateien, Eigenschaftendateien, Derby-Datenbanken usw. in ein solches Lösungsverzeichnis und nicht in das Installationsverzeichnis der Version von IBM Security Directory Integrator zu kopieren.

Nachdem Sie die oben genannten Objekte an eine neue Position kopiert haben, können Sie deren Inhalt migrieren und für die Verwendung mit IBM Security Directory Integrator anpassen. Die entsprechenden Anweisungen finden Sie in den folgenden Abschnitten:

1. Eigenschaftendateien
2. Konfigurationen
3. Angepasste Scripts
4. Hinzugefügte oder ersetzte JAR-Dateien in der Installation
5. Konfigurationen der Kennwortsynchronisationsprogramme (Password Synchronizer)

Anmerkung: Sandboxdaten sind versionspezifisch. Dies bedeutet, dass Daten, die unter einer Vorversion aufgezeichnet wurden, in Version 7.2 nicht wiedergegeben werden können.

Eigenschaftendateien

- *global.properties*:

In den folgenden Tabellen sind die Eigenschaften aufgeführt, die in verschiedenen Versionen von IBM Directory Integrator gelöscht, geändert oder hinzugefügt wurden:

Tabelle 2. Gelöschte und geänderte Eigenschaften

Eigenschaften in 'global/solution.properties'	Geändert, gelöscht oder hinzugefügt	Anmerkungen
web.server.ssl.on	*GEÄNDERT*	Weitere Informationen zu dieser Eigenschaft finden Sie in der Tabelle <i>Neue Eigenschaften in Version 7.1.1</i> . Ab IBM Security Directory Integrator Version 7.2 ist der Standardwert dieser Eigenschaft true. Beispiel: web.server.ssl.on=true
{protect}-dashboard.auth.user.admin	*HINZUGEFÜGT*	Diese Eigenschaft wurde in IBM Security Directory Integrator Version 7.2 hinzugefügt. Sie wird zur Angabe des Benutzernamens und Kennworts für Federated Directory Server verwendet. Der Standardwert dieser Eigenschaft ist admin. Beispiel: {protect}-dashboard.auth.user.admin=admin Das folgende Beispiel zeigt, wie mehrere Benutzeranmeldeaccounts für Federated Directory Server angegeben werden: {protect}-dashboard.auth.user.admin=admin {protect}-dashboard.auth.user.user1=user1passwd {protect}-dashboard.auth.user.user2=user2passwd
dashboard.auth.localhost	*GEÄNDERT*	Ab IBM Security Directory Integrator Version 7.2 ist der Standardwert dieser Eigenschaft properties.
dashboard.auth.remote	*GEÄNDERT*	Ab IBM Security Directory Integrator Version 7.2 ist der Standardwert dieser Eigenschaft properties.
com.ibm.di.server.NIST.on	*HINZUGEFÜGT*	Ab IBM Security Directory Integrator Version 7.2 wird diese Eigenschaft zur Aktivierung des NIST-Modus in IBM Security Directory Integrator verwendet. Wenn diese Eigenschaft auf true gesetzt ist, wird die Ausführung von IBM Security Directory Integrator im NIST-konformen Modus aktiviert. Der Standardwert ist false, d. h., die Ausführung erfolgt standardmäßig nicht im NIST-Modus.

Tabelle 3. Gelöschte und geänderte Eigenschaften in Version 7.1.1

Alte Eigenschaft (vor Version 7.0)	Neue Eigenschaft	Anmerkungen
## Active Correlation Technology engine settings # act.engine.rule.set.file=myrules.acts	*GELÖSCHT*	ACT-Steuerkomponente und ACT-Connector entfernt
# Location of directory where the JRE that SDI will use is installed com.ibm.di.jvmdir=\$jvmRoot\$	*GELÖSCHT*	Diese Eigenschaft kann nicht mehr angegeben werden.
com.ibm.di.scriptengine.precompile=true	*GELÖSCHT*	Diese Eigenschaft kann nicht mehr angegeben werden. Die aktuelle Scriptsteuerkomponente ist nicht mit dieser Funktionalität ausgestattet.
com.ibm.di.scriptengine.regex=java	*GELÖSCHT*	Diese Eigenschaft kann nicht mehr angegeben werden. Es wird immer die Java-Syntax befolgt.

Tabelle 3. Gelöschte und geänderte Eigenschaften in Version 7.1.1 (Forts.)

Alte Eigenschaft (vor Version 7.0)	Neue Eigenschaft	Anmerkungen
ibmjs.options=com.ibm.di.script.ScriptEngineOptions	*GELÖSCHT*	Diese Eigenschaft steht mit der vorangehenden Eigenschaft in Zusammenhang und stellt keine gültige Option mehr da.
com.ibm.di.store.create.checkpoint.store=<multiple statements>	*GELÖSCHT*	Die Funktionalität für den Neustart am Prüfpunkt wurde entfernt. Alle hiermit zusammenhängenden Anweisungen des Systemspeichers für die Tabellenerstellung sollten ebenfalls entfernt werden.
com.ibm.di.admin.library.dir=	*GELÖSCHT*	Diese Eigenschaft wird vom aktuellen Konfigurationseditor nicht verwendet und kann daher nicht mehr angegeben werden.
api.remote.on=false	api.remote.on=true	RMI ist im IBM Security Directory Integrator-Server standardmäßig aktiviert. Diese Eigenschaft ist auf "true" gesetzt, da sie standardmäßig aktiviert ist.
javax.net.ssl.trustStore={protect}-javax.net.ssl.trustStorePassword=javax.net.ssl.trustStoreType=	javax.net.ssl.trustStore=serverapi\testadmin.jks {protect}-javax.net.ssl.trustStorePassword=administrator javax.net.ssl.trustStoreType=jks	RMI ist im IBM Security Directory Integrator-Server standardmäßig aktiviert. Leere Werte wurden durch den Standard-Truststore ersetzt.
javax.net.ssl.keyStore={protect}-javax.net.ssl.keyStorePassword=javax.net.ssl.keyStoreType=	javax.net.ssl.keyStore=serverapi\testadmin.jks {protect}-javax.net.ssl.keyStorePassword=administrator javax.net.ssl.keyStoreType=jks	RMI ist im IBM Security Directory Integrator-Server standardmäßig aktiviert. Leere Werte wurden durch den Standardschlüsselspeicher ersetzt.
com.metamerge.securityTransformation=DES/ECB/NoPadding	com.ibm.di.securityTransformation=DES/ECB/NoPadding	Der Name der Eigenschaft für die Zertifizierung nach FIPS 140-2 wurde geändert.
com.ibm.di.server.keystore=myKeyStore.jks com.ibm.di.server.key.alias=myKeyAlias	api.keystore=myKeyStore.jks api.key.alias=myKeyAlias	Die Schlüsselspeichereigenschaften für die Server-API wurden umbenannt.
com.ibm.di.store.database=TDISysStore com.ibm.di.store.jdbc.driver=org.apache.derby.jdbc.EmbeddedDriver com.ibm.di.store.jdbc.urlprefix=jdbc:derby: com.ibm.di.store.jdbc.user=APP	#com.ibm.di.store.database=TDISysStore #com.ibm.di.store.jdbc.driver=org.apache.derby.jdbc.EmbeddedDriver #com.ibm.di.store.jdbc.urlprefix=jdbc:derby: #com.ibm.di.store.jdbc.user=APP	Die Eigenschaften im Abschnitt EMBEDDED MODE für den Systemspeicher wurden auf Kommentar gesetzt, da der Systemspeicher jetzt standardmäßig im Netzmodus ausgeführt wird. Das Installationsprogramm nimmt diese Änderung in keinem Fall vor; falls Sie zuvor Cloudscape/Derby im integrierten Modus verwendet haben, müssen Sie diese Änderung manuell nachvollziehen.
#com.ibm.di.store.database=jdbc:derby://localhost:1527/TDISysStore;create=true #com.ibm.di.store.jdbc.driver=org.apache.derby.jdbc.ClientDriver #com.ibm.di.store.jdbc.urlprefix=jdbc:derby: #com.ibm.di.store.jdbc.user=APP #com.ibm.di.store.jdbc.password=APP #com.ibm.di.store.jdbc.start.mode=automatic #com.ibm.di.store.jdbc.host=localhost #com.ibm.di.store.jdbc.port=1527 #com.ibm.di.store.jdbc.sysibm=true	com.ibm.di.store.database=jdbc:derby://localhost:1527/TDISysStore;create=true com.ibm.di.store.jdbc.driver=org.apache.derby.jdbc.ClientDriver com.ibm.di.store.jdbc.urlprefix=jdbc:derby://localhost:1527/ com.ibm.di.store.jdbc.user=APP com.ibm.di.store.jdbc.password=APP com.ibm.di.store.jdbc.start.mode=automatic com.ibm.di.store.jdbc.host=localhost com.ibm.di.store.jdbc.port=1527 com.ibm.di.store.jdbc.sysibm=true	Dies sind die neuen Standardeigenschaften für den Systemspeicher in IBM Security Directory Integrator Version 7.1.1. Falls Sie Ihre Installation migriert haben, müssen Sie diese Änderungen ebenfalls in Ihrer Datei global.properties vornehmen, wenn der Systemspeicher im Netzmodus ausgeführt werden soll. Die neue Architektur des Konfigurationseditors führt in Kombination mit anderen Änderungen am Entwicklungsprozess dazu, dass eine Ausführung des Systemspeichers im integrierten Modus sehr schwerfällig ist. Es wird daher dringend empfohlen, die Ausführung im Netzmodus vorzunehmen.

Tabelle 3. Gelöschte und geänderte Eigenschaften in Version 7.1.1 (Forts.)

Alte Eigenschaft (vor Version 7.0)	Neue Eigenschaft	Anmerkungen
api.config.folder=\$change\$/configs	api.config.folder=configs	Der Ordner configs ist jetzt immer ein lokaler Ordner für das Lösungsverzeichnis.
<pre>##----- ## System Queue settings ##----- ## If set to "true" the System Queue ## is initialized on startup and can ## be used; ## otherwise the System Queue is not ## initialized and cannot be used. systemqueue.on=false ### MQe JMS driver initialization Eigenschaften## Specifies the location of the MQe initialization file. ## This file is used to initialize MQe on TDI server startup. systemqueue.jmsdriver.param.mqe .file.ini=\$change\$/MQePWStore /pwstore_server.ini</pre>	<pre>##----- ## System Queue settings ##----- ## If set to "true" the System Queue ## is initialized on startup and ## can be used; ## otherwise the System Queue is not ## initialized and cannot be used. ## systemqueue.on=true ### MQe JMS driver initialization Eigenschaften## Specifies the location of the MQe initialization file. ## This file is used to initialize MQe on TDI server startup. systemqueue.jmsdriver.param.mqe .file.ini=MQePWStore/pwstore_server .ini</pre>	Die Systemwarteschlange ist in IBM Security Directory Integrator jetzt standardmäßig aktiviert. Auch die IBM WebSphere MQ Everyplace-Initialisierungsdatei befindet sich nun in einem Unterverzeichnis des Lösungsverzeichnisses.

Tabelle 4. Neue Eigenschaften in Version 7.0

Eigenschaft	Anmerkungen
com.ibm.di.server.fipsmode.on=false	Es wurde eine neue Eigenschaft für die Aktivierung/Inaktivierung des FIPS-Modus hinzugefügt.
<pre>## To enable the built-in JAAS Authentication mechanism, ## set this property to "[jaas]". api.custom.authentication ## JAAS Authentication properties ## ----- ## java.security.auth.login.config=</pre>	Diese Eigenschaft stellt die Unterstützung für JAAS als Provider für die Server-API-Authentifizierung. Es wird eine leere Eigenschaft zur Verfügung gestellt, in der Sie die JAAS-Konfigurationsdatei angeben können.
<pre>## Encryption certificate properties com.ibm.di.server.encryption.keystore = <<value of com.ibm.di.server.keystore from 6.1.1 global.properties>> com.ibm.di.server.encryption.key.alias = <<value of com.ibm.di.server.key.alias from 6.1.1 global.properties >> ## Server API keystore passwords {protect}-api.keystore.password= << keystore password from idisrv.sth>> {protect}-api.key.password= << key password from idisrv.sth if present>></pre>	Diese Eigenschaften sind separate Konfigurationsoptionen für das Zertifikat, das für die PKI-Verschlüsselung und für SSL verwendet wird.
<pre>## TDI Logging com.ibm.di.logging.enabled=true</pre>	Diese Eigenschaft stellt Mechanismen bereit, mit denen die Protokollierung vollständig inaktiviert werden kann. Setzen Sie sie auf den Wert "false", falls Sie die gesamte Protokollierung inaktivieren wollen.
<pre>derby.connection.requireAuthentication=true derby.authentication.provider=BUILTIN derby.database.defaultConnectionMode=fullAccess</pre>	Diese Eigenschaften sind zusätzliche Parameter für den Systemspeicher (in Derby) im Netzmodus.
<pre>##PKCS11 options ##Set the value of following properties to use PKCS11 enabled ## devices to store TDI servers private key / certificate. com.ibm.di.pkcs11cfg=etc\pkcs11.cfg com.ibm.di.server.pkcs11=false com.ibm.di.server.pkcs11.library= com.ibm.di.server.pkcs11.slot= {protect}-com.ibm.di.server.pkcs11.password =PASSWORD</pre>	Diese Eigenschaften unterstützen bei mit PKCS 11 kompatiblen Verschlüsselungseinheiten den privaten Schlüssel und das Zertifikat des IBM Security Directory Integrator-Servers.

Tabelle 4. Neue Eigenschaften in Version 7.0 (Forts.)

Eigenschaft	Anmerkungen
<pre>## Specify the unique ID for the TDI Server ## ----- ## This property helps a client connecting to the TDI server ## to identify different servers ## running on the same IP and the same port in different time. ## (Default is blank) com.ibm.di.server.id=</pre>	<p>Der IBM Security Directory Integrator-Server muss eine verfügbare eindeutige Server-ID für Clients des fernen Servers bereitstellen, damit diese den Server erkennen können, mit dem die Kommunikation stattfindet.</p>
<pre>## Timeout in minutes for loading configuration. api.config.load.timeout=2</pre>	<p>Die Initialisierung der Konfiguration und die Initialisierung der Server-API müssen synchronisiert werden.</p>
<pre>com.ibm.di.server.encryption.keystoretype = jks com.ibm.di.server.encryption.transformation = RSA</pre>	<p>Diese Eigenschaften sind mit der Unterstützung des symmetrischen Chiffrierwerts (konform mit FIPS 140-2) verbunden.</p>
<pre>## Specifies a list of Server notification types, which will be ## suppressed. ## Notifications of suppressed types will not be propagated ## by the notifications framework. ## The notification types in the list are separated by spaces. ## Wildcards may be included. ## Example: ## api.notification.suppress=di.al.* di.ci.start ## The above example will suppress all AssemblyLine related ## notifications as well as ## notifications for starting a configuration instance. ## If the property is missing or is empty, no notifications ## will be suppressed. api.notification.suppress=di.server.api.authenticate di .server.api.authorize.*</pre>	<p>Diese Eigenschaften stellen die Unterdrückung der Serverbenachrichtigung für die IBM Security Directory Integrator-Prüffunktionen bereit.</p>
<pre>api.audit.on=false</pre>	<p>Diese Eigenschaft stellt die IBM Security Directory Integrator-Prüffunktionen bereit.</p>

Tabelle 4. Neue Eigenschaften in Version 7.0 (Forts.)

Eigenschaft	Anmerkungen
<pre> ## This property specifies whether LDAP Group authentication ## is turned on. ## If it is set to 'true', the group membership of ## the authenticating user will be resolved and will ## be taken into account during authorization. ## If it is missing, the default value 'false' is used. api.custom.authentication.ldap.groupsupport= false ## Specifies the name of the attribute of a user in LDAP that ## contains a list of the groups of which the user is a member. ## It is taken into account only if 'api.custom.authentication. ## ldap.groupsupport' is set to true. api.custom.authentication.ldap.usermembershipattribute= ## Specifies how groups are named in the membership attribute ## of a user. ## For example, if the user's membership attribute contains values, ## which correspond to the 'objectSID' attributes of groups, ## set this property to 'objectSID'. ## If the user's membership attribute contains distinguished names ## of groups, then set this property to 'dn'. ## The property is required in case 'api.custom.authentication.ldap. ## groupsupport' is set to true. api.custom.authentication.ldap.usermembershipattributecontent= ## Specifies the name of a group's attribute in LDAP which corresponds to the way the group is named in the TDI User Registry. ## For example, if LDAP groups are addressed in the TDI registry by ## their common name, then set this property to 'cn'. ## If the User Registry contains the distinguished names of the ## groups, then set this property to 'dn'. api.custom.authentication.ldap.groupnameattribute= ## Represents the LDAP directory context, where groups will be searched. ## It is required only when LDAP group support ## is enabled api.custom.authentication.ldap.groupsearchbase= ## Optional property, which represents a list of space-separated ## attribute names. ## Specifies attributes which have non-string syntax. ## api.custom.authentication.ldap.binaryattributes= </pre>	<p>Diese Eigenschaften erweitern die Autorisierung durch die Unterstützung von LDAP-Gruppen.</p>

Tabelle 5. Neue Eigenschaften in Version 7.1

Eigenschaft	Anmerkungen
<pre> # api.remote.server.ports=8700-8900 </pre>	<p>Diese Eigenschaft ist standardmäßig auf Kommentar gesetzt. Sie wird verwendet, um RMI-Ports zu konfigurieren. Dies ist hilfreich, falls die Standardports einen Konflikt mit der Firewall verursachen.</p> <p>Der Server verwendet diese Ports, um zusätzlich zu der Empfangsbereitschaft an den durch andere Eigenschaften definierten Ports für eingehende RMI-Serviceanforderungen empfangsbereit zu sein. Für ausgehende RMI-Serviceanforderungen können wahlfreie Portnummern verwendet werden.</p>

Tabelle 5. Neue Eigenschaften in Version 7.1 (Forts.)

Eigenschaft	Anmerkungen
<pre>## The properties determine the default bind address and the remote ## bind address for the Server API. ## * means bind to all network interfaces. The Remote Bind Address ## overrides the Default one. ## Only one IP address should be set. No hostnames are accepted. ## Mind that the java.rmi.server.hostname property is set implicitly ## to equal the Remote Bind Address property when used. This will cause the client stubs to create sockets on the specified Remote Bind Address. # com.ibm.di.default.bind.address=* # api.remote.bind.address=*</pre>	<p>Diese beiden Eigenschaften sind standardmäßig auf Kommentar gesetzt. Sie werden verwendet, um die Netzchnittstelle (Hostname oder IP-Adresse) zu konfigurieren, über die die ferne API empfangsbereit ist.</p>
<pre>## Touchpoint Server properties tp.server.on=false tp.server.port=1098 tp.server.config=etc/tp.xml tp.server.auth=false tp.server.auth.realm=Tivoli Directory Integrator Touchpoint-Server</pre>	<p>Diese Eigenschaften konfigurieren die REST-Schnittstelle für IBM Security Directory Integrator-Connectors unter Verwendung eines Service, der auf SCMP (Service Control Management Protocol) basiert.</p>
<pre>## ## Server API client properties ## api.client.ssl.custom.properties.on=true api.client.keystore=serverapi/testadmin.jks {protect}-api.client.keystore.pass=administrator api.client.keystore.type=jks {protect}-api.client.key.pass=administrator api.client.truststore=serverapi/testadmin.jks {protect}-api.client.truststore.pass=administrator api.client.truststore.type=jks</pre>	<p>Diese Eigenschaften aktivieren angepasste SSL-Eigenschaften für Clients der Server-API. Bei der Einstellung "api.client.ssl.custom.properties.on=true" werden die Eigenschaften "api.client.*" von den Clients der Server-API verwendet. Andernfalls werden die Standardeigenschaften "javax.net.ssl.*" verwendet.</p>

Tabelle 6. Neue Eigenschaften in Version 7.1.1

Eigenschaft	Anmerkungen
<pre>## Web container web.server.port=1098 web.server.ssl.on=false web.server.ssl.client.auth.on=false # web.server.session.timeout=300</pre>	<p>Bei diesen Eigenschaften handelt es sich um allgemeine Einstellungen für die REST-API und das Dashboard. Dadurch werden die Port- und Sicherheitseinstellungen des HTTP-Zugriffspunkts in IBM Security Directory Integrator-REST und das Dashboard angegeben.</p>
<pre>## Dashboard properties ## dashboard.on=true dashboard.templates.folder=dashboard/templates ## Dashboard authentication properties ## ## The values for localhost and remotehost can be: ## none: No authentication is required ## deny: All connections denied ## ldap: Authentication is done by logging into an LDAP ## server and optionally validating group membership ## ## dashboard.ldap.url ## Specify the LDAP host port and optionally a search ## base (ldap://<host>:<port>[/<search base>]) ## ## dashboard.ldap.url.group ## Specify the LDAP host port and optionally a search ## base (ldap://<host>:<port>[/<search base>]) ## dashboard.auth=true dashboard.auth.localhost=none ## dashboard.auth.remote=deny # dashboard.auth.ldap.url=ldap://localhost:389/ou=users, ## ou=system # dashboard.auth.ldap.url.group=ldap://localhost: ## 389/cn=group1,ou=groups,ou=system</pre>	<p>Diese Eigenschaften sind dashboardspezifisch. Die Eigenschaften, die manuell festgelegt werden, sind „dashboard.on“ (aktiviert/inaktiviert die Dashboard-Webanwendung) und „dashboard.templates.folder“ (Position von Schablonenlösungen).</p> <p>Alle anderen Eigenschaften können im Dashboard bearbeitet werden.</p>

Tabelle 6. Neue Eigenschaften in Version 7.1.1 (Forts.)

Eigenschaft	Anmerkungen
<pre>## REST API ## ----- api.rest.on=true api.rest.auth=false api.rest.auth.realm=Tivoli Directory Integrator REST API api.rest.jmsdriver.name =com.ibm.di.systemqueue.driver.ActiveMQ api.rest.jmsdriver.queue.sender.persistence=false api.rest.jmsdriver.queue.sender.timeToLive=60000 api.rest.jmsdriver.param.jms oker=vm://localhost?brokerConfig=xbean:etc/activemq.xml # api.rest.jmsdriver.auth.username # api.rest.jmsdriver.auth.password</pre>	<p>Mit diesen Eigenschaften werden die IBM Security Directory Integrator-REST-API-Unterstützungs- und Sicherheitseinstellungen konfiguriert. Die Eigenschaft "api.rest.jmsdriver" gibt die in asynchronen Protokollnachrichten zu verwendende JMS-Warteschlange an. Nachrichten werden vom Dashboard verwendet.</p>

Tabelle 7. In Version 7.1.1 gelöscht/geändert

Alte Eigenschaft	Neue Eigenschaft	Anmerkungen
<pre>com.ibm.di.store.database =jdbc:derby://localhost :1527/\$change\$/TDISysStore; create=true</pre>	<pre>com.ibm.di.store.database =jdbc:derby://localhost :1527/\$soldir\$/TDISysStore; create=true</pre>	<p>Ab IBM Security Directory Integrator wird \$soldir\$ standardmäßig nicht durch <tdi-installationsverzeichnis> ersetzt. Das Verzeichnis wird bei der Laufzeit innerhalb der JVM mit dem aktuellen Lösungsverzeichnis des Benutzers aktualisiert. Daher ist der IBM Security Directory Integrator-Systemspeicher für jedes Lösungsverzeichnis eindeutig.</p>
<pre>tp.server.port=1098</pre>	*GELÖSCHT*	<p>Diese Eigenschaft wurde als web.server.port=1098 neu definiert.</p>

- amc.properties:

In der folgenden Tabelle sind die Eigenschaften angegeben, die in IBM Security Directory Integrator Version 7.1.1 gelöscht oder geändert wurden:

Tabelle 8. Gelöschte und geänderte Eigenschaften in AMC

Alte Eigenschaft (vor Version 7.0)	Neue Eigenschaft	Anmerkungen
AMC.auth	*GELÖSCHT*	
monitor.refresh.rate	*GELÖSCHT*	<p>Diese Eigenschaft gab die Aktualisierungsrate für die Anzeige "Status überwachen" in Minuten an.</p>
monitor.startup	*GELÖSCHT*	<p>Mit dieser Eigenschaft wurde festgelegt, dass die Anzeige "Status überwachen" als erste Anzeige ausgegeben wird, nachdem sich der Benutzer angemeldet hat.</p>

Table 8. Gelöschte und geänderte Eigenschaften in AMC (Forts.)

Alte Eigenschaft (vor Version 7.0)	Neue Eigenschaft	Anmerkungen
LDAPHostName LDAPPort LDAPAdminUID LDAPAdminPwd LDAPServerType LDAPBindID LDAPBindPassword LDAPSuffix LdapUserPrefix LDAPUserSuffix LdapGroupPrefix LDAPGroupSuffix LDAPUserObjectClass LDAPGroupObjectClass LDAPGroupMember LDAPUserFilter LDAPGroupFilter LDAPsearchTimeout LDAPsslEnabled LDAPIgnoreCase	*GELÖSCHT*	Diese Eigenschaften gaben LDAP-Details an.
com.ibm.di.amc.jdbc.start.mode	Neuer Standardwert: Automatic	
com.ibm.di.amc.jdbc.host	Neuer Standardwert: localhost	
com.ibm.di.amc.jdbc.port	Neuer Standardwert: 1528	
com.ibm.di.amc.jdbc.sysibm	Neuer Standardwert: True	

In der folgenden Tabelle sind die Eigenschaften angegeben, die in IBM Security Directory Integrator Version 7.0 hinzugefügt wurden:

Table 9. Neue Eigenschaften in AMC

Neue Eigenschaft (Standardwert)	Anmerkungen
am.logrotate (10)	Hiermit wird das maximale Alter von AM-Protokolldateien in Tagen bestimmt. Die Protokolldateien, die älter als der angegebene Wert sind, werden gelöscht. Der Minimalwert ist 1; er kann bis zu einem Wert von 2147483647 erhöht werden.
amc.session.timeout (20)	Hiermit wird die maximale Zeit (in Minuten) bestimmt, die ein Benutzer inaktiv ist, bevor die AMC-Sitzung abläuft und der Benutzer automatisch von AMC abgemeldet wird. Der Wert muss eine positive ganze Zahl sein.
al.workEntries.cacheSize (100)	Diese Eigenschaft wird von AMC verwendet, wenn die Fertigungslinie im synchronen Modus gestartet wird. Anhand der hier angegebenen Cachegröße wird die Größe des Cache für Work-Einträge ermittelt.
amc.db.type (derby)	Diese Eigenschaft gibt die durch AMC verwendete Datenbank an.
am.api.host (localhost)	Diese Eigenschaft gibt RMI-Details für Action Manager an.
am.api.port (13104)	Diese Eigenschaft gibt RMI-Details für Action Manager an.
com.ibm.di.server.port.default (1099)	Diese Eigenschaft gibt den Standardport für den IBM Security Directory Integrator-Server an. Diese Eigenschaft kann durch das Installationsprogramm von IBM Security Directory Integrator in einen anderen Wert als 1099 modifiziert werden. Wenn AMC erstmalig gestartet wird (oder die AMC-Datenbank nicht mehr verfügbar war), wird diese Eigenschaft gelesen und ihr Wert in der neu erstellten AMC-Datenbank gespeichert. Später wird sie verwendet, wenn AMC eine Verbindung zur Standardinstanz des IBM Security Directory Integrator-Servers herstellt.

- am_config.properties:

In der folgenden Tabelle sind die Eigenschaften angegeben, die in IBM Security Directory Integrator Version 7.1.1 gelöscht oder geändert wurden:

Tabelle 10. Gelöschte und geänderte Eigenschaften in Action Manager

Alte Eigenschaft (vor Version 7.0)	Neue Eigenschaft	Anmerkungen
com.ibm.di.amc.am.serverapi .fail.interval.time=120 com.ibm.di.amc.am.queryProperty .interval.time=600 com.ibm.di.amc.am.healthAL.interval .time=5	*GELÖSCHT*	Diese Eigenschaften sollten auf Kommentar gesetzt werden, da ihre Werte durch den Benutzer über AMC konfiguriert werden, wenn der Auslöser "Fehler für Server-API" oder der Auslöser "Bei einer Eigenschaft" erstellt bzw. eine Fertigungslinie für den ordnungsgemäßen Betrieb konfiguriert wird. Die in der Datei am-config.properties angegebenen Eigenschaften werden somit nicht verwendet.
com.ibm.di.amc.am.queryAL.interval .time	*GELÖSCHT*	
javax.net.ssl.trustStore=\$change\$/bin/amc/ActionManager/testadmin.jks javax.net.ssl.keyStore=\$change\$/bin/amc/ActionManager/testadmin.jks	javax.net.ssl.trustStore=bin/amc/ActionManager/testadmin.jks javax.net.ssl.keyStore=bin/amc/ActionManager/testadmin.jks	Die Truststore-Dateien sind nun lokale Dateien für das Lösungsverzeichnis.

In der folgenden Tabelle sind die Eigenschaften angegeben, die in IBM Security Directory Integrator Version 7.1.1 hinzugefügt wurden:

Tabelle 11. Neue Eigenschaften in Action Manager

Neue Eigenschaft (Standardwert)	Anmerkungen
smtp.host= smtp.port= smtp.user= {protect}-smtp.password=	Diese Eigenschaften geben Details für den SMTP-Server an und wurden in IBM Security Directory Integrator 7.0 hinzugefügt.
javax.net.ssl.trustStore= tdi-installationsverzeichnis/serverapi/testadmin.jks{protect} -javax.net.ssl.trustStorePassword=administrator javax.net.ssl.trustStoreType=jks javax.net.ssl.keyStore=tdi-installationsverzeichnis/serverapi/ testadmin.jks {protect}-javax.net.ssl.keyStorePassword=administrator javax.net.ssl.keyStoreType=jks	Dies sind SSL-Eigenschaften für Action Manager. Sie wurden in IBM Security Directory Integrator 7.1 hinzugefügt.
com.ibm.di.amc.am.encryption.keystore = tdi-installationsverzeichnis/ testserver.jks com.ibm.di.amc.am.encryption.key.alias = server com.ibm.di.amc.am.encryption.keystoretype = jks com.ibm.di.amc.am.encryption.transformation = RSA com.ibm.di.amc.am.stash.file = tdi-installationsverzeichnis/idisrv.sth	Dies sind Verschlüsselungseigenschaften für Action Manager. Sie wurden in IBM Security Directory Integrator 7.1 hinzugefügt. Diese Eigenschaften haben Ähnlichkeit mit den Verschlüsselungseigenschaften, die durch den Server verwendet werden. Zur Vereinfachung wurde die Position der Stashdatei als Eigenschaft com.ibm.di.amc.am.stash.file hinzugefügt. Action Manager verwendet standardmäßig die Verschlüsselung/Entschlüsselung aus dem Schlüsselspeicher und der Stashdatei des Servers für geschützte Action Manager-Eigenschaften.

Konfigurationen

Bestimmte Directory Integrator-Komponenten und -Funktionen wurden modifiziert oder entfernt. Konfigurationen, die sich auf diese Komponenten/Funktionen beziehen, müssen manuell migriert werden. Die betreffenden Komponenten/Funktionen sind in der folgenden Liste aufgeführt:

- Funktion für Neustart am Prüfpunkt

Diese Funktionalität wurde in Version 7.0 entfernt. Hierdurch verfügen Connectors, die den Iteratormodus unterstützen, nur noch über die Standardfunktionsweise, die darin besteht, dass sie eine einfache Verbindungswiederherstellung durchführen und automatisch so oft vorwärts springen können, wie es der Anzahl der erfolgreichen Lesevorgänge entspricht. Dieser Funktionsweise liegt die Annahme zugrunde, dass Sie die zuletzt verlassene Position wieder erreichen, wenn so oft vorwärts gesprungen wird, wie es der Anzahl der Einträge entspricht. Die meisten IBM Security Directory Integrator-Connectors versuchen nicht automatisch, dies durchzuführen, da das Verhalten unbestimmt oder ungeeignet sein kann. Das Standardverhalten ist jedoch für jeden Connector spezifisch. Die Fähigkeit, so oft automatisch vorwärts zu springen, wie erfolgreiche Lesevorgänge ausgeführt wurden, ist eine neue Verbindungswiederherstellungsoption, die für jeden Connector verfügbar ist und in der Anzeige "Verbindungsfehler" konfiguriert wird. Siehe hierzu die Abschnitte zu **Konfigurationseditor** -> **Connectoreditor** -> **Verbindungsfehler** im Abschnitt *Konfiguration* im IBM Knowledge Center for IBM Security Directory Integrator. Falls Sie eine umfangreichere Funktionsweise als das automatische Überspringen von verarbeiteten Einträgen benötigen, müssen Sie in Ihren Lösungen eine der folgenden Optionen verwenden:

- Delta für einen Iteratormodus für das dynamische Ändern von Ergebnismengen konfigurieren
- Hook on_connection_failure überschreiben und angepasste Logik für die Verbindungswiederherstellung ausführen
- Verwendung von Derby/Cloudscape im integrierten Modus als Systemspeicher durch verschiedene JVMs

Das standardmäßige und empfohlene Verhalten in IBM Security Directory Integrator ist die Ausführung von Derby im Netzmodus. Falls Sie Derby weiterhin im integrierten Modus verwenden, gelten weiterhin die Überlegungen in Bezug auf mehrere JVMs, die versuchen, gleichzeitig dieselbe Datenbank zu nutzen. Entsprechende Informationen enthält der Abschnitt „Derby als Systemspeicher verwenden“ auf Seite 226. Angaben über die Migration von Datenbanken können Sie unter „Cloudscape-Datenbank auf Derby migrieren“ auf Seite 97 nachlesen.

- Exchange-Änderungsprotokollconnector
Dieser Connector wurde in Version 7.0 entfernt. Die Verwendung des nicht unterstützten Exchange-Änderungsprotokollconnectors, der nun als Beispiel im Verzeichnis *tdi-installationsverzeichnis/examples/ExchangeChangelogConnector* bereitgestellt wird, ist möglich.
- B-Baum-Connector
Dieser Connector wurde in Version 7.0 aus der Standardinstallation entfernt. Verwenden Sie stattdessen den Systemspeicherconnector (siehe „B-Baum-Tabellen und B-Baum-Connector auf Systemspeicher migrieren“ auf Seite 97) oder alternativ den (nicht unterstützten) B-Baum-Connector, der nun als Beispiel im Verzeichnis *tdi-installationsverzeichnis/examples/BTreeDBCConnector* bereitgestellt wird.
- Domino-Änderungserkennungsconnector:
Die folgenden Angaben gelten nur für Version 6.0 und 6.1.
Der Parameter für den **Übermittlungsmodus** wurde entfernt. Stattdessen wird der Parameter für die **Statusschlüsselpermanenz** verwendet. Alte Konfigurationen, die diesen Parameter verwenden, verhalten sich folgendermaßen:
 - Falls der Parameter für den **Übermittlungsmodus** auf die Einstellung gesetzt ist, dass die Zusicherung nur einmalig und nur ein einziges Mal pro Übermittlung stattfindet, wird der Parameter für die **Statusschlüsselpermanenz**

auf die Einstellung "Nach Lesevorgang" gesetzt, was ein identisches Verhalten bewirkt: Der Synchronisationsstatus wird unmittelbar nach dem Lesen des Notes-Dokuments gespeichert.

- Falls der Parameter für den **Übermittlungsmodus** auf die Einstellung für eine normale zugesicherte Übermittlung gesetzt ist, wird geprüft, ob ein gültiger Parameter für die **Statusschlüsselpermanenz** vorhanden ist. Bleibt die Suche erfolglos, wird der Wert des Parameters für die **Statusschlüsselpermanenz** auf die Einstellung "Nach Lesevorgang" gesetzt. Falls der Parameter in der Konfiguration gefunden wird, wird sein ursprünglicher Wert verwendet.

- **IDS-Änderungsprotokollconnector**

Die Option "CRAM-MD5" ist in Version 7.0 nicht mehr verfügbar. Sie müssen manuell einen anderen Authentifizierungsmechanismus auswählen.

In Version 6.2 von IBM Security Directory Server wurden die Klassen "BEREncoder" und "BERDecoder" aus dem Paket `com.ibm.asn1` in das Paket `com.ibm.ldap.bp.asn1` versetzt. Ab IBM Security Directory Server Version 7.0 müssen angepasste Benutzerlösungen, die die alten Klassen (`com.ibm.asn1.BEREncoder` und `com.ibm.asn1.BERDecoder`) direkt verwenden, aktualisiert werden, damit diese Änderung berücksichtigt wird.

- Funktionskomponenten "EMFXMLToSDO" und "EMFSDToXML"

Diese Funktionskomponenten werden in Version 7.0 nicht mehr unterstützt. Verwenden Sie künftig eine andere Funktionalität.

- **DSMLv2-Parser**

Die folgenden Angaben gelten nur für Version 6.0.

Die Attribute "dsml.request" und "dsml.response" wurden entfernt. Diese Attribute stellten unaufbereitete Anforderungs- und Antwortobjekte aus der ITIM-Bibliothek "DSMLv2" bereit. Falls alte Konfigurationen vorhanden sind, die eines dieser Attribute verwenden, müssen Sie Ihre alten Konfigurationen so bearbeiten, dass diese Attribute nicht mehr verwendet werden. Alle Daten, die durch die unaufbereiteten Anforderungs- und Antwortobjekte verfügbar sind, sind auch über die anderen, vom DSMLv2-Parser bereitgestellten Attribute verfügbar.

- **ITIM-Agentenconnector**

Falls Sie in einer früheren Version von IBM Security Directory Server den ITIM-Agentenconnector verwendet haben, müssen Sie möglicherweise die Methode für die Konfiguration von SSL-Verbindungen ändern. Der ITIM-Agentenconnector in IBM Security Directory Server verwendet JSSE (Java-basierter Schlüssel-speicher oder Truststore) für die SSL-Authentifizierung. Dies macht es erforderlich, dass Sie die SSL-bezogenen Zertifikatsdetails in der Datei `global.properties` oder `solution.properties` konfigurieren, statt den Zertifikatsnamen im Parameter "Zertifikatsdatei einer Zertifizierungsstelle" des alten ITIM-Agentenconnectors anzugeben. Hierzu sind die folgenden Schritte erforderlich:

1. Importieren Sie das Zertifikat des ITIM-Agenten, das zuvor im Parameter "Zertifikatsdatei einer Zertifizierungsstelle" angegeben war, in den IBM Security Directory Integrator-Truststore, beispielsweise mit dem Tool *keytool* (die Verwendung von "Ikeyman" ist ebenfalls möglich):

```
keytool -import -file servercertificate.der -keystore tim.jks
```

In diesem Beispiel wird der Truststore in der Datei `tim.jks` gespeichert.

2. Konfigurieren Sie diesen Truststore im Abschnitt "server authentication" der Datei `global.properties` oder `solution.properties`:

```
## server authentication
javax.net.ssl.trustStore=serverapi\tim.jks
{protect}-javax.net.ssl.trustStorePassword=administrator
javax.net.ssl.trustStoreType=jks
```


Der ITIM-Agentenconnector verwendet nun dieselbe JSSE-basierte Architektur für die sichere Kommunikation wie die übrigen Komponenten von IBM Security Directory Integrator.

Falls Sie bereits eine Truststore-Datei in der Datei `global.properties` oder `solution.properties` konfiguriert haben, importieren Sie das Zertifikat in diesen Truststore, statt einen neuen zu erstellen.

- XML-Parser

Der XML-Parser aus den Versionen vor Version 7.0 wurde umbenannt und wird nun als "einfacher XML-Parser" bezeichnet. Der aktuelle XML-Parser ist ein neuer Parser mit einer größeren Funktionalität, insbesondere in Bezug auf hierarchische Objekte. Konfigurationsdateien, die mit älteren Versionen von IBM Security Directory Server erstellt wurden und den XML-Parser referenzieren, referenzieren nach einem Import in Version 7.1 und spätere Versionen den einfachen XML-Parser (da der Klassenname nicht geändert wurde). Falls Sie stattdessen den neuen XML-Parser verwenden wollen, müssen Sie dies in den Fertigungslinien und/oder Connectors ändern. Damit der neue XML-Parser dasselbe Verhalten wie der alte Parser aufweist, müssen Sie die Parameter "Eintragstag" und "Werttag" auf die Werte setzen, die beim einfachen XML-Parser verwendet wurden.

Anmerkung: Der einfache XML-Parser exportiert eine Scriptvariable namens "xmldom", die durch den neuen XML-Parser nicht exportiert wird. Der neue XML-Parser stellt die tiefere Hierarchie durch den Eintrag selbst dar. Logik, die sich auf die Variable "xmldom" stützt und nicht dahingehend überarbeitet werden kann, dass die durch die Eintragsklasse bereitgestellte hierarchische Struktur verwendet wird, darf nicht auf den neuen XML-Parser migriert werden.

- Funktionskomponenten für Castor-Java in XML und XML in Java

Ab IBM Security Directory Server Version 7.1 wurde die Position der Castor-Zuordnungsdatei von `tdi-installationsverzeichnis/jars/functions/di_castor_mapping.xml` in `tdi-installationsverzeichnis/etc/di_castor_mapping.xml` geändert.

Infolgedessen gibt der Standardwert für die **Castor-Zuordnungsdatei** nun die neue Position wieder.

- HTTP-Client-Connector

Ab IBM Security Directory Server Version 7.1 wurde der HTTP-Client-Connector so modifiziert, dass automatisch ein HTTP-Header "Connection" mit dem Wert "close" gesendet wird, wenn die TCP-Verbindung nicht für mehrere HTTP-Anforderungen wiederverwendet werden soll. Grund für diese Modifizierung ist die Einhaltung der Empfehlung gemäß HTTP 1.1 (<http://tools.ietf.org/html/rfc2616#section-14.10>).

Dieses Verhalten ist nach der Spezifikation von HTTP 1.1 verbindlich. Zuvor mussten Sie es selbst in der Fertigungslinie codieren.

- HTTP-Server-Connector

Ab IBM Security Directory Server Version 7.1 wurde der HTTP-Server-Connector so modifiziert, dass standardmäßig persistente HTTP-Verbindungen verwendet werden. Dies bedeutet, dass eine einzige TCP-Verbindung von demselben HTTP-Client für mehrere HTTP-Anforderungen verwendet werden kann (<http://tools.ietf.org/html/rfc2616#section-8.1>). HTTP-Clients können weiterhin einen Header "Connection" mit dem Wert "Keep-Alive" senden; dies ist allerdings keine Voraussetzung mehr für die Verwendung einer persistenten Verbindung. Inaktive TCP-Verbindungen werden nach einer Inaktivitätsdauer von 20 Sekunden automatisch geschlossen.

Angepasste Scripts

Falls Sie eines der Directory Integrator-Scripts angepasst haben (beispielsweise durch das Hinzufügen von Einträgen zu den Umgebungsvariablen PATH oder LD_LIBRARY_PATH in den Startscripts `ibmdisrv` und `ibmditk`), sollten Sie diese Anpassungen auf die entsprechenden Scripts der neuen Version übertragen.

Vorversionen von IBM Security Directory Server verwendeten in diesen Scripts die Variable (MY)CLASSPATH. In der aktuellen Version sind die erforderlichen Pfadinformationen integriert, weshalb diese Variable nicht mehr benötigt wird. Falls Sie die vorgenannten Scripts zuvor angepasst haben, um eigene Bibliotheken aufzunehmen, müssen Sie die Variable CLASSPATH nicht bearbeiten. Stellen Sie in diesem Fall lediglich sicher, dass sich Ihre Bibliothek an der richtigen Position befindet (normalerweise im Verzeichnis `jars/`), damit sie für IBM Security Directory Server auffindbar ist. Alternativ können Sie die Eigenschaft `com.ibm.di.loader.userjars` in der Datei `global.properties` verwenden und dort auf Ihr eigenes Verzeichnis verweisen, das in den Pfad des Ladeprogramms aufgenommen werden soll. In IBM Security Directory Server kann die Eigenschaft mehrere Verzeichnisse oder JAR-Dateien durch die Java-Eigenschaft "path.separator" getrennt angeben. Diese Eigenschaft hat bei Linux den Wert ":" und bei Windows den Wert ";". Das TDI-Ladeprogramm für JAR-Datei durchsucht Verzeichnisse rekursiv nach Dateien, die Klassen und Ressourcen enthalten. Nur Dateien mit der Erweiterung ".zip" oder ".jar" werden durchsucht.

Hinzugefügte oder ersetzte JAR-Dateien in der Installation

Falls Sie JAR-Dateien zur Installation hinzugefügt haben, sollten Sie diese auch in die neue Version kopieren.

IBM Security Directory Server erfordert und enthält nun eine mit Java 7 kompatible JVM (J2SE-Version 7.0.4). Falls Sie eigenen Code in Java entwickelt, diesen Code mit den JVM-Bibliotheken verknüpft und dies in Ihre IBM Security Directory Server-Lösung integriert hatten, müssen Sie Ihren Code unter Umständen erneut kompilieren und verknüpfen.

Falls Sie eine der ursprünglichen JAR-Dateien der Installation überschrieben hatten (z. B. durch die Aufnahme der erforderlichen MQ-JAR-Dateien im Verzeichnis *tdi-installationsverzeichnis/jars/3rdparty/IBM*), sollten Sie dies auch bei der neuen Version tun.

Unter Windows x86-64, Linux x86-64 und Linux s390 wird nun eine 64-Bit-Java Runtime Environment (JRE) verwendet. Im Vergleich zu einer 32-Bit-JRE wurden in einigen Szenarios einige Leistungseinbußen beobachtet. Sie können weiterhin das Installationsprogramm für Windows x86-32 für Aktivitäten verwenden, die sich nicht auf die Kennwort-Plug-ins beziehen, wenn Sie der Ansicht sind, dass Leistungseinbußen potenziell Probleme verursachen könnten.

Wenn Sie die 64-Bit-JRE verwenden, müssen Sie beachten, dass gemeinsam genutzte 64-Bit-Bibliotheken für jede angepasste Komponente (Connector, Parser, Funktionskomponente) benötigt werden, die von JNI abhängig ist.

Konfigurationen der Kennwortsynchronisationsprogramme (Password Synchronizer)

- Password Synchronizer für Windows

Befolgen Sie die Schritte, die im Abschnitt "Migration von vorhergehenden Installationen" im Abschnitt "Kennwortsynchronisationsprogramme für Windows" im Abschnitt *Plug-ins für den Kennwortabgleich* im IBM Knowledge Center for IBM Security Directory Integrator beschrieben sind.

- Andere Kennwortsynchronisationsprogramme

Hierfür gibt es keine speziellen Migrationsschritte. Deinstallieren Sie die alte Version, installieren Sie IBM Security Directory Integrator und konfigurieren Sie diese Version entsprechend Ihren Anforderungen.

Wichtige Daten sichern

Nachstehend erhalten Sie umfassende Informationen zum Sichern von Daten.

Durch das Installationsprogramm gesicherte Dateien

Nachstehend finden Sie eine umfassende Liste der vom Installationsprogramm in verschiedenen Versionsupgrades gesicherten Dateien.

Upgrade von Version 6.0 auf Version 7.1:

Falls das Upgrade für die Serverkomponente durchgeführt wird, werden die aufgeführten Dateien gesichert.

```
"tdi-installationsverzeichnis\global.properties" als "tdi-installationsverzeichnis\etc\global.properties.v60"  
"tdi-installationsverzeichnis\serverapi\testadmin.jks" als "tdi-installationsverzeichnis\serverapi\testadmin.jks.v60"  
"tdi-installationsverzeichnis\serverapi\testadmin.der" als "tdi-installationsverzeichnis\serverapi\testadmin.der.v60"  
"tdi-installationsverzeichnis\serverapi\registry.enc" als "tdi-installationsverzeichnis\serverapi\registry.enc.v60"  
"tdi-installationsverzeichnis\serverapi\registry.txt" als "tdi-installationsverzeichnis\serverapi\registry.txt.v60"  
"tdi-installationsverzeichnis\idisrv.sth" als "tdi-installationsverzeichnis\idisrv.sth.v60"  
"tdi-installationsverzeichnis\testserver.jks" als "tdi-installationsverzeichnis\testserver.jks.v60"  
"tdi-installationsverzeichnis\testserver.der" als "tdi-installationsverzeichnis\testserver.der.v60"
```

Außerdem werden Konfigurationsdateien und die Datei `solution.properties` gesichert.

Upgrade von Version 6.1 auf Version 7.1:

Falls die Serverkomponente migriert wird, werden die aufgeführten Dateien gesichert (das neue Suffix lautet je nach Vorversion ".v61" oder ".v611"):

```
"tdi-installationsverzeichnis\etc\global.properties" als "tdi-installationsverzeichnis\etc\global.properties.v61x"  
"tdi-installationsverzeichnis\serverapi\testadmin.jks" als "tdi-installationsverzeichnis\serverapi\testadmin.jks.v61x"  
"tdi-installationsverzeichnis\serverapi\testadmin.der" als "tdi-installationsverzeichnis\serverapi\testadmin.der.v61x"  
"tdi-installationsverzeichnis\serverapi\registry.enc" als "tdi-installationsverzeichnis\serverapi\registry.enc.v61x"  
"tdi-installationsverzeichnis\serverapi\registry.txt" als "tdi-installationsverzeichnis\serverapi\registry.txt.v61x"  
"tdi-installationsverzeichnis\idisrv.sth" als "tdi-installationsverzeichnis\idisrv.sth.v61x"  
"tdi-installationsverzeichnis\testserver.jks" als "tdi-installationsverzeichnis\testserver.jks.v61x"  
"tdi-installationsverzeichnis\testserver.der" als "tdi-installationsverzeichnis\testserver.der.v61x"  
"tdi-installationsverzeichnis\etc\reconnect.rules" als "tdi-installationsverzeichnis\etc\reconnect.rules.v61x"  
"tdi-installationsverzeichnis\etc\derby.properties" als "tdi-installationsverzeichnis\etc\derby.properties.v61x"  
"tdi-installationsverzeichnis\etc\jlog.properties" als "tdi-installationsverzeichnis\etc\jlog.properties.v61x"  
"tdi-installationsverzeichnis\etc\log4j.properties" als "tdi-installationsverzeichnis\etc\log4j.properties.v61x"  
"tdi-installationsverzeichnis\etc\tdisrvctl-log4j.properties" als "tdi-installationsverzeichnis\etc\tdisrvctl-log4j.properties.v61x"  
tdi-installationsverzeichnis\etc\act-jlog.properties als tdi-installationsverzeichnis\etc\act-jlog.properties.v611  
(nur IBM Security Directory Integrator 6.1.1)
```

Außerdem werden Konfigurationsdateien und die Datei `solution.properties` gesichert.

Upgrade von Version 7.0 auf Version 7.1:

Falls das Upgrade für die Serverkomponente durchgeführt wird, werden die aufgeführten Dateien gesichert.

```
"tdi-installationsverzeichnis\etc\global.properties" als "tdi-installationsverzeichnis\etc\global.properties.v70"  
"tdi-installationsverzeichnis\serverapi\testadmin.jks" als "tdi-installationsverzeichnis\serverapi\testadmin.jks.v70"  
"tdi-installationsverzeichnis\serverapi\testadmin.der" als "tdi-installationsverzeichnis\serverapi\testadmin.der.v70"  
"tdi-installationsverzeichnis\serverapi\registry.enc" als "tdi-installationsverzeichnis\serverapi\registry.enc.v70"  
"tdi-installationsverzeichnis\serverapi\registry.txt" als "tdi-installationsverzeichnis\serverapi\registry.txt.v70"  
"tdi-installationsverzeichnis\idisrv.sth" als "tdi-installationsverzeichnis\idisrv.sth.v70"  
"tdi-installationsverzeichnis\testserver.jks" als "tdi-installationsverzeichnis\testserver.jks.v70"
```

```
"tdi-installationsverzeichnis\testserver.der" als "tdi-installationsverzeichnis\testserver.der.v70"
"tdi-installationsverzeichnis\etc\reconnect.rules" als "tdi-installationsverzeichnis\etc\reconnect.rules.v70"
"tdi-installationsverzeichnis\etc\derby.properties" als "tdi-installationsverzeichnis\etc\derby.properties.v70"
"tdi-installationsverzeichnis\etc\jlog.properties" als "tdi-installationsverzeichnis\etc\jlog.properties.v70"
"tdi-installationsverzeichnis\etc\log4j.properties" als "tdi-installationsverzeichnis\etc\log4j.properties.v70"
"tdi-installationsverzeichnis\etc\tdisrvctl-log4j.properties" als "tdi-installationsverzeichnis\etc\tdisrvctl-log4j.properties.v70"
"tdi-installationsverzeichnis\etc\act-jlog.properties" als "tdi-installationsverzeichnis\etc\act-jlog.properties.v70"
```

Außerdem werden Konfigurationsdateien, der Arbeitsbereich und die Datei `solution.properties` gesichert.

Upgrade von Version 7.1 auf Version 7.1.1:

Falls das Upgrade für die Serverkomponente durchgeführt wird, werden die aufgeführten Dateien gesichert.

```
tdi-installationsverzeichnis\etc\global.properties nach tdi-installationsverzeichnis\backup_tdi\global.properties
tdi-installationsverzeichnis\serverapi\testadmin.jks nach tdi-installationsverzeichnis\backup_tdi\testadmin.jks
tdi-installationsverzeichnis\serverapi\testadmin.der nach tdi-installationsverzeichnis\backup_tdi\testadmin.der
tdi-installationsverzeichnis\serverapi\registry.enc nach tdi-installationsverzeichnis\backup_tdi\registry.enc
tdi-installationsverzeichnis\serverapi\registry.txt nach tdi-installationsverzeichnis\backup_tdi\registry.txt
tdi-installationsverzeichnis\idisrv.sth nach
tdi-installationsverzeichnis\backup_tdi\idisrv.sth
tdi-installationsverzeichnis\testserver.jks nach tdi-installationsverzeichnis\backup_tdi\testserver.jks
tdi-installationsverzeichnis\testserver.der nach tdi-installationsverzeichnis\backup_tdi\testserver.der
tdi-installationsverzeichnis\etc\reconnect.rules nach tdi-installationsverzeichnis\backup_tdi\reconnect.rules
tdi-installationsverzeichnis\etc\derby.properties nach tdi-installationsverzeichnis\backup_tdi\derby.properties
tdi-installationsverzeichnis\etc\jlog.properties nach tdi-installationsverzeichnis\backup_tdi\jlog.properties
tdi-installationsverzeichnis\etc\log4j.properties nach tdi-installationsverzeichnis\backup_tdi\log4j.properties
tdi-installationsverzeichnis\etc\tdisrvctl-log4j.properties nach tdi-installationsverzeichnis\backup_tdi\tdisrvctl-log4j.properties
tdi-installationsverzeichnis\etc\tdimigbl-log4j.properties nach tdi-installationsverzeichnis\backup_tdi\tdimigbl-log4j.properties
tdi-installationsverzeichnis\etc\updateinstaller-log4j.properties nach
tdi-installationsverzeichnis\backup_tdi\updateinstaller-log4j.properties
tdi-installationsverzeichnis\etc\it_registry.properties nach tdi-installationsverzeichnis\backup_tdi\it_registry.properties
tdi-installationsverzeichnis\etc\tp.xml nach
tdi-installationsverzeichnis\backup_tdi\tp.xml
```

Außerdem werden Konfigurationsdateien aus dem Ordner `tdi-installationsverzeichnis\configs`, der Arbeitsbereich und die Datei `solution.properties` gesichert.

Upgrade von Version 7.1.1 auf Version 7.2:

Falls das Upgrade für die Serverkomponente durchgeführt wird, werden die aufgeführten Dateien gesichert.

```
tdi-installationsverzeichnis\etc\reconnect.rules
tdi-installationsverzeichnis\etc\derby.properties
tdi-installationsverzeichnis\etc\jlog.properties
tdi-installationsverzeichnis\etc\log4j.properties
tdi-installationsverzeichnis\etc\tdisrvctl-log4j.properties
tdi-installationsverzeichnis\ etc\tdimigbl-log4j.properties
tdi-installationsverzeichnis\ etc\updateinstaller-log4j.properties
tdi-installationsverzeichnis\ etc\it_registry.properties
tdi-installationsverzeichnis\etc\tp.xml
tdi-installationsverzeichnis\ etc\activemq.xml
tdi-installationsverzeichnis\etc\global.properties
solution.properties (sofern im Standardlösungsverzeichnis vorhanden)pwsync.props (für jedes installierte Kennwort-Plug-in)
AMC-Datenbankisc-position/runtime/isc/eclipse/plugins/AMC_7.2.0.0\amc.properties
isc-position/runtime/isc/eclipse/plugins/AMC_7.2.0.0\conf\amcdbhandler.properties
isc-position/runtime/isc/eclipse/plugins/AMC_7.2.0.0\conf\logging.properties
tdi-installationsverzeichnis\bin\amc\ActionManager\am_config.properties
tdi-installationsverzeichnis\bin\amc\ActionManager\am_logging.properties
```

Sicherungstools

Die folgenden Tools werden durch das Installationsprogramm für die Sicherung/Wiederherstellung verwendet.

Sie können auch zur manuellen Migration eingesetzt werden.

backupamc/restreamc

Dieses Tool wird für die Sicherung/Wiederherstellung von AMC-Konfigurationsdateien verwendet.

backupamcdb/restreamcdb

Dieses Tool wird für die Sicherung/Wiederherstellung der AMC-Datenbank verwendet.

backupam/restream

Dieses Tool wird für die Sicherung/Wiederherstellung der Action Manager-Datenbank verwendet.

Weitere Informationen finden Sie im Abschnitt „Befehlszeilendienstprogramme für AMC und Action Manager“ auf Seite 321.

Manuelle Sicherung

Beachten Sie die hier aufgeführten Anweisungen, wenn Sie eine manuelle Sicherung durchführen.

Bei einer manuellen Sicherung wird die entsprechende Datei in einen dedizierten Sicherungsordner kopiert. Umgekehrt bedeutet die Wiederherstellung das Kopieren der Datei aus dem dedizierten Sicherungsordner an ihre ursprüngliche Position.

In einigen Fällen müssen Sie Abhängigkeiten zwischen Dateien beachten. Eine Gruppe voneinander abhängiger Dateien muss in ihrer Gesamtheit gesichert werden. Zu solchen Dateigruppen gehören folgende:

Derby-Datenbankdateien

Sichern Sie bei der Sicherung einer Datenbank den gesamten Ordner, der die Datenbankdateien enthält. Kopieren Sie beispielsweise den Ordner *tdi-installationsverzeichnis/TDISysStore*, um die Standarddatenbank für den Systemspeicher zu sichern, oder kopieren Sie den Ordner *tdi-installationsverzeichnis/bin/amc/tdiamcdb*, um die AMC-Standarddatenbank zu sichern.

IBM WebSphere MQ Everyplace-Warteschlangenmanagerdateien

Sichern Sie den gesamten Ordner des IBM WebSphere MQ Everyplace-Warteschlangenmanagers. Kopieren Sie beispielsweise den Ordner *tdi-installationsverzeichnis/MQePWStore*, um die Standardsystemwarteschlange zu sichern.

Arbeitsbereichsdateien des Konfigurationseditors

Sichern Sie den gesamten Arbeitsbereichsordner.

OSGI Sichern Sie den gesamten OSGI-Ordner.

LDAPSync

Sichern Sie den gesamten LDAPSync-Ordner.

SCIM Sichern Sie den gesamten SCIM-Ordner.

Konfigurationseinstellungen von AMC 7.x auf eine andere AMC-Implementierung migrieren

Mit den hier aufgeführten Anweisungen können Sie AMC-Konfigurationsdaten auf eine neue AMC-Implementierung migrieren.

Anmerkung: Die AMC-Komponente ist veraltet und wird in einer zukünftigen Version von IBM Security Directory Integrator entfernt.

In diesem Abschnitt sind die Schritte erläutert, mit denen Sie AMC-Konfigurationsdaten auf eine neue AMC-Implementierung migrieren können. Diese Anweisungen sind hilfreich, wenn Sie AMC 7.0 und höher aus ISC SE auf IBM Dashboard Ap-

plication Services Hub oder IBM Dashboard Application Services Hub auf ISC SE migrieren bzw. eine Spiegelungsinstanz von AMC auf einer anderen Maschine erstellen.

1. Sichern Sie alle AMC-Konfigurationsdateien und -daten:
 - a. Stoppen Sie die AMC-Instanz, deren Konfigurationsdaten Sie migrieren wollen, mit dem Script `stop_tdiadc.bat (sh)`. Dieses Script stoppt den Server, auf dem AMC implementiert ist.
 - b. Führen Sie das Script `backupamc.bat (.sh)` unter Angabe des Verzeichnisses aus, in dem die AMC-Konfigurationsdateien gesichert werden sollen.
2. Migrieren Sie alle AMC-Konfigurationsdaten auf die neue AMC-Instanz:
 - a. Falls sich die ISC-Implementierung, auf die Sie die AMC-Konfiguration migrieren wollen, auf einer anderen Maschine befindet, müssen Sie das AMC-Sicherungsverzeichnis auf die neue Maschine kopieren.
 - b. Stellen Sie sicher, dass die zu migrierende AMC-Instanz gestoppt ist. Informationen zum Stoppen von AMC enthält Schritt 1a.
 - c. Führen Sie das Script `restoreamc.bat (.sh)` unter Angabe des Verzeichnisses aus, in dem Sie die AMC-Konfigurationsdaten gesichert haben. Dieser Befehl legt die AMC-Konfigurationsdateien an der richtigen Position der AMC-Instanz ab, auf die Sie migrieren. Der Migrationsprozess ist hiermit abgeschlossen.

Anmerkung:

1. Die Anweisungen gelten für eine Migration von AMC 7.0 und höher auf eine beliebige andere ISC-Implementierung.
2. Die Anweisungen setzen voraus, dass AMC bereits sowohl im Quellensystem der Migration als auch im Zielsystem der Migration mit dem Installationsprogramm von IBM Security Directory Integrator implementiert wurde. Auf den Systemen kann entweder AMC in ISC SE oder IBM Dashboard Application Services Hub ausgeführt werden.
3. Falls Sie versuchen, die AMC-Konfiguration auf eine andere AMC-Implementierung zu migrieren, die sich auf derselben Maschine befindet, und alle mit der IBM Security Directory Integrator-Implementierung bereitgestellten AMC-Befehle anschließend diese AMC-Instanz verwenden sollen, müssen Sie die konfigurierte ISC-Position im AMC-Befehlszeilendienstprogramm aktualisieren. Hierzu verwenden Sie den Befehl `setISCHome.bat (sh)`. Der Befehl verwendet als Parameter die Position des ISC-Installationsverzeichnisses, also die Installationsposition von IBM WebSphere Application Server für IBM Dashboard Application Services Hub und die Position der integrierten Webplattform für ISC SE. Der Befehl muss zwischen den vorgenannten Schritten 1 und 2 ausgeführt werden.

EventHandler in entsprechende Fertigungslinie konvertieren

Sie können bestimmte Teile eines typischen EventHandlers migrieren. In den nachstehenden Anweisungen erhalten Sie detaillierte Informationen dazu.

In IBM Security Directory Integrator Version 7.2 gibt es keine EventHandlerer. Um die gelöschte Funktionalität in alten Lösungen zu ersetzen, müssen Sie daher Ihre EventHandlerer-Konfigurationen auf Konfigurationen für Server-/Iterator- oder Änderungsprotokollconnectors migrieren.

Für jeden EventHandlerer muss eine entsprechende Fertigungslinie erstellt werden. Anschließend muss ein Server-/Iteratorconnector, der dem EventHandlerer entspricht, in den Abschnitt "Zuführungen" der Fertigungslinie eingefügt werden. Da-

nach müssen die Connectorparameter festgelegt werden. Dieser Vorgang ist bei jedem EventHandler/Connector-Paar spezifisch. Generell müssen jedoch die Connectorparameter auf dieselben Werte wie die entsprechenden EventHandler-Parameter gesetzt werden (die normalerweise dieselben Namen haben).

Jede im EventHandler konfigurierte Verarbeitung muss im Abschnitt "Datenfluss" der Fertigungslinie erneut implementiert werden.

Die Funktionalität des EventHandler-Parameters für die Aktivierung (enabled), die auch als "automatischer Servicestart" bezeichnet wird, ist bei Fertigungslinien ebenfalls verfügbar. Wenn Ihre Fertigungslinie unmittelbar nach dem Starten des IBM Security Directory Integrator-Servers gestartet werden soll, wechseln Sie im Konfigurationseditor in Ihrem Arbeitsbereich zum Abschnitt **Lösungsprotokollierung und Einstellungen** im Navigator und fügen Sie Ihre Fertigungslinie hinzu.

Ein EventHandler führt in der Regel eine bestimmte Logik aus, wenn ein bestimmtes Ereignis eintritt. Bei jedem EventHandler hat der Begriff "Ereignis" eine andere Bedeutung. Beim EventHandler für HTTP ist ein "Ereignis" eine HTTP-Anforderung. Beim EventHandler für IBM Security Directory Integrator ist ein "Ereignis" eine Benachrichtigung über eine Änderung, die aus einem IBM LDAP-Verzeichnis stammt.

Die folgenden allgemeinen Richtlinien gelten für die Migration bestimmter Teile eines typischen EventHandlers. Sie sind anhand der Titel der Registerkarten in der Benutzerschnittstelle gegliedert, die für einen EventHandler im Konfigurationseditor vor Version 7.0 verwendet wurden:

Hooks

Der Hook "Prolog" eines EventHandlers entspricht dem Hook "Prolog - After Init" einer Fertigungslinie. Dieser Hook wird für jedes eingehende Ereignis aufgerufen.

Der Hook "Epilog" eines EventHandlers entspricht dem Hook "Epilog - After Close" der Fertigungslinie. Dieser Hook wird nach der Verarbeitung jedes eingehenden Ereignisses ein Mal aufgerufen.

In beiden EventHandler-Hooks ("Prolog" und "Epilog") ist der Eintrag für das Ereignis unter den Namen "conn" und "event" zugänglich. In den Hooks der Fertigungslinie sollten Sie Ihr Script jedoch so modifizieren, dass der Name "work" anstelle von "conn" oder "event" verwendet wird.

Der Hook "Shutdown Request" eines EventHandlers entspricht dem Hook "Shutdown Request" einer Fertigungslinie.

Aktionszuordnung

Die Aktionszuordnung eines EventHandlers definiert, welche Aktionen ausgeführt werden sollen, wenn ein Ereignis eintritt. Sie sollten dieselben Aktionen in die Logik der Fertigungslinie einbauen, die Sie als Ersatz für den EventHandler vorbereiten.

Falls die Aktionszuordnung beispielsweise vorschrieb, dass ein angepasstes Script ausgeführt werden soll, wenn das Attribut "x" des Ereignisses den Wert "3" hat, könnten Sie eine Komponente "IF" zur Fertigungslinie hinzufügen, die überprüft, ob das Attribut "x" den Wert "3" hat, und eine Scriptkomponente ausführt.

Protokollierung

Falls Sie angepasste Protokollappender für den EventHandler konfiguriert

haben, sollten Sie dieselben Appender in den Protokollierungseinstellungen der Fertigungslinie(n) konfigurieren, die Sie als Ersatz für den EventHandler vorbereiten

Konfiguration

Die hier angegebenen Konfigurationsparameter sind für jeden EventHandler spezifisch. Anweisungen zu ihrer Migration enthalten die nachfolgenden Unterabschnitte. Die Unterabschnitte sind nach den entsprechenden Connectors benannt.

TCP-Server-Connector

Mit den hier aufgeführten Schritten können Sie die Konfiguration eines alten EventHandlers in der Konfiguration eines neuen Connectors reproduzieren.

1. Erstellen Sie eine neue Fertigungslinie und fügen Sie den TCP-Server-Connector in die Fertigungslinie ein.
2. Legen Sie für die Connectorparameter **tcp.port** und **debug** die Werte der entsprechenden EventHandler-Parameter fest.
3. Setzen Sie die Connectorparameter **useSSL** und **requireClientAuth** auf "false" (dies entspricht einem abgewählten Markierungsfeld im Konfigurationseditor).

Mailbox-Connector

Konfigurationen, die den Mailbox-EventHandler verwenden, müssen Sie mit den nachstehenden Schritten migrieren.

Vorhandene Konfigurationen, die den Mailbox-Connector von IBM Security Directory Integrator Version 6.0 verwenden, müssen nicht migriert werden, weil der Mailbox-Connector von IBM Security Directory Integrator IBM Security Directory Integrator mit dem Mailbox-Connector von IBM Security Directory Integrator Version 6.0 kompatibel ist.

1. Erstellen Sie eine neue Fertigungslinie und fügen Sie den Mailbox-Connector in die Fertigungslinie ein.
2. Kopieren Sie den Inhalt des EventHandler-Parameters **mailServer** in den gleichnamigen Connectorparameter.
3. Setzen Sie den Connectorparameter **mailProtocol** auf den Wert des gleichnamigen EventHandler-Parameters.
4. Kopieren Sie den Inhalt der EventHandler-Parameter **mailUser** und **mailPassword** in die gleichnamigen Parameter des Mailbox-Connectors.
5. Kopieren Sie den Inhalt des EventHandler-Parameters **mailFolder** in den gleichnamigen Connectorparameter.
6. Kopieren Sie den Inhalt des EventHandler-Parameters **pollInterval** in den gleichnamigen Connectorparameter.
7. Falls der EventHandler-Parameter für die Aktivierung (enabled) den Wert "true" hat, fügen Sie Ihre Fertigungslinie zum Ordner "Konfiguration -> Auto-start" im Konfigurationseditor hinzu. Hierdurch startet der IBM Security Directory Integrator-Server bei seinem Start auch die Fertigungslinie.
8. Falls der EventHandler-Parameter **debug** auf "true" gesetzt ist, setzen Sie den gleichnamigen Connectorparameter auf "true".

JMX-Connector

Mit den hier aufgeführten Anweisungen können Sie vorhandene Konfigurationen, die den JMX-EventHandler aus IBM Security Directory Integrator Version 6.0 ver-

wenden, in Konfigurationen von IBM Security Directory Integrator Version 7.2 umwandeln, die den JMX-Connector verwenden.

1. Erstellen Sie eine neue Fertigungslinie und fügen Sie den JMX-Connector in die Fertigungslinie ein.
2. Kopieren Sie den Inhalt des Parameters **eventTypes** für den JMX-EventHandler in den gleichnamigen Parameter des JMX-Connectors.
3. Wählen Sie für den Connectorparameter **mode** den Wert "local" aus.
4. Lassen Sie den Connectorparameter **url** leer.
5. Setzen Sie den Connectorparameter **allMBeans** auf true.
6. Lassen Sie den Connectorparameter **mBeanTypes** leer.

SNMP-Server-Connector

Nachstehend erhalten Sie Informationen zum SNMP-Server-Connector.

Der SNMP-Server-Connector von IBM Security Directory Integrator bietet - mit Ausnahme der Unterstützung für den Einzelthreadmodus - alle Funktionen des SNMP-EventHandlers von IBM Security Directory Integrator Version 6.0. Der SNMP-Server-Connector von IBM Security Directory Integrator kann nur im Multithreadmodus verwendet werden. Falls Sie eine vorhandene Konfiguration von IBM Security Directory Integrator Version 6.0, die den SNMP-EventHandler verwendete, in eine Konfiguration von IBM Security Directory Integrator Version 7.0 migrieren müssen, die eine Fertigungslinie mit dem SNMP-Server-Connector verwendet, müssen Sie Folgendes ausführen:

1. Erstellen Sie eine neue Fertigungslinie.
2. Fügen Sie eine Instanz des SNMP-Server-Connectors in die Fertigungslinie ein.
3. Setzen Sie den Connectorparameter **udp.port** auf den Wert, der für diesen Parameter in der Konfiguration des SNMP-EventHandlers verwendet wurde.
4. Setzen Sie den Connectorparameter **snmp.community** auf den Wert, der für diesen Parameter in der Konfiguration des SNMP-EventHandlers verwendet wurde.
5. Falls für den SNMP-EventHandler das automatische Starten durch den IBM Security Directory Integrator-Server konfiguriert war, fügen Sie die neue Fertigungslinie zum Ordner "Konfiguration -> Autostart" des Konfigurationseditors hinzu.

IBM Security Directory Server-Änderungsprotokollconnector

Eine vorhandene Konfiguration, die den IBM Security Directory Server-EventHandler verwendet, kann mit den nachstehenden Informationen auf die Verwendung des IBM Security Directory Integrator-Änderungsprotokollconnectors migriert werden.

1. Setzen Sie die folgenden Connectorparameter auf die Werte der gleichnamigen EventHandler-Parameter: **ldapUrl**, **ldapUsername**, **ldapPassword**, **ldapAuthenticationMethod**, **ldapUseSSL**, **ldapSearchBase**.
2. Lassen Sie den Connectorparameter **jndiExtraProviderParams** leer.
3. Setzen Sie den Connectorparameter **iteratorStateKey** auf eine beliebige eindeutige Kennung, für die im Systemspeicher kein entsprechender Status gespeichert ist.
4. Setzen Sie den Connectorparameter **nsChangenumber** auf die nächste Änderungsnummer, die vom EventHandler verarbeitet werden würde. Die letzte Änderungsnummer, die der EventHandler verarbeitet hat, ist normalerweise in

einer externen Eigenschaftendatei gespeichert, die durch den Parameter **ldap-ChangeNumberFileName** referenziert wird.

5. Setzen Sie den Connectorparameter **stateKeyPersistence** auf die Einstellung "Nach Lesevorgang". Der EventHandler schreibt die letzte empfangene Änderungsnummer in seine Back-End-Datei, nachdem er einen Änderungsprotokolleintrag gelesen hat und bevor er diesen zur Verarbeitung zuteilt.
6. Setzen Sie den Connectorparameter **mergeMode** auf die Einstellung "Änderungsprotokoll und geänderte Daten mischen". Dies stellt sicher, dass die Änderungsprotokollattribute (`changenumber`, `targetdn`, ...) als Attribute des Eintrags angezeigt werden.
7. Setzen Sie den Connectorparameter **useNotifications** auf "true".
8. Setzen Sie den Connectorparameter **batchRetrieval** auf "false".

Anmerkung: Anders als beim EventHandler ist es beim Connector nicht zulässig, einen Teil der Verzeichnisbaumstruktur auszuwählen, damit der Connector für dessen Benachrichtigungen empfangsbereit ist. Der Connector ist für Änderungen in der gesamten Verzeichnisstruktur subskribiert (es gibt beim Connector keine Entsprechungen für die EventHandler-Parameter **ldapEventBase** und **ldapSearchScope**). Falls dies in Ihrer Umgebung ein kritischer Aspekt ist, können Sie in Ihrer Lösung eine angepasste Filterung implementieren, um diese Einschränkung des Connectors zu umgehen.

HTTP-Server-Connector

Eine Konfiguration, die den HTTP-EventHandler verwendet, kann mit den nachstehenden Schritten auf die Verwendung des HTTP-Server-Connectors migriert werden.

1. Setzen Sie den Wert des Connectorparameters **tcpPort** auf den Wert des EventHandler-Parameters **Port**.
2. Lassen Sie den Connectorparameter **backlog** leer.
3. Setzen Sie den Connectorparameter **contentType** auf "text/html".
4. Setzen Sie den Connectorparameter **tcpDataAsProperties** auf true (der EventHandler gibt die TCP-Informationen immer als Eigenschaften zurück).
5. Setzen Sie den Wert des Connectorparameters **headersAsProperties** auf den Wert des EventHandler-Parameters **headersAsProperties**.
6. Setzen Sie den Connectorparameter **httpAuth** auf "true", falls der EventHandler die HTTP-Basisauthentifizierung verwendet (also ein Authentifizierungsconnector konfiguriert ist).
7. Falls der EventHandler die HTTP-Basisauthentifizierung verwendet, setzen Sie den Connectorparameter **authRealm** auf den Wert des EventHandler-Parameters **authrealm**. Falls der EventHandler-Parameter **authrealm** entweder nicht angegeben oder leer ist, setzen Sie den Connectorparameter **authRealm** auf "IBM-Directory-Integrator".
8. Setzen Sie den Wert des Connectorparameters **authConnector** auf den Wert des EventHandler-Parameters **AuthConnector**.
9. Setzen Sie den Wert des Connectorparameters **useSSL** auf den Wert des EventHandler-Parameters **useSSL**.
10. Setzen Sie den Connectorparameter **needClientAuth** auf "false" (die SSL-Clientauthentifizierung wird vom EventHandler nicht unterstützt).
11. Setzen Sie den Connectorparameter **msgChunked** auf "false" (die Bildung von Chunks aus HTTP-Antworten wird vom EventHandler nicht unterstützt).

LDAP-Server-Connector

Eine Konfiguration, die den LDAP-EventHandler verwendet, kann mit den nachstehenden Anweisungen auf die Verwendung des LDAP-Server-Connectors migriert werden.

1. Setzen Sie den Wert des Connectorparameters **ldapPort** auf den Wert des EventHandler-Parameters **tcp.port**.
2. Lassen Sie den Connectorparameter **backlog** leer.
3. Setzen Sie den Wert des Connectorparameters **ldapUseSSL** auf den Wert des EventHandler-Parameters **ldapUseSSL**.
4. Setzen Sie den Wert des Connectorparameters **charset** auf den Wert des EventHandler-Parameters **charset**.
5. Setzen Sie den Wert des Connectorparameters **ldapBinaryAttributes** auf den Wert des EventHandler-Parameters **binary**.

Änderungserkennungsconnector für Sun Directory

Nachstehend erhalten Sie Informationen zum Änderungserkennungsconnector für Sun Directory.

Der LDAP-EventHandler fängt Benachrichtigungen über Änderungen in einer Verzeichnisstruktur ab. Er verwendet kein Änderungsprotokoll und empfängt daher nur Echtzeitbenachrichtigungen. Der Sun Directory-Änderungserkennungsconnector bietet bei einer Ausführung im Modus für die Echtzeitzustellung im Grunde genommen dieselbe Funktionalität. Einige wenige Unterschiede sind jedoch zu beachten:

Der Connector besitzt keine Entsprechungen für die EventHandler-Parameter **ldapSearchFilter** und **ldapSearchScope**. Um dieselbe Funktionalität wie beim EventHandler zu erzielen, sollten Sie eine angepasste Filterung implementieren, die die Menge der empfangenen Benachrichtigungen begrenzt.

Im Hinblick auf das Schema der zurückgegebenen Daten gibt es zwischen dem Connector und dem EventHandler Unterschiede. Der Connector wendet auf jeden Eintrag, den er zurückgibt, das Deltatagging an, wohingegen der EventHandler den Typ der Änderung in der Eigenschaft "ldap.operation" bereitstellt. Details über das Schema können Sie der Dokumentation für die jeweilige Komponente entnehmen.

Unter Berücksichtigung der obigen Hinweise können Sie eine vorhandene Konfiguration mit dem LDAP-EventHandler folgendermaßen für die Verwendung des Sun Directory-Änderungserkennungsconnectors migrieren:

1. Setzen Sie die folgenden Connectorparameter auf die Werte der gleichnamigen EventHandler-Parameter: **ldapUrl**, **ldapUsername**, **ldapPassword**, **ldapAuthenticationMethod**, **ldapUseSSL**, **ldapSearchBase**.
2. Lassen Sie den Connectorparameter **jndiExtraProviderParams** leer.
3. Setzen Sie den Connectorparameter **deliveryMode** auf "Realtime" (der EventHandler verwendet kein Änderungsprotokoll, sondern fängt ausschließlich Echtzeitbenachrichtigungen ab).
4. Setzen Sie den Connectorparameter **mergeMode** auf "Nur geänderte Daten zurückgeben" (im Modus für die Echtzeitzustellung wird durch den Connector kein Änderungsprotokoll verwendet).

Änderungserkennungsconnector für Active Directory

Nachstehend erhalten Sie Informationen zum Änderungserkennungsconnector für Active Directory.

Die Migration vom Active Directory-Änderungsprotokoll-EventHandler in den für Active Directory-Änderungserkennungsconnector ist in vielerlei Hinsicht direkt möglich, weil im EventHandler selbst die ältere Version dieses Connectors (AD-Änderungsprotokollconnector) integriert ist, um Änderungen von Active Directory (AD) zu empfangen.

Ähnlich wie der EventHandler kann auch der entsprechende Connector während des Synchronisationsprozesses jederzeit unterbrochen werden. In diesem Fall wird sein Status im Eigenschaftsspeicher des Benutzers gespeichert. Sowohl der EventHandler als auch der Connector verwenden in diesem Prozess den Mechanismus "uSNChanged", bei dem die USN-Nummer im Eigenschaftenspeicher abgelegt wird. Außerdem bieten beide eine SN-API zum Abrufen der aktuellen USN-Synchronisationswerte. Der Unterschied besteht darin, dass die Methode `getUSNvalues` des EventHandlers einen Eintrag mit Attributen zurückgibt:

```
START_USN
END_USN
CURRENT_USN_CREATED
CURRENT_USN_CHANGEDT
```

Der Connector gibt hingegen den aktuellen Synchronisationswert mit dem Typ *long* zurück.

Ebenfalls unterschiedlich ist die Tatsache, dass der AD-EventHandler intern einen LDAP-Connector initialisiert, um Benachrichtigungen über Änderungen zu blockieren und zu empfangen. Dieses Verhalten kann durch den Active Directory-Änderungserkennungsconnector ebenfalls simuliert werden, indem der Parameter **useNotifications** aktiviert wird.

Zur Migration einer EventHandler-basierten Lösung auf eine connectorbasierte Lösung müssen die folgenden Schritte ausgeführt werden:

1. Erstellen Sie eine neue Fertigungslinie mit einer Instanz des Active Directory-Änderungserkennungsconnectors im Iteratormodus.
2. Setzen Sie die Parameter **ldapUrl**, **ldapUsername**, **ldapPassword** und **ldapAuthenticationMethod** auf die Werte, die für diese Verbindungsparameter in der EventHandler-Konfiguration verwendet wurden.
3. Geben Sie gemäß dem Wert in der alten Konfiguration an, ob eine SSL-Verbindung verwendet wird.
4. Kopieren Sie den Inhalt des EventHandler-Parameters **ldapSearchBase** in den gleichnamigen Parameter der Connectorkonfiguration.
5. Kopieren Sie den Inhalt des EventHandler-Parameters **persistentParameterName** in den Connectorparameter "persistentStateKey".
6. Setzen Sie den Parameter **useNotifications** auf "true".
7. Legen Sie den Wert für den Parameter **startAt** gemäß dem Wert im EventHandler fest.
8. Nehmen Sie an den übrigen Connectorparametern keine Änderung vor.
9. Übertragen Sie die gesamte Logik im EventHandler-Abschnitt für die Aktionszuordnung, damit sie aus der neuen Fertigungslinie heraus aufgerufen wird.

DSMLv2-SOAP-Server-Connector

Nachstehend erhalten Sie Informationen zum DSMLv2-SOAP-Server-Connector.

Zur Migration vom DSMLv2-EventHandler auf den DSMLv2-SOAP-Server-Connector müssen die Fertigungslinien, die zuvor mit dem EventHandler verwendet wurden, überarbeitet werden, damit sie in die Lösung mit dem DSMLv2-SOAP-Server-Connector integriert werden können. Dies liegt daran, dass die Kernarchitektur geändert wurde und nun eine einzige Fertigungslinie alle Operationen verarbeitet. Daher sollte die gesamte alte Fertigungslinienlogik, mit der die verschiedenen Typen von DSMLv2-Operationen verarbeitet wurden, in die neue Fertigungslinie, die den DSMLv2-SOAP-Server-Connector enthält, integriert oder unter Verwendung eines Fertigungslinienconnectors aufgerufen werden. Zu diesem Zweck können Verzweigungskomponenten verwendet werden, um die Logik für die speziellen DSMLv2-Operationen (verfügbar im Attribut `dsm1.operation`) voneinander abzugrenzen.

Die Migration einer Konfiguration mit dem DSMLv2-EventHandler auf eine ähnliche Konfiguration mit dem DSMLv2-SOAP-Server-Connector umfasst die folgenden Schritte:

1. Erstellen Sie eine neue Fertigungslinie mit einer Instanz des DSMLv2-SOAP-Server-Connectors im Servermodus.
2. Kopieren Sie den Inhalt des EventHandler-Parameters **port** in den Connectorparameter **dsm1Port**.
3. Setzen Sie die Parameter **authRealm**, **useSSL**, **binaryAttributes** und **msg-Chunked** auf die Werte, die für diese Verbindungsparameter in der EventHandler-Konfiguration verwendet wurden.
4. Erstellen Sie eine Verzweigungskomponente für jede DSMLv2-Operation, die als Parameter in der EventHandler-Konfiguration aufgeführt ist, und wenden Sie in den Verzweigungen die Logik an, die in der entsprechenden alten Fertigungslinie implementiert ist - entweder durch eine Übertragung der entsprechenden Fertigungslinienkomponenten oder durch einen Aufruf der eigentlichen alten Fertigungslinie mit einem Fertigungslinienconnector. In beiden Fällen wird der Namenskontext nicht mehr benötigt.
5. Kopieren Sie den Inhalt der beiden EventHandler-Parameter **WaySSL** in den Connectorparameter **needClientAuth**.
6. Das EventHandler-Attribut **headerAsProperties** kann nicht an den Connector übergeben werden, weil der von ihm intern initialisierte HTTP-Parser so konfiguriert ist, dass dieser Wert immer auf "false" gesetzt ist. Falls die Lösung auf Header als Eigenschaften zugreift, sollte aus diesem Grund dahingehend eine Modifizierung erfolgen, dass zu diesem Zweck Attribute verwendet werden (`getAttribute()` anstelle von `getProperty()`).
7. Aus Kompatibilitätsgründen sollte das Connectorattribut **soapbinding** auf "false" gesetzt sein, da der intern vom EventHandler verwendete DSMLv2-Parser dieses Attribut nicht nutzt.
8. Falls in der Konfiguration des DSMLv2-EventHandlers ein Parameter **authConnector** angegeben ist, muss die HTTP-Basisauthentifizierung des Connectors aktiviert und die entsprechende Logik im Hook "After Accepting connection" implementiert werden. (Ein Beispiel hierfür ist die Initialisierung des Authentifizierungsconnectors und der Aufruf seiner Methode `lookup()` unter Verwendung eines Eintrags mit den Attributen "username" und "password" als Suchkriterien. Ähnlich wie beim EventHandler gilt die Authentifizierung bei der Rückgabe eines Eintrags als erfolgreich.)
9. Der Parameter **indentoutput** des intern vom Connector verwendeten DSMLv2-Parsers kann (anders als beim EventHandler) nicht festgelegt werden.

B-Baum-Tabellen und B-Baum-Connector auf Systemspeicher migrieren

Nachstehend erhalten Sie Informationen zum Migrieren von B-Baum-Tabellen und des B-Baum-Connectors auf Systemspeicher.

Der B-Baum-Connector ist veraltet und wird nunmehr lediglich als nicht unterstütztes Beispiel bereitgestellt. Daher kann es sinnvoll sein, zur Verwaltung der Deltainformationen nicht mehr die alten B-Baum-Objekte, sondern die Deltatabellen des Systemspeichers zu verwenden. Die beste Strategie besteht darin, zu diesem Zweck eine Situation zu konstruieren, in der die Deltainformationen leer sind (beispielsweise durch den Aufbau einer neuen Referenzkonfiguration) und dann von den B-Baum-Objekten zu den Deltatabellen des Systemspeichers zu wechseln. Bitte beachten Sie, dass der Parameter, der früher für den Dateinamen der B-Baum-Objekte verwendet wurde, jetzt einen Tabellennamen in einer Datenbank angibt, weshalb dieser Wert möglicherweise bearbeitet werden muss.

Die Änderung einer Lösung in die Verwendung des Systemspeicherconnectors anstelle des B-Baum-Connectors zum Speichern von IBM Security Directory Integrator-Einträgen ist direkt möglich, weil beide Connectors bei der Angabe der Attribute "Schlüsselattributname" und "Auswahlmodus" dieselbe Logik befolgen. Der einzige Unterschied besteht darin, dass der Systemspeicherconnector nicht die zugrunde liegende B-Baum-Datenbank, sondern eine vordefinierte Datenbank (z. B. die integrierte Derby-Datenbank) verwenden und eine Tabelle zum Speichern der Daten angeben muss.

Die Speicherung von Objekten, bei denen es sich nicht um Java-Objekte handelt, weicht bei Verwendung des Systemspeicherconnectors erheblich von der Speicherung mit dem B-Baum ab, weshalb hier eine komplexere Umsetzung erfolgen muss. Die folgende Lösung, die Java-Objekte in der zugrunde liegenden B-Baum-Datenbank ablegt, kann nicht direkt auf den Systemspeicherconnector angewendet werden, weil sie keinen direkten Zugriff auf die Back-End-Datenbank bietet:

```
scripts var bt = system.getConnector("btreedb");
bt.initialize (null); var db = bt.getDatabase();
db.insert ("my key", new java.lang.String("my value"));
var value = db.search ("my key"); value = value + " - modified";
db.replace ("my key", value);
```

Stattdessen können die Standardmethoden (`put()`, `find()` und `modify()`) aus der Connector-API verwendet werden. Das Objekt sollte jedoch zuerst in ein Eintragsobjekt eingeschlossen werden, das anschließend im Systemspeicher abgelegt werden kann.

Cloudscape-Datenbank auf Derby migrieren

Mit den hier aufgeführten Anweisungen können Sie eine Cloudscape-Datenbank auf Derby migrieren.

IBM Security Directory Integrator Version 7.2 verwendet als paketierte Datenbank Apache Derby Version 10.8, die standardmäßig vom Systemspeicher verwendet wird. Vorhandene Cloudscape- oder Derby-Datenbanken (die mit Vorversionen von IBM Security Directory Integrator erstellt wurden) müssen migriert werden, damit IBM Security Directory Integrator Version 7.2 verwendet werden kann. Die mit IBM Security Directory Integrator ausgelieferten Treiber für Apache Derby Version 10.8 Version 7.2 können nicht zur Kommunikation mit älteren Cloudscape-Versionen eingesetzt werden.

Details sowie Informationen zu Unterschieden zwischen Cloudscape/Apache Derby Version 10.8 und seinen Vorversionen finden Sie auf der folgenden Webseite: <http://publibfp.boulder.ibm.com/epubs/html/c1894710.html>.

Die folgenden zu beachtenden Unterschiede können sich direkt auswirken:

- Der Datentyp `long varbinary` wird nicht mehr unterstützt. Stattdessen wurde der Datentyp `BLOB` eingeführt (wodurch Derby mit DB2 kompatibel wird). Aus diesem Grund müssen alle SQL-Anweisungen, die den Datentyp "long varbinary" verwendeten, jetzt in die Verwendung des Datentyps `BLOB` modifiziert werden.
- JDBC-Java-Paketnamen wurden von "com.ibm.db2j.*" bei den Vorgängerreleases in "org.apache.derby.*" bei Derby Version 10 geändert.
- Die JDBC-URL für Derby Version 10 (integrierter und Netzmoduszugriff) ist anders als bei Cloudscape 5.1. Daher wurden die in der Datei `global.properties` / `solution.properties` angegebenen JDBC-Eigenschaften für die aktuelle Version von IBM Security Directory Integrator ebenfalls modifiziert.

Tabelle 12. Unterschiede bei JDBC-URL

Verbindungstyp	Cloudscape Version 5.1	Derby Version 10
Derby / Cloudscape integriert	<code>jdbc:db2j:</code>	<code>jdbc:derby:</code>
JDBC-Treiber von DB2 Universal Database (Netzmodus)	<code>jdbc:db2j:net</code>	<code>jdbc:derby:net</code> (Verwendung wird nicht empfohlen)
Derby-Clienttreiber	-	<code>jdbc:derby</code> (Empfohlen)

Das Derby-Team hat ein Migrationsdienstprogramm bereitgestellt, das eine Datenbank von Cloudscape Version 5.1 auf eine neue Datenbank von Derby Version 10 migriert. Das Dienstprogramm migriert alle Tabellen und ihre entsprechenden Daten auf eine neu generierte Datenbank von Derby Version 10. Alle Tabellen mit dem Datentyp "varbinary" werden in den Datentyp `BLOB` modifiziert, was den Migrationsprozess relativ unproblematisch macht.

Dieses Dienstprogramm ist im Paket von IBM Security Directory Integrator im Ordner `tdi-installationsverzeichnis/tools/CSMigration` zusammen mit einem Wrapper-Script namens "migrateCS.bat(sh)" enthalten, von dem das Migrationstool aufgerufen wird. Um eine mit IBM Security Directory Integrator Version 6.0 erstellte Cloudscape 5.1-Datenbank für den Systemspeicher auf Derby Version 10 zu migrieren, müssen Sie das Migrationsscript folgendermaßen aufrufen:

```
migrateCS [pfad_der_cloudscapev51-datenbank] [pfad_der_neuen_derbyv10-datenbank]
```

Anmerkung: Dieses Migrationsdienstprogramm kann nur für die Migration von Cloudscape 5.1 auf Derby Version 10 verwendet werden. Daher kann die Datei `tdi-installationsverzeichnis/tools/CSMigration/migrateCS.bat(sh)` zur Migration der Systemspeicherdatenbank von IBM Security Directory Integrator Version 6.0 auf Versionen ab 6.1.1 verwendet werden. Zur Migration der Systemspeicherdatenbank von IBM Security Directory Integrator Version 6.1.1 und späteren Versionen müssen Sie nur den alten Systemspeicher (TDISysStore) aus dem Installationsverzeichnis von Version 6.1.1 in die neue Installation der neuen Version kopieren.

Möglicherweise müssen Sie hinsichtlich der Position für die neue Derby-Datenbank einige Punkte beachten. In IBM Security Directory Integrator Version 6.0 und 6.1.x befand sich die Systemspeicherdatenbank häufig im Installationsverzeichnis von IBM Security Directory Integrator. Diese Position ist aus vielen Gründen keine gute Wahl. Es wird dringend empfohlen, bei IBM Security Directory Integrator Version 7.2 ein Lösungsverzeichnis und nicht das Installationsverzeichnis zu verwenden.

Neben der Migration von Daten müssen Sie auch Ihre Dateien `global.properties` / `solution.properties` (mit dem Migrationstool oder manuell) so modifizieren, dass die neuen Parameter für die JDBC-URL enthalten sind.

Dateien "global.properties" und "solution.properties" mit dem Migrationstool migrieren

Verwenden Sie das Tool `tdimiggb1`, das sich im Verzeichnis `tdi-installationsverzeichnis/bin` befindet, um jede Datei `global.properties` ab IBM Security Directory Integrator 6.x auf Version 7.2 zu migrieren.

Der Dateiname des Tools lautet `tdimiggb1.bat` (Windows) bzw. `tdimiggb1.sh` (UNIX/Linux). Verwenden Sie die Datei `tdimiggb1-4log4j.properties`, um die Protokollierung für `tdimiggb1.bat(sh)` zu steuern. Der Befehl hat die folgende Syntax:

```
tdimiggb1 -f eigenschaftendatei [-b sicherungsdatei] [-n neue_datei] [-v] [-?]
```

Hierbei gilt Folgendes:

- f eigenschaftendatei - Name der zu migrierenden Datei
- b sicherungsdatei - Ursprüngliche Datei unter dem angegebenen Namen sichern
- n neue_datei - Name für die migrierte Datei
- s verzeichnis - Arbeitsverzeichnis, in dem sich das Lösungsverzeichnis befindet
- v - Ausführlichen Modus aktivieren
- ? - Verwendungsanweisung ausgeben

Während der Installation von IBM Security Directory Integrator sichert das Installationsprogramm die vorhandene Datei `global.properties` und ruft anschließend diesen Befehl auf, um die Datei `global.properties` zu migrieren.

Das Migrationstool versucht, eine Datei `global.properties` (oder bei Bedarf eine Datei `solution.properties`) auf die neueste Version von IBM Security Directory Integrator zu migrieren. Es bestehen bei dem Tool (`tdimiggb1`) keine Annahmen hinsichtlich des Releases, bei dem die Dateien `global.properties` beginnen. Das Tool kann Dateien `global.properties` ab IBM Security Directory Integrator Version 6.0 verarbeiten. Außerdem versucht es, alle Migrationsänderungen anzuwenden, es sein denn, ein bestimmter Migrationsschritt ist ausdrücklich als für die Migration durch das Migrationstool ungeeignet deklariert. In diesen Fällen müssen die Migrationsschritte manuell ausgeführt werden.

Die Aktivitäten des Migrationstools sind in verschiedene Phasen untergliedert. Das Tool führt nacheinander Folgendes aus:

1. Es überprüft, ob Ihre Derby-Datenbank (Cloudscape-Datenbank) migriert werden muss (Migration von IBM Security Directory Integrator 6.0).
2. Es führt alle Migrationsaktionen in der folgenden Reihenfolge aus:
 - a. Löschaktionen
 - b. Hinzufügungsaktionen
 - c. Dateiänderungen zur Migration von Derby (Cloudscape) (nur erforderlichenfalls und nur bei Migrationen von IBM Security Directory Integrator 6.0)
 - d. Migrationsmodifizierungsaktionen
3. Es ruft das Tool für die Migration von Derby (Cloudscape) namens `migrateCS` auf, um die Datenbank bis auf die aktuelle Derby-Version zu migrieren (nur bei Migrationen von IBM Security Directory Integrator 6.0).

Bei jeder Aktionsgruppe (beispielsweise den Migrationsmodifizierungsaktionen) versucht das Migrationstool, die Migrationsaktionen ab dem frühesten Release bis zum aktuellen Release auszuführen. Zur Migration ab IBM Security Directory Integrator 6.0 muss das aufrufende Modul das Migrationstool für Derby (Cloudscape) separat aufrufen, damit die Datenbank bis auf die aktuelle Derby-Version migriert wird. Das Tool `tdimiggb1` nimmt nur die erforderlichen Modifizierungen für Derby (Cloudscape) an der Eigenschaftendatei selbst vor.

4. Es verwendet die `log4j`-Protokollierungs-APIs zur Protokollierung von Fehler-
nachrichten.

Die `log4j`-Konfigurationsdatei ist im Startscript (Datei `".bat"` oder `".sh"`) angegeben. Der Befehl verwendet eine Datei namens `tdimiggb1-log4j.properties`, um die `log4j`-Protokollierung zu konfigurieren. Der Befehl ändert das Verzeichnis in das Lösungsverzeichnis und verwendet daher die Datei `tdimiggb1-log4j.properties` im Lösungsverzeichnis, falls das Installationsverzeichnis von IBM Security Directory Integrator nicht angegeben ist.

Eigenschaftendateien der Kennwort-Plug-ins mit dem Migrationstool migrieren

IBM Security Directory Integrator enthält ein Migrationsdienstprogramm, um für die Dateien `pwsync.props` aller installierten Kennwort-Plug-ins ein Upgrade durchzuführen. Das Dienstprogramm heißt `migpwsync` und wird zur Migration der Dateien vom Typ `pwsync.props` bereitgestellt, die sowohl vom nativen Plug-in als auch vom Java-Proxy gelesen werden.

Das Dienstprogramm `"migpwsync"` wird im Verzeichnis `tdi-installationsverzeichnis/pwd_plugins/bin` ausgeliefert.

Es besitzt die folgenden Optionen:

- `-?` - Bei Verwendung dieser Option gibt das Dienstprogramm den Hilfetext aus und wird beendet.
- `-v` - Bei Verwendung dieser Option gibt das Dienstprogramm ausführlichere Informationen als bei der Standardausgabe aus.
- `-f` - Diese Option ist erforderlich. Mit ihr wird die Position der Datei `pwsync.props` angegeben.
- `-b` - Diese Option gibt die Position der Datei an, die für die Sicherung verwendet wird. Es handelt sich um ein optionales Feld. Wenn es nicht angegeben ist, wird der Wert der Option `"-f"` verwendet, an den die Zeichenfolge `".backup"` angehängt wird.
- `-n` - Diese Option gibt die Position der Datei an, in die die migrierten Informationen geschrieben werden. Es handelt sich um ein optionales Feld. Wenn es nicht angegeben ist, wird der Wert der Option `"-f"` als Position für die Ausgabe der migrierten Konfiguration verwendet.

Beispiele

- Migration der Konfigurationsdatei für das PAM-Plug-in:

```
# tdi-installationsverzeichnis/pwd_plugins/bin/migpwsync.sh -f tdi-installationsverzeichnis/  
pwd_plugins/pam/pwsync.props
```
- Migration der Konfigurationsdatei für das Windows-Plug-in:

```
> tdi-installationsverzeichnis\pwd_plugins\bin\migpwsync.bat  
- f tdi-installationsverzeichnis\pwd_plugins\windows\pwsync.props
```

Anmerkung:

1. Das Installationsprogramm aktualisiert alle Dateien vom Typ `pwsync.props`, die während der Installation von ihm im Verzeichnis `tdi-installationsverzeichnis/pwd_plugins` eingerichtet wurden. Falls Sie eine der Dateien `pwsync.props` an eine andere Position versetzt haben, müssen Sie sie mit einem Befehl ähnlich der oben angegebenen manuell migrieren.
2. Das Dienstprogramm "migpwsync" ändert das aktuelle Verzeichnis in das Ausgangsverzeichnis für die Plug-ins (`tdi-installationsverzeichnis/pwd_plugins`.) Die angegebenen Dateipfade werden als relative Pfade interpretiert, die sich auf dieses Verzeichnis beziehen, sofern es sich nicht um absolute Pfade handelt.
3. Nach der Migration der alten Datei des Typs `pwsync.props` fügen Sie die folgenden zugehörigen ActiveMQ-Eigenschaften hinzu, wenn Sie ActiveMQ als standardmäßigen JMS-Kennwortspeicher konfigurieren möchten:
 - `jmsDriverClass=com.ibm.di.plugin.pwstore.jms.driver.ActiveMQ`
 - `jms`
 `oker=<JMS-Serveradresse>`. Beispiel: `jms`
 `oker=tcp://<activeMQhost>:61616` or `jms`
 `oker=ssl://<activeMQhost>:61617`
4. Setzen Sie die Eigenschaft `jms.clientId` in der Datei `pwsync.props`, um ActiveMQ als standardmäßigen JMS-Kennwortspeicher zu konfigurieren.

Kapitel 6. Sicherheit

Mit den hier aufgeführten Anweisungen können Sie die Sicherheitseinrichtungen bearbeiten und verwenden sowie Fehler beheben.

IBM Security Directory Integrator (IBM Security Directory Integrator) enthält eine Vielzahl von Sicherheitseinrichtungen. Einige Funktionen schützen den Zugriff von IBM Security Directory Integrator auf ferne Systeme, andere schützen den Zugriff von fernem Systemen auf IBM Security Directory Integrator und wieder andere Funktionen stellen Mechanismen für den Schutz von Daten (beispielsweise Benutzerberechtigungsmechanismen in fernem Systemen) bereit.

Viele der in diesem Abschnitt beschriebenen Funktionen sind bei der Ausführung von IBM Security Directory Integrator im Standalone-Modus in einer gesicherten Umgebung nicht erforderlich. Die Funktionen erweisen sich jedoch als nützlich, wenn andere Systeme mit IBM Security Directory Integrator kommunizieren müssen, beispielsweise über das Verwaltungstool für die ferne Webadministrationskonsole (AMC) oder die ferne Server-API von IBM Security Directory Integrator. Wenn mehrere Personen auf den IBM Security Directory Integrator-Server zugreifen können, kann es darüber hinaus erforderlich sein, den Zugriff auf vertrauliche Daten zu schützen sowie die Integrität der von IBM Security Directory Integrator ausgeführten Integrationsregeln zu erhalten.

Im vorliegenden Abschnitt werden die folgenden Funktionen erläutert:

1. „Schlüssel, Zertifikate und Schlüsselspeicher verwalten“
2. „Unterstützung für Secure Sockets Layer (SSL)“ auf Seite 110
3. „Ferne Server-API“ auf Seite 119
4. „Sicherheit für IBM Security Directory Integrator-Serverinstanz“ auf Seite 146
5. „Verschiedene Funktionen für Konfigurationsdateien“ auf Seite 157
6. „Sicherheit für die Webverwaltungskonsole“ auf Seite 166
7. „Übersicht über Konfigurationsdateien und Eigenschaften für Sicherheit“ auf Seite 162
8. „Sonstige Sicherheitsaspekte“ auf Seite 166

In diesem Abschnitt wird nicht das gesamte sicherheitsbezogene Leistungsspektrum der einzelnen IBM Security Directory Integrator-Komponenten beschrieben. Der Abschnitt „Sonstige Sicherheitsaspekte“ auf Seite 166 enthält Beschreibungen für einige allgemeine Elemente, wenn Sie jedoch Informationen zu einzelnen Elementen der Sicherheitskonfiguration in den verschiedenen IBM Security Directory Integrator-Komponenten benötigen, lesen Sie den Abschnitt *Referenzinformationen* im IBM Knowledge Center for IBM Security Directory Integrator.

Schlüssel, Zertifikate und Schlüsselspeicher verwalten

Nachstehend erhalten Sie Informationen zur Verwaltung verschiedener Schlüsseltypen, zu deren Auflistung in einem Schlüsselspeicher und zur Schlüsselerstellung.

Hintergrund

Unter den hier aufgeführten Links erhalten Sie Informationen zu Schlüsseln in SSL, zur Verschlüsselung und zu Sicherheitskonzepten.

Verschlüsselungsschlüssel werden im Produkt hauptsächlich im Zusammenhang mit SSL (siehe „Unterstützung für Secure Sockets Layer (SSL)“ auf Seite 110) und zur Verschlüsselung (siehe „Sicherheit für IBM Security Directory Integrator-Serverinstanz“ auf Seite 146) eingesetzt.

Ausführliche Informationen zu Sicherheitskonzepten sowie deren Verwendung in IBM JVM finden Sie unter der Adresse <http://www.ibm.com/developerworks/java/jdk/security/>.

Öffentliche/private Schlüssel und Zertifikate

Nachstehend erhalten Sie Informationen zu öffentlichen und privaten Schlüsseln und dazu, wie diese unabhängig voneinander bzw. miteinander verwendet werden.

SSL und asymmetrische Verschlüsselungsalgorithmen wie RSA (der Standardverschlüsselungsalgorithmus des Servers) verwenden öffentliche/private Schlüssel. Öffentliche und private Schlüssel weisen eine Eins-zu-eins-Entsprechung auf. Die zueinander gehörenden öffentlichen und privaten Schlüssel werden als "Schlüssel-paar" bezeichnet.

Normalerweise ist ein öffentlicher Schlüssel in einem Schlüsselspeicher in ein X.509-Zertifikat eingeschlossen. Die meisten Schlüsselspeicheroperationen betreffen das gesamte Zertifikat für den öffentlichen Schlüssel und nicht nur den öffentlichen Schlüssel selbst.

Auch wird für einen privaten Schlüssel in einem Schlüsselspeicher das entsprechende Zertifikat für den öffentlichen Schlüssel bereitgestellt.

Geheime Schlüssel

Nachstehend erhalten Sie Informationen dazu, welche Algorithmen und Schlüsselspeicher geheime Schlüssel verwenden.

Geheime Schlüssel werden von symmetrischen Verschlüsselungsalgorithmen wie DES (Data Encryption Standard), AES (Advanced Encryption Standard) und RC4 verwendet. Bitte beachten Sie, dass einige Schlüsselspeicherformate (z. B. JKS und PKCS#12) geheime Schlüssel nicht unterstützen.

Die Verwendung geheimer Schlüssel für SSL ist nicht möglich (das SSL-Protokoll generiert zwar während der Verarbeitung geheime Schlüssel, aber diese können von Ihnen nicht gesteuert werden).

Schlüsselspeicher

Nachstehend erhalten Sie Informationen zu Schlüsselspeichern, ihren Dateiformaten und ihrem Ursprung sowie zum Vergleich verschiedener Schlüsselspeicher.

Ein Schlüsselspeicher stellt, wie der Name bereits andeutet, den Speicher für Schlüssel bereit. Hierbei kann es sich um eine Datei oder eine Hardwareeinheit handeln. Die gängigsten Formate für Schlüsselspeicherdateien, die durch Java-Programme verwendet werden, sind JKS, JCEKS und PKCS#12. Die folgende Tabelle enthält Vergleichsdaten für diese Formate:

Tabelle 13. Formate der Schlüsselspeicherdatei

Format der Schlüsselspeicherdatei	Ursprung	Speicher für öffentliche/private Schlüssel und Zertifikate	Speicher für geheime Schlüssel
JKS	Proprietär	Ja	Nein
JCEKS	Proprietär	Ja	Ja

Tabelle 13. *Formate der Schlüsselspeicherdatei (Forts.)*

Format der Schlüsselspeicherdatei	Ursprung	Speicher für öffentliche/private Schlüssel und Zertifikate	Speicher für geheime Schlüssel
PKCS#12	Standard	Ja	Nein

Bitte beachten Sie, dass unter den obigen Schlüsselspeicherformaten lediglich das Format JCEKS die Speicherung geheimer Schlüssel zulässt. JCEKS bietet zusätzlich generell einen größeren Schutz als JKS. JKS-, JCEKS- und PKCS#12-Schlüsselspeicher werden durch ein Kennwort geschützt. Darüber hinaus kann jeder private oder geheime Schlüssel in einem Schlüsselspeicher durch ein individuelles Kennwort geschützt werden. Zertifikate für öffentliche Schlüssel sind nicht mit einem Kennwort versehen, weil sie normalerweise nicht geheimgehalten werden müssen.

Schlüssel für SSL

Zum Arbeiten mit SSL ist eine Gruppe von öffentlichen/privaten Schlüsseln erforderlich. Nachstehend erhalten Sie Informationen dazu, wie Sie diese Schlüssel einrichten.

Informationen zu diesem Vorgang

Ausführliche Informationen zur Verwendung von SSL mit IBM JVM finden Sie unter der Adresse <http://www.ibm.com/developerworks/java/jdk/security/60/secguides/jsse2Docs/JSSE2RefGuide.html>.

Zur Verwendung von SSL müssen Sie eine Gruppe von öffentlichen/privaten Schlüsseln bereitstellen. Geheime Schlüssel können für SSL nicht verwendet werden.

Eine SSL-Verbindung besitzt zwei Seiten, nämlich die SSL-Serverseite und die SSL-Clientseite. Auf jeder Seite sind zwei Schlüsselspeicher vorhanden (nämlich ein SSL-Schlüsselspeicher und ein SSL-Truststore). Der Begriff "Schlüsselspeicher" wird synonym für einen Speicher für Schlüssel und für einen SSL-Schlüsselspeicher verwendet. Daher handelt es sich sowohl beim SSL-Schlüsselspeicher als auch beim SSL-Truststore um einen Schlüsselspeicher. Eigentlich sind der SSL-Schlüsselspeicher und der SSL-Truststore nur logische Rollen. Daher ist es ohne Weiteres möglich, für beide dieselbe physische Schlüsselspeicherdatei zu verwenden. Der SSL-Schlüsselspeicher enthält einen privaten Schlüssel, mit dem der anderen Seite einer SSL-Verbindung die Authentizität dieser SSL-Seite nachgewiesen wird. Der SSL-Truststore enthält Zertifikate für öffentliche Schlüssel von vertrauenswürdigen Parteien.

Vorgehensweise

1. Zur Konfiguration von Schlüsseln für Ihren SSL-Server können Sie einen privaten Schlüssel sowie ein entsprechendes selbst signiertes Zertifikat für den öffentlichen Schlüssel generieren und im SSL-Schlüsselspeicher ablegen. (Weitere Informationen hierzu finden Sie im Abschnitt "Paar aus öffentlichem und privatem Schlüssel sowie selbst signiertes Zertifikat generieren"). Dieser Schritt wird nur dann benötigt, wenn Ihre Seite der SSL-Verbindung ihre Authentizität gegenüber ihren Partnern nachweisen muss, wenn Sie sich also auf der SSL-Serverseite oder aber auf der SSL-Clientseite befinden und eine Clientauthentifizierung erforderlich ist.
2. [Optional] Fordern Sie von einer Zertifizierungsstelle ein Zertifikat an und ersetzen Sie Ihr selbst signiertes Zertifikat durch dieses Zertifikat (siehe "Zertifikat für öffentlichen Schlüssel in einen Schlüsselspeicher importieren").
3. [Optional] Exportieren Sie das Zertifikat für den öffentlichen Schlüssel Ihres privaten Schlüssels und verteilen Sie es an die SSL-Parteien, die mit Ihnen interagieren (siehe "Zertifikat für öffentlichen Schlüssel aus einem Schlüsselspei-

cher exportieren"). Falls Sie ein Zertifikat von einer Zertifizierungsstelle verwenden, müssen andere lediglich über das Zertifikat der Zertifizierungsstelle selbst verfügen.

4. Importieren Sie Zertifikate von vertrauenswürdigen Parteien in Ihren SSL-Truststore (siehe "Zertifikat für öffentlichen Schlüssel in einen Schlüsselspeicher importieren"). Dieser Schritt ist für SSL-Clients obligatorisch. Bei SSL-Servern muss er nur dann ausgeführt werden, wenn die Clientauthentifizierung erforderlich ist.

Ergebnisse

Anmerkung: Falls Sie SSL mit den Standardeigenschaften konfigurieren (jvax-net.ssl.*), muss der SSL-Schlüsselspeicher genau 1 privaten Schlüssel enthalten, da nicht angegeben werden kann, welcher Schlüssel verwendet wird.

Schlüssel für die Verschlüsselung

Nachstehend erhalten Sie Informationen zur Verschlüsselung, zu den dafür verwendeten Schlüsseln und zu den entsprechenden Algorithmen.

Bei der Verschlüsselung gibt es zwei Alternativen:

- Verwendung eines Paares aus öffentlichem und privatem Schlüssel
- Verwendung eines geheimen Schlüssels

Der gängigste Algorithmus für die Verschlüsselung öffentlicher Schlüssel ist RSA (Rivest-Shamir-Adleman-Algorithmus). Bitte beachten Sie, dass andere vielfach eingesetzte Algorithmen mit öffentlichem Schlüssel wie DiffieHellman (Schlüsselaustausch) und DSA (digitale Signatur) nicht zur Verschlüsselung verwendet werden können.

Die Verschlüsselung mit geheimen Schlüsseln ist generell schneller und sicherer als die Verschlüsselung mit öffentlichen Schlüsseln. Der Directory Integrator-Server verwendet jedoch standardmäßig die Verschlüsselung mit öffentlichen Schlüsseln nach RSA, um die Abwärtskompatibilität zu gewährleisten.

Tools

Sie können die Dienstprogramme "keytool" und "Ikeyman" für die Arbeit mit Tasten und Zertifikaten verwenden. Nachstehend erhalten Sie Informationen zu diesen Tools.

IBM JVM bietet zwei Dienstprogramme für die Arbeit mit Schlüssel und Zertifikaten: keytool und Ikeyman. Das Befehlszeilendienstprogramm keytool wird in der Java-Benutzergemeinschaft vielfach eingesetzt. Das GUI-Tool Ikeyman von IBM bietet viele Funktionen, die auch von "keytool" bereitgestellt werden. Beide Tools befinden sich im Ordner *tdi-installationsverzeichnis/jvm/jre/bin*. Ausführliche Informationen zu diesen Tools enthält die Dokumentation zu IBM JVM, die Sie unter der Adresse <http://www.ibm.com/developerworks/java/jdk/security/> einsehen können.

Mit dem Produkt werden die folgenden Standardschlüsselspeicher ausgeliefert:

Tabelle 14. IBM Security Directory Integrator-Schlüsselspeicher

Schlüsselspeicherposition	Schlüsselspeicherkeywort	Anerkannte öffentliche Schlüssel	Private Schlüssel
<i>tdi-installationsverzeichnis/testserver.jks</i>	server	admin	server
<i>tdi-installationsverzeichnis/serverapi/testadmin.jks</i>	administrator	server	admin

Inhalt eines Schlüsselspeichers auflisten

Mit dem Befehl `list` von `keytool` können Sie den Inhalt eines Schlüsselspeichers auflisten.

Der folgende Befehl listet beispielsweise Informationen zu den Schlüsseln (Aliasname und Typ) auf, die sich in der Schlüsselspeicherdatei `mystore.jck` befinden (das Format des Schlüsselspeichers ist JCEKS, das Kennwort lautet "mystorepass"):

```
keytool -list -storetype jceks -keystore mystore.jck -storepass mystorepass
```

Schlüssel erstellen

Nachstehend erhalten Sie Informationen zum Erstellen von Schlüsseln, zur Verwaltung von Schlüsseln in einem Schlüsselspeicher sowie zu deren Verwendung.

Paar aus öffentlichem und privatem Schlüssel sowie selbst signiertes Zertifikat generieren

Der folgende `keytool`-Befehl generiert beispielsweise ein Paar aus öffentlichem und privatem RSA-Schlüssel mit dem Aliasnamen "myserverkey" sowie ein selbst signiertes X.509-Zertifikat für öffentlichen Schlüssel:

```
keytool -genkeypair -alias myserverkey -dname cn=myserver.mydomain.com -validity 365 -keyalg RSA -keysize 1024 -keypass mykeypass -storetype jceks -keystore mystore.jck -storepass mystorepass
```

Der definierte Name des Zertifikateigners ist "cn=myserver.mydomain.com". Er sollte mit dem DNS-Namen des Servers identisch sein, der das selbst signierte Zertifikat für SSL verwendet (bei der Verschlüsselung mit öffentlichem Schlüssel ist der Inhalt des Zertifikats nicht von besonderer Bedeutung). Das Zertifikat ist 365 Tage gültig. Die Größe des generierten RSA-Schlüssels beträgt 1.024 Byte. Das Kennwort des privaten Schlüssels lautet "mykeypass". Das Schlüsselpaar wird in der Schlüsselspeicherdatei `mystore.jck` mit dem Format JCEKS gespeichert (falls die Datei nicht vorhanden ist, wird sie erstellt). Das Kennwort des Schlüsselspeichers ist "mystorepass".

Der Schlüsselspeicher `mystore.jck` kann als SSL-Schlüsselspeicher eines Serverprogramms verwendet werden, das auf dem Host "myserver.mydomain.com" ausgeführt wird. Der Schlüsselspeicher enthält außerdem für den privaten Schlüssel ein Zertifikat für den öffentlichen Schlüssel und kann daher für Clients, die eine Verbindung zum Server auf "myserver.mydomain.com" herstellen, als SSL-Truststore verwendet werden. (Die Weitergabe Ihres privaten Schlüssels an Clients ist allerdings vollkommen unnötig und generell eine fragwürdige Sicherheitspraxis.)

Zertifikat von einer Zertifizierungsstelle anfordern

Normalerweise vollzieht sich der Prozess für die Anforderung und Verwendung von Zertifikaten, die durch eine Zertifizierungsstelle signiert wurden, folgendermaßen:

Zunächst werden ein Schlüsselpaar und ein selbst signiertes Zertifikat generiert (siehe "Paar aus öffentlichem und privatem Schlüssel sowie selbst signiertes Zertifikat generieren"). Anschließend wird bei der Zertifizierungsstelle ein Zertifikat für den öffentlichen Schlüssel angefordert. Nachdem die Zertifizierungsstelle das signierte Zertifikat zurückgesendet hat, wird es in den entsprechenden Truststore importiert und ersetzt dort das selbst signierte Zertifikat.

Beispielsweise können Sie mit `keytool` folgendermaßen eine Zertifikatssignieranforderung für den Schlüssel "myserverkey" aus dem Schlüsselspeicher `mystore.jck` generieren:

```
keytool -certreq -file myreq.csr -alias myserverkey -keypass mykeypass -storetype jceks
-keystore mystore.jck -storepass mystorepass
```

Dieser Befehl erstellt eine Zertifikatssignieranforderung in der Datei `myRequest.csr` für den öffentlichen Schlüssel mit dem Aliasnamen "myserverkey". Die erstellte Zertifikatssignieranforderung kann dann an eine Zertifizierungsstelle gesendet werden. Sobald das neue Zertifikat eintrifft, können Sie es in den Schlüsselspeicher importieren (siehe "Zertifikat für öffentlichen Schlüssel in einen Schlüsselspeicher importieren"). Der folgende `keytool`-Befehl generiert einen 256-Bit-AES-Schlüssel mit dem Aliasnamen "myseckey":

```
keytool -genseckey -keyalg AES -alias myseckey -keysize 256 -keypass mykeypass -storetype jceks
-keystore mystore.jck -storepass mystorepass
```

Der neue Schlüssel wird in einer JCEKS-Schlüsselspeicherdatei `mystore.jck` mit dem Kennwort "mystorepass" gespeichert. Das Kennwort, das den geheimen Schlüssel schützt, lautet "mykeypass".

Schlüssel aus einem Schlüsselspeicher in einen anderen Schlüsselspeicher kopieren Sie können beispielsweise das im Abschnitt "Paar aus öffentlichem und privatem Schlüssel sowie selbst signiertes Zertifikat generieren" erstellte Schlüsselpaar mit dem folgenden `keytool`-Befehl kopieren:

```
keytool -importkeystore -srckeystore mystore.jck -destkeystore myotherstore.jks
-srcstoretype jceks
-deststoretype jks -srcstorepass mystorepass -deststorepass myotherstorepass
-srcalias myserverkey
-destalias myotherserverkey -srckeypass mykeypass -destkeypass myotherkeypass
```

Die Kopie wird unter dem Aliasnamen "myotherserverkey" in der JKS-Schlüsselspeicherdatei `myotherstore.jks` gespeichert (wenn die Datei nicht vorhanden ist, wird sie erstellt).

Schlüsselspeicher aus einem Format in ein anderes Format konvertieren

Sie können beispielsweise den im Abschnitt "Paar aus öffentlichem und privatem Schlüssel sowie selbst signiertes Zertifikat generieren" erstellten JCEKS-Schlüsselspeicher mit dem folgenden `keytool`-Befehl in einen JKS-Schlüsselspeicher namens `myotherstore.jks` konvertieren:

```
keytool -importkeystore -srckeystore mystore.jck -destkeystore
myotherstore.jks -srcstoretype jcek
-deststoretype jks -srcstorepass mystorepass -deststorepass
myotherstorepass
```

Der Befehl fordert später dann für jeden einzelnen privaten oder geheimen Schlüssel im Quellenschlüsselspeicher das Kennwort an. Bitte beachten Sie, dass geheime Schlüssel in JKS- und PKCS#12-Schlüsselspeichern nicht gespeichert werden können. Sie sollten nicht versuchen, einen Schlüsselspeicher, der geheime Schlüssel enthält, in JKS oder PKCS#12 zu konvertieren.

Zertifikat für öffentlichen Schlüssel aus einem Schlüsselspeicher exportieren

Mit dem folgenden Befehl wird das im Abschnitt "Paar aus öffentlichem und privatem Schlüssel sowie selbst signiertes Zertifikat generieren" erstellte Zertifikat für öffentlichen Schlüssel in eine Binärdatei namens `myserverkey.der` exportiert:

```
keytool -exportcert -alias myserverkey -file myserverkey.der
-storetype JCEKS -keystore mystore.jck
-storepass mystorepass
```

Die resultierende Datei ".der" enthält die DER-Verschlüsselung des X.509-Zertifikats. Es handelt sich um eine Binärdatei. Um dieselben Binärdaten im Textformat (mit Base64-Codierung der DER-Verschlüsselung für das X.509-Zertifikat) abzurufen, verwenden Sie die Option "-rfc" von `keytool`:

```
keytool -exportcert -alias myserverkey -file myserverkey.arm
-storetype JCEKS -keystore mystore.jck
-storepass mystorepass -rfc
```

Zertifikat für öffentlichen Schlüssel in einen Schlüsselspeicher importieren

Mit dem folgenden Befehl können Sie ein neues gesichertes Zertifikat in einen Schlüsselspeicher importieren:

```
keytool -importcert -alias myserverkey -file myserverkey.der
-storetype JCEKS -keystore mystore.jck
-storepass mystorepass
```

Das Dienstprogramm `keytool` versucht, den Unterzeichner des Zertifikats zu prüfen, das Sie importieren wollen. Zu diesem Zweck wird eine Zertifizierungskette vom importierten Zertifikat zu einem anderen gesicherten Zertifikat erstellt. Falls eine solche Kette nicht aufgebaut werden kann, gibt `keytool` an Sie die Frage aus, ob Sie sicher sind, dass das Zertifikat importiert werden muss.

Verwenden Sie zum Importieren eines Zertifikats, das von einer Zertifizierungsstelle als Antwort auf eine Zertifikatssignieranforderung gesendet wurde (in diesem Fall ist im Schlüsselspeicher für dieses Zertifikat bereits ein privater Schlüssel vorhanden) einen Befehl wie den Folgenden:

```
keytool -importcert -alias myserverkey -keypass mykeypass -file
myserverkey.der -storetype JCEKS -keystore mystore.jck
-storepass mystorepass
```

Beachten Sie, dass Sie beim Importieren eines Zertifikats für einen vorhandenen privaten Schlüssel das Kennwort des privaten Schlüssels angeben müssen. Das Dienstprogramm `keytool` versucht, den Unterzeichner des Zertifikats zu prüfen, indem eine Zertifizierungskette zu einem gesicherten Zertifikat aufgebaut wird. Falls eine solche Kette nicht aufgebaut werden kann, schlägt der Import fehl und Sie werden nicht aufgefordert, die Authentizität des Zertifikats zu bestätigen. Damit eine Antwort auf eine Zertifikatssignieranforderung erfolgreich importiert werden kann, muss die Zertifizierungsstelle, die das Zertifikat ausgegeben hat, vertrauenswürdig sein. Falls Sie eine der gängigen Zertifizierungsstellen verwenden (z. B. VeriSign oder Thawte), können Sie sich ohne Weiteres auf die Zertifikate im Standardtruststore der JVM (`java.home/lib/security/cacerts`) stützen (hierzu verwenden Sie die Option `-trustcacerts` von `keytool`):

```
keytool -importcert -alias myserverkey -keypass mykeypass -file
myserverkey.der -storetype JCEKS -keystore mystore.jck
-storepass mystorepass -trustcacerts
```

Gültigkeit eines Zertifikats mit "keytool" verlängern

Angenommen, Sie besitzen einen JCEKS-Schlüsselspeicher namens `mystore.jck`, der ein abgelaufenes (oder fast abgelaufenes) selbst signiertes Zertifikat mit dem Aliasnamen `myserverkey` enthält. Der zugeordnete private Schlüssel ist im Schlüsselspeicher vorhanden. Das Kennwort für den Schlüsselspeicher lautet `mystorepass`; das Kennwort für den privaten Schlüssel ist `mykeypass`. Wenn Sie nun die Gültigkeit dieses Zertifikats um weitere 365 Tage verlängern wollen, können Sie den folgenden Befehl mit `keytool` ausführen:

```
keytool -selfcert -v -alias myserverkey -keypass mykeypass -validity 365
-storetype jceks -keystore mystore.jck
-storepass mystorepass
```

Die obige Operation generiert ein neues selbst signiertes Zertifikat mit demselben definierten Namen, Signieralgorithmus und denselben Schlüsseln wie das ursprüngliche Zertifikat, jedoch einer neuen Seriennummer und einem neuen Gültigkeitszeitraum.

Anmerkung: Das ursprüngliche Zertifikat wird automatisch durch das generierte neue Zertifikat ersetzt.

Wenn Sie das ursprüngliche Zertifikat zur Referenz oder aus einem anderen Grund zu einem späteren Zeitpunkt benötigen, müssen Sie daher eine Kopie des ursprünglichen Schlüsselspeichers aufbewahren, bevor Sie die oben erläuterte Verlängerung des Zertifikats vornehmen.

Beachten Sie, dass dieses Verfahren nur bei selbst signierten Zertifikaten angewendet werden kann. Es generiert nämlich in Wirklichkeit ein neues selbst signiertes Zertifikat für den öffentlichen Schlüssel, weshalb Sie es exportieren und die Truststores der SSL-Parteien, mit denen Sie kommunizieren wollen, aktualisieren müssen.

Mit in PFX/PKCS#12-Dateien gespeicherten Schlüsseln arbeiten

Soweit es Java betrifft, ist PKCS#12 lediglich ein weiterer Schlüsselspeichertyp (wie JCEKS und JKS). Um mit PKCS#12-Schlüsselspeichern zu arbeiten, setzen Sie die Option "-storetype" von keytool einfach auf "pkcs12". Der folgende Befehl listet beispielsweise den Inhalt einer PKCS#12-Datei namens mystore.p12 mit dem Kennwort "mystorepass" auf:

```
keytool -list -storetype pkcs12 -keystore mystore.p12 -storepass mystorepass
```

Schlüsselspeicherdatei erstellen

Wenn Sie Schlüsselspeicherdateien verwenden wollen, müssen Sie diese nicht gesondert erstellen. Das Dienstprogramm keytool erstellt automatisch eine neue Schlüsselspeicherdatei, wenn es Daten in eine nicht vorhandene Datei schreiben muss. Falls Sie beispielsweise in einem nicht vorhandenen Schlüsselspeicher einen neuen Schlüssel generieren oder ein Zertifikat importieren wollen, erstellt keytool zuerst die Schlüsselspeicherdatei.

Dienstprogramm "keytool" im FIPS-Modus ausführen

Um das Dienstprogramm keytool im FIPS-konformen Modus auszuführen, verwenden Sie in jeder Befehlszeile die Option "-providerClass". Beispiel:

```
keytool -list -storetype JCEKS -keystore mystore.jck -storepass mystorepass  
-providerClass com.ibm.crypto.fips.provider.IBMJCEFIPS
```

Unterstützung für Secure Sockets Layer (SSL)

Sie können den Datenaustausch im Netz mithilfe von SSL-Sicherheitseinrichtungen verschlüsseln und authentifizieren. Außerdem können Sie eine Liste der unterstützten Connectors anzeigen und erhalten Informationen zur Konfiguration.

SSL ist für viele IBM Security Directory Integrator-Sicherheitseinrichtungen eine wichtige Grundlage. Um das in IBM Security Directory Integrator verfügbare Leistungsspektrum voll ausschöpfen zu können, müssen Sie praktische Kenntnisse über SSL besitzen.

Die folgenden Connectors unterstützen SSL bei ordnungsgemäß konfigurierten IBM Security Directory Integrator-Servern:

- Connectors
 - AD-Änderungserkennungsconnector
 - Server-Connector für Axis Easy-Web-Service
 - Server-Connector für Axis2-Web-Service
 - Domino-Änderungserkennungsconnector
 - Domino-Benutzerconnector
 - DSML v2-SOAP-Server-Connector
 - FTP-Client-Connector

- HTTP-Server-Connector
- IDS-Änderungsprotokollconnector
- IBM MQ Series-Connector
- JMS-Connector
- Connector für JMS-Kennwortspeicher
- JNDI-Connector
- LDAP-Connector
- LDAP-Gruppen-Connector
- LDAP-Server-Connector
- Lotus Notes-Connector
- Mailbox-Connector
- Änderungserkennungsconnector für Sun Directory
- LDAP-Connector
- TADDM-Änderungserkennungsconnector
- TADDM-Connector
- TCP-Connector
- TCP-Server-Connector
- TPAE-IF-Änderungserkennungsconnector
- Server-Connector für Web-Service-Empfänger
- Änderungsprotokollconnector für z/OS-LDAP

SSL ermöglicht die Verschlüsselung und Authentifizierung des Datenaustausches im Netz zwischen zwei fernen kommunizierenden Partnern. Die meisten IBM Security Directory Integrator-Implementierungen in der Produktionsumgebung nutzen SSL. Aus diesem Grund ist die SSL-Unterstützung eine der wichtigsten Sicherheitseinrichtungen von IBM Security Directory Integrator. Weitere Informationen zu SSL sowie Angaben über die Verwendung von SSL in Java-Programmen aus der Entwicklerperspektive finden Sie unter der Adresse <http://download.oracle.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html>.

IBM Security Directory Integrator kann gleichzeitig als Client und/oder als Server verwendet werden. Die SSL-Konfiguration von IBM Security Directory Integrator für die Verwendung als Client unterscheidet sich von der SSL-Konfiguration von IBM Security Directory Integrator für die Verwendung als Server. Aus diesem Grund ist dieser Abschnitt in zwei Teile untergliedert: „SSL-Konfiguration von IBM Security Directory Integrator-Komponenten als Server“ und „SSL-Konfiguration von IBM Security Directory Integrator-Komponenten als Client“ auf Seite 113.

SSL-Konfiguration von IBM Security Directory Integrator-Komponenten als Server

Sie müssen einen Schlüsselspeicher definieren, um die SSL-Unterstützung für IBM Security Directory Integrator als Server zu aktivieren. Mit den hier aufgeführten Schritten können Sie diese Task ausführen.

Informationen zu diesem Vorgang

Wenn eine IBM Security Directory Server-Komponente (z. B. ein Connector im Modus "Server") als Server verwendet wird, ist es für SSL erforderlich, dass ein von IBM Security Directory Integrator zu verwendender Schlüsselspeicher definiert werden muss. Informationen zu Schlüsselspeichern und Truststores enthält die Do-

kumentation, die unter der Adresse <http://download.oracle.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html> verfügbar ist. Die folgenden Schritte müssen ausgeführt werden, um die SSL-Unterstützung für IBM Security Directory Integrator als Server zu aktivieren:

Anmerkung: RMI ist im IBM Security Directory Integrator-Server standardmäßig aktiviert. Die Eigenschaften für die Serverauthentifizierung enthalten die Eigenschaftswerte für den Standardschlüsselspeicher.

1. Falls noch keine Java-Schlüsselspeicherdatei (JKS) in IBM Security Directory Integrator vorhanden ist, erstellen Sie mit dem Dienstprogramm *keytool* eine Schlüsselspeicherdatei (das Dienstprogramm befindet sich je nach Plattform an der Position *tdi-installationsverzeichnis/jvm/jre/bin* oder *tdi-installationsverzeichnis/jvm/bin*). Ist kein persönlicher Schlüssel für die Verwendung in IBM Security Directory Integrator vorhanden, fordern Sie einen solchen Schlüssel von einer Zertifizierungsstelle an oder erstellen Sie einen selbst signierten Schlüssel.
2. Falls es sich bei dem Zertifikat in IBM Security Directory Integrator um ein selbst signiertes Zertifikat handelt, exportieren Sie das Zertifikat.
3. Ist das IBM Security Directory Integrator-Zertifikat ein selbst signiertes Zertifikat, importieren Sie das exportierte IBM Security Directory Integrator-Zertifikat mit einem Schlüsseltool als Rootberechtigungs-zertifikat in die Schlüsselspeicherdatei des Clients.
4. Bearbeiten Sie in der Datei *tdi-installationsverzeichnis/etc/global.properties* die Position der Schlüsselspeicherdatei, das Kennwort der Schlüsselspeicherdatei und den Typ der Schlüsselspeicherdatei.

```
## client authentication javax.net.ssl.keyStore=serverapi\testadmin.jks {protect}-javax.net.ssl.keyStorePassword=administrator javax.net.ssl.keyStoreType=jks
```
5. Aktivieren Sie SSL für die Clients (beispielsweise durch Verwendung von "https" im Web-Browser).
6. Starten Sie IBM Security Directory Integrator erneut.

Anmerkung:

1. Der IBM Security Directory Integrator-Server verwaltet die Schlüsselspeicher/Truststores nicht. In Bezug auf die Schlüsselspeicherunterstützung stellt der IBM Security Directory Integrator-Server für die IBM Security Directory Integrator-Komponenten lediglich die Dateien *global.properties* oder *solution.properties* bereit, in denen die Java-Standard-eigenschaften für den Schlüsselspeicher oder Truststore angegeben werden können.
2. Eine IBM Security Directory Integrator-Komponente kann den konfigurierten Standardschlüsselspeicher/-truststore in der Datei *global.properties* oder *solution.properties* verwenden oder alternativ eine eigene Handhabung von SSL-Sockets implementieren (z. B. durch Implementierung einer angepassten Java-Klasse "SSLServerSocket"), damit von der Standardeinstellung abweichende Schlüsselspeicher/Truststores verwendet werden können.
3. Falls IBM Security Directory Integrator sowohl ein Client- als auch ein Serverzertifikat verwenden muss und nur das in der Datei *global.properties* oder *solution.properties* konfigurierte Standardzertifikat verwendet wird, muss es identisch sein. Alternativ kann eine angepasste Implementierung der Java-Klasse "SSLServerSocket" oder "SSLServerSocket" geschrieben werden, die ein vom Standard abweichendes Zertifikat verwendet.

4. Der Abschnitt „Zertifikate für Web-Service-Suite von IBM Security Directory Integrator“ auf Seite 167 enthält spezielle Informationen über die Zertifikate für IBM Security Directory Integrator-Web-Service-Komponenten.

SSL-Konfiguration von IBM Security Directory Integrator-Komponenten als Client

Sie müssen einen Truststore definieren, um die SSL-Unterstützung für IBM Security Directory Integrator als Client zu aktivieren. Mit den hier aufgeführten Schritten können Sie diese Task ausführen.

Informationen zu diesem Vorgang

Wenn eine IBM Security Directory Integrator-Komponente (z. B. der LDAP-Connector) als Client verwendet wird, ist es für SSL erforderlich, dass ein von IBM Security Directory Integrator zu verwendender Truststore definiert werden muss. Informationen zu Schlüsselspeichern und Truststores enthält die Dokumentation, die unter der Adresse <http://download.oracle.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html> verfügbar ist.

Die folgenden Schritte müssen ausgeführt werden, um die SSL-Unterstützung für IBM Security Directory Integrator als Client zu aktivieren:

1. Konfigurieren Sie einen Server (z. B. IBM Security Directory Integrator), um SSL zu aktivieren.
2. Falls es sich bei dem Zertifikat im Server um ein selbst signiertes Zertifikat handelt, exportieren Sie das Zertifikat.
3. Falls noch keine Java-Schlüsselspeicherdatei (jks) vorhanden ist, erstellen Sie mit *keytool* für IBM Security Directory Integrator eine Schlüsselspeicherdatei (das Dienstprogramm befindet sich je nach Plattform an der Position *stammverzeichnis/jvm/jre/bin* oder *stammverzeichnis/jvm/bin*).
4. Falls es sich bei dem Serverzertifikat um ein selbst signiertes Zertifikat handelt, importieren Sie das Serverzertifikat mit dem Dienstprogramm "keytool" als Rootberechtigungszertifikat in die IBM Security Directory Integrator-Schlüsselspeicherdatei.
5. Bearbeiten Sie in der Datei *stammverzeichnis/etc/global.properties* die Position der Schlüsselspeicherdatei, das Kennwort der Schlüsselspeicherdatei und den Typ der Schlüsselspeicherdatei.

Anmerkung: Die folgenden vier Zeilen (mit dem Zeichen # beginnende Kommentare) werden für die Client- und Serverauthentifizierung beim IBM Security Directory Integrator-Server nicht mehr benötigt. Nach der Standardkonfiguration werden die zu IBM Security Directory Integrator gehörenden Speicher verwendet. Dies ist standardmäßig Bestandteil der Aktivierung von RMI (Remote Method Invocation).

```
# Keystore file information for the server TDI authentication.  
# It is used to provide the public key of the TDI to the SSL enabled client.  
# javax.net.ssl.keyStore=D:\test\clientStore.jks  
# javax.net.ssl.keyStorePassword=secret  
# javax.net.ssl.keyStoreType=jks
```

6. Aktivieren Sie SSL für die Connectors.
7. Starten Sie IBM Security Directory Integrator erneut.

Anmerkung: Der IBM Security Directory Integrator-Truststore und -Schlüsselspeicher spielen in der SSL-Konfiguration des Domino-Änderungserkennungsconnectors keine Rolle. Weitere Informationen finden Sie im Abschnitt „Spezielle Aspekte für SSL bei Lotus Domino“ auf Seite 166.

SSL-Clientauthentifizierung

Falls Sie eine IBM Security Directory Integrator-Komponente als Client und einen Server verwenden möchten, ist für SSL die Verwendung eines Schlüsselspeichers und eines Truststores erforderlich. Nachstehend erhalten Sie Informationen dazu.

Falls eine IBM Security Directory Integrator-Komponente als Client verwendet wird und der Server, mit dem sie kommuniziert, die SSL-Clientauthentifizierung erfordert, benötigt IBM Security Directory Integrator neben einem Truststore auch einen Schlüsselspeicher. In diesem Fall kann der Schlüsselspeicher genauso wie bei der Verwendung von IBM Security Directory Integrator als Server definiert werden (siehe „SSL-Konfiguration von IBM Security Directory Integrator-Komponenten als Server“ auf Seite 111).

Anmerkung: IBM Security Directory Integrator-Clientkomponenten, die die SSL-Clientauthentifizierung unterstützen, benötigen - anders als IBM Security Directory Integrator-Serverkomponenten - normalerweise kein Markierungsfeld für die SSL-Clientauthentifizierung. Um die eigene Identität gegenüber dem Server nachzuweisen, müssen solche IBM Security Directory Integrator-Clientkomponenten lediglich einen Schlüsselspeicher generiert und in der Datei `global.properties` oder `solution.properties` konfiguriert haben. Falls der Server ein SSL-Clientzertifikat erfordert, sendet die SSL-Bibliothek des Clients dann automatisch das Zertifikat des Clients aus dem Schlüsselspeicher, der in der Datei `global.properties` oder `solution.properties` konfiguriert ist.

SSL-Konfiguration für IBM Security Directory Integrator und Microsoft Active Directory

Führen Sie die hier aufgeführten Schritte aus, um SSL für IBM Security Directory Integrator und Microsoft Active Directory zu konfigurieren.

Informationen zu diesem Vorgang

1. Installieren Sie Zertifikatsservices auf dem Windows-Server und eine Enterprise Certificate Authority (Zertifizierungsstelle im Unternehmen) in der Active Directory-Domäne. Details sind unter der Adresse <http://windowsitpro.com/article/articleid/14923/how-do-i-install-an-enterprise-certificate-authority.html> verfügbar. Achten Sie darauf, eine **Enterprise Certificate Authority** zu installieren.
2. Starten Sie den Service für den Zertifikatserver (Certificate Server Service). Dies erstellt in Internet Information Service (IIS) ein virtuelles Verzeichnis, über das Sie Zertifikate verteilen können.
3. Erstellen Sie eine Richtlinie des Typs "Security (Group)", um Domänencontroller anzuweisen, ein SSL-Zertifikat von der Zertifizierungsstelle anzufordern.
 - a. Öffnen Sie das Tool **Active Directory Users and Computers Administrative**.
 - b. Klicken Sie im Bereich **Domain Controllers** mit der rechten Maustaste.
 - c. Wählen Sie die Option **Properties** aus.
 - d. Wählen Sie die Registerkarte **Group Policy** aus und klicken Sie, um das Element **Default Domain Controllers Policy** zu bearbeiten.
 - e. Navigieren Sie zu **Computer Configuration->Windows Settings->Security Settings->Public Key Policies**.
 - f. Klicken Sie mit der rechten Maustaste auf **Automatic Certificate Request Settings**.
 - g. Wählen Sie die Option **New** aus.
 - h. Wählen Sie die Option **Automatic Certificate Request** aus.

- i. Führen Sie den Assistenten aus. Wählen Sie die Option **Certificate Template for a Domain Controller** aus.
- j. Wählen Sie als Zertifizierungsstelle Ihre **Enterprise Certificate Authority** aus. Die Auswahl der Zertifizierungsstelle eines anderen Anbieters ist ebenfalls möglich.
- k. Arbeiten Sie den Assistenten vollständig durch.

Anmerkung: Alle Domänencontroller fordern automatisch ein Zertifikat von der Zertifizierungsstelle an und unterstützen LDAP mit SSL an Port 636.

- 4. Rufen Sie das Zertifikat einer Zertifizierungsstelle an dem Computer ab, auf dem Sie IBM Security Directory Integrator installiert haben.

Anmerkung: Vor der Installation des Zertifikatservers müssen Sie IIS installieren.

- a. Öffnen Sie auf dem Computer, auf dem Sie IBM Security Directory Integrator installiert haben, einen Web-Browser.
- b. Rufen Sie die Adresse "http://*servername*/certsrv/" auf (hierbei steht *servername* für den Namen des Windows 2000-Servers). Sie werden aufgefordert, sich anzumelden.
- c. Wählen Sie die Task **Retrieve the CA certificate or certificate revocation list** aus.
- d. Klicken Sie auf **Weiter**.
Auf der nächsten Seite ist das Zertifikat einer Zertifizierungsstelle automatisch hervorgehoben.
- e. Klicken Sie auf **Download CA certificate**
Daraufhin wird ein neues Fenster für den Download geöffnet.
- f. Speichern Sie die Datei auf der Festplatte.
- 5. Erstellen Sie mit "keytool" einen Zertifikatsspeicher. Verwenden Sie das Dienstprogramm "keytool.exe", um den Zertifikatsspeicher zu erstellen und das Zertifikat einer Zertifizierungsstelle in diesen Speicher zu importieren.

Anmerkung: Die Datei "keytool.exe" befindet sich je nach Plattform im Verzeichnis "*stammverzeichnis*/jvm/jre/bin" oder "*stammverzeichnis*/jvm/bin".

Verwenden Sie den folgenden Befehl:

```
jvm\jre\bin\keytool -import -file
certnew.cer -keystore schlüssel-speichersname.jks
-storepass kennwort-alias schlüsselaliasname
```

Im folgenden Beispiel werden diese Werte verwendet:

```
schlüssel-speichersname = idi.jks
kennwort = secret
schlüsselaliasname = AD_CA
```

Der Befehl lautet wie das folgende Script:

```
C:\Program Files\IBM\TDI\V7.2\jvm\jre\bin\keytool -import
-file certnew.cer -keystore idi.jks -storepass secret -alias AD_CA
```

Geben Sie das folgende Script ein, um den Inhalt Ihres SchlüsselSpeichers zu prüfen:

```
C:\Program Files\IBM\TDI\V7.2\jvm\jre\bin\keytool
-list -keystore idi.jks -storepass secret
```

Die folgenden Zeilen werden ausgegeben:

Keystore type: jks
Keystore provider: SUN

Your keystore contains 1 entry:

ad_ca, Mon Nov 04 22:11:46 MST 2002, trustedCertEntry,
Certificate fingerprint (MD5): A0:2D:0E:4A:68:34:7F:A0:21:36:78:65:A7:1B:25:55

6. Konfigurieren Sie IBM Security Directory Integrator für die Verwendung des im vorherigen Schritt erstellten Schlüsselspeichers. Bearbeiten Sie in der Datei `stammverzeichnis/global.properties` die Position der Schlüsselspeicherdatei, das Kennwort der Schlüsselspeicherdatei und den Typ der Schlüsselspeicherdatei. Im aktuellen Release wird ausschließlich der Typ "jks" unterstützt.

```
#server authentication
#example
javax.net.ssl.trustStore=c:\test\idi.jks
javax.net.ssl.trustStorePassword=secret
javax.net.ssl.trustStoreType=jks
#client authentication
#example
javax.net.ssl.keyStore=c:\test\idi.jks
javax.net.ssl.keyStorePassword=secret
javax.net.ssl.keyStoreType=jks
```

7. Aktivieren Sie SSL für Ihren LDAP-Connector.
 - a. Wechseln Sie in das Fenster für die Konfiguration des LDAP-Connectors.
 - b. Ändern Sie die **LDAP-URL** in Port 636.
 - c. Wählen Sie das Markierungsfeld **SSL verwenden** aus.
8. Starten Sie IBM Security Directory Integrator erneut.

Anmerkung: Der IBM Security Directory Integrator-Wrapper für Windows-Dienste ermöglicht es Ihnen, IBM Security Directory Integrator mit mehreren Serviceinstanzen zu starten.

Zusammenfassung der Eigenschaften für die SSL-Aktivierung und die PKCS#11-Unterstützung

Nachstehend finden Sie einen Link sowie Informationen zum Konfigurieren der SSL-Eigenschaften.

Sie können SSL-Eigenschaften für die Serverauthentifizierung, für die Clientauthentifizierung und für die PKCS#11-Unterstützung konfigurieren. Informationen zu PKCS (Public Key Cryptography Standards) finden Sie unter „Verschlüsselungsschlüssel auf Hardwareeinheiten verwenden“ auf Seite 206.

Tabelle 15. SSL-Serverauthentifizierung

Eigenschaft	Standardwert	Beschreibung
<code>javax.net.ssl.trustStore</code>	<code>serverapi\testadmin.jks</code>	Position der Truststore-Dateien
<code>{protect}-javax.net.ssl.trustStorePassword</code>	administrator (standardmäßig verschlüsselt)	Truststore-Kennwort
<code>javax.net.ssl.trustStoreType</code>	jks	Truststore-Typ

Tabelle 16. SSL-Clientauthentifizierung

Eigenschaft	Standardwert	Beschreibung
<code>javax.net.ssl.keyStore</code>	<code>serverapi\testadmin.jks</code>	Position der Schlüsselspeicherdateien
<code>{protect}-javax.net.ssl.keyStorePassword</code>	administrator (standardmäßig verschlüsselt)	Schlüsselspeicher kennwort

Table 16. SSL-Clientauthentifizierung (Forts.)

Eigenschaft	Standardwert	Beschreibung
javax.net.ssl.keyStoreType	jks	Schlüsselspeichertyp

Table 17. PKCS#11-Unterstützung

Eigenschaft	Standardwert	Beschreibung
com.ibm.di.pkcs11cfg	etc\pkcs11.cfg	Verwenden Sie diese Eigenschaft, um den Pfad der Konfigurationsdatei anzugeben, die zur Initialisierung des IBM PKCS11-Implementierungsproviders benötigt wird. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.server.pkcs11	false	Verwenden Sie mit PKCS11 kompatible Verschlüsselungseinheiten für SSL. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.server.pkcs11.library	Keiner	Geben Sie den Pfad zur PKCS11-Clientbibliothek an. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.server.pkcs11.slot	Keiner	Geben Sie die Steckplatznummer der Einheit an.
{protect}- com.ibm.di.server.pkcs11.pass	Keiner	Der Zugriff auf die mit PKCS11 kompatible Verschlüsselungseinheit erfolgt unter Verwendung dieses Kennworts. Es ist standardmäßig verschlüsselt. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.

Beispiel für SSL

Anhand von zwei Beispielen wird im Folgenden veranschaulicht, wie IBM Security Directory Integrator bei Verwendung als Server und auch bei Verwendung als Client für SSL konfiguriert werden kann. Im ersten Beispiel wird der LDAP-Server-Connector implementiert, im anderen Beispiel der LDAP-Connector.

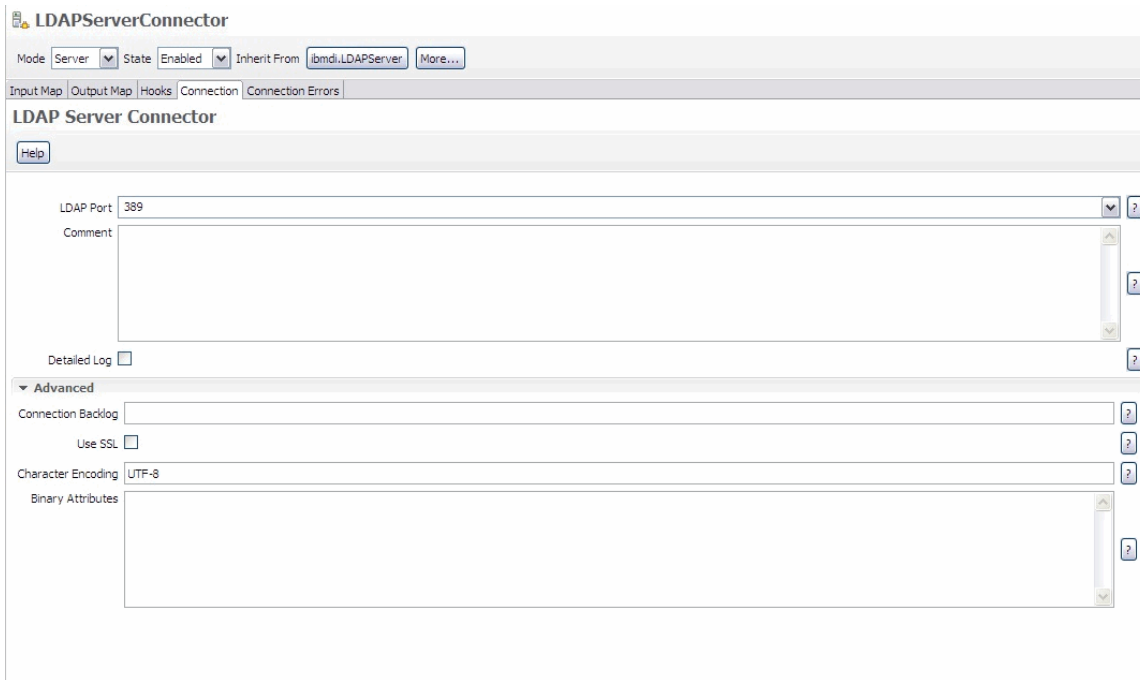
IBM Security Directory Integrator-Komponente als Server

Mit den hier aufgeführten Schritten können Sie eine IBM Security Directory Integrator-Komponente für SSL als Server konfigurieren.

Informationen zu diesem Vorgang

In diesem Beispiel wird der LDAP-Server-Connector verwendet. Der LDAP-Server-Connector überwacht LDAP-Anforderungen. Wenn eine LDAP-Anforderung eintrifft, analysiert der Connector die Anforderung und stellt die Anforderungsdaten für die entsprechende Fertigungslinie bereit. Die Fertigungslinie verarbeitet dann die Anforderung und stellt die Daten für die Antwort dem LDAP-Server-Connector bereit, damit dieser die LDAP-Antwort erstellen und zurück an den LDAP-Client senden kann. In der nachstehenden Anleitung wird Schritt für Schritt erläutert, wie Sie IBM Security Directory Integrator für SSL konfigurieren, wenn der LDAP-Server-Connector verwendet wird:

1. Rufen Sie den Serverschlüsselspeicher ab, indem Sie ihn entweder von einer Zertifizierungsstelle anfordern oder ein selbst signiertes Zertifikat erstellen (dies ist im Abschnitt Paar aus öffentlichem und privatem Schlüssel sowie selbst signiertes Zertifikat generieren erläutert).
2. Legen Sie die Details für den Schlüsselspeicher in der Datei `global.properties` oder `solution.properties` fest (siehe „SSL-Konfiguration von IBM Security Directory Integrator-Komponenten als Server“ auf Seite 111“).
3. Wählen Sie im GUI-Konfigurationsfenster für den Connector die Einstellung **SSL verwenden** aus. Möglicherweise müssen Sie den Abschnitt "Erweitert" einblenden, damit der Parameter sichtbar ist.



IBM Security Directory Integrator-Komponente als Client

Mit den hier aufgeführten Schritten können Sie eine IBM Security Directory Integrator-Komponente für SSL als Client konfigurieren.

Informationen zu diesem Vorgang

In diesem Beispiel wird der LDAP-Connector verwendet. Der LDAP-Connector stellt eine Verbindung zu einem LDAP-Server her und sendet eine LDAP-Anforderung. Nachdem der Server die LDAP-Antwort zurückgegeben hat, stellt der LDAP-Connector diese Antwort der Fertigungslinie zur weiteren Verarbeitung bereit. In der nachstehenden Anleitung wird Schritt für Schritt erläutert, wie Sie IBM Security Directory Integrator für SSL konfigurieren, wenn der LDAP-Connector verwendet wird:

1. Generieren Sie den Client-Truststore.
2. Importieren Sie das LDAP-Serverzertifikat in den Client-Truststore.
3. Legen Sie die Details für den Truststore in der Datei `global.properties` oder `solution.properties` fest (siehe „SSL-Konfiguration von IBM Security Directory Integrator-Komponenten als Client“ auf Seite 113“).

Mit der folgenden Befehlszeile wird ein vorhandenes Zertifikat in einen Schlüsselspeicher importiert (der Schlüsselspeicher wird erstellt, falls er noch nicht vorhanden ist):

```
keytool -import -trustcacerts -file myLDAPServerCert.cer
        -keystore myClientTruststore.jks -storepass myclientTruststorePassword
        -alias myTrustedLDAPServerAlias
```

Diese Befehlszeile importiert das Zertifikat `myLDAPServerCert.cer` unter dem Aliasnamen `myTrustedLDAPServerAlias` in den Schlüsselspeicher `myClientTruststore.jks`. Das Kennwort für den Zugriff auf den Schlüsselspeicher lautet `myclientTruststorePassword`.

Ferne Server-API

Sie können veranlassen, dass eine Clientanwendung Kontakt mit einem Server in IBM Security Directory Integrator aufnimmt. Eine Client-Task kann über Fernzugriff eine Server-Task aufrufen. Mithilfe der hier bereitgestellten Informationen und Links können Sie mehr darüber erfahren.

Die Absicherung einer Instanz eines IBM Security Directory Integrator-Servers wird im vorliegenden Abschnitt nicht behandelt; dies ist unter „Sicherheit für IBM Security Directory Integrator-Serverinstanz“ auf Seite 146 erläutert. Stattdessen ist in diesem Abschnitt beschrieben, wie Clientanwendungen einen Server ansprechen können.

IBM Security Directory Integrator unterstützt das Konzept einer fernen API, die einfach als "Server-API" bezeichnet wird. Bei diesem Konzept können Client-Tasks auf einem fernen IBM Security Directory Integrator-Server über eine Zugriffsschicht namens "RMI" Tasks aufrufen.

Anmerkung: Der "ferne Server" kann ohne Weiteres auf demselben Computer wie die Clientanwendung ausgeführt werden, beispielsweise dann, wenn Sie auf Ihrem lokalen Computer eine Serverinstanz starten und anschließend mit der fernen API über die Loopback-Adresse (127.0.0.1) auf diese Instanz zugreifen. Alle nachstehend erläuterten Konzepte sind auch dann gültig, wenn der ferne Server lokal ausgeführt wird.

Die Aufrufe der Server-API betreffen die folgenden Bereiche:

- Abruf von Serverinformationen
- Abruf von Informationen zu auf dem Server installierten Komponenten
- Lese- und Schreibvorgänge für durch den Server geladene Konfigurationen
- Ladevorgänge für neue Konfigurationen im Server
- Start, Abfrage und Stopp von Fertigungslinien
- Navigation in Fertigungslinien

Anmerkung: Da jeder aktive IBM Security Directory Integrator-Server immer mehr den Zugriff auf einen fernen Server benötigt, wurde der standardmäßige lokale Zugriff in einen standardmäßigen Fernzugriff geändert. Ab Version 7.1.1 ist die ferne Server-API standardmäßig aktiviert. Vor Version 7.1.1 war die Server-API nur für den lokalen Zugriff standardmäßig aktiviert. Als lokaler Zugriff ist hier ein Zugriff zu verstehen, der aus derselben Java Runtime Environment (JRE) heraus erfolgt. Zur Gewährleistung der Sicherheit ist für den Fernzugriff eine SSL-Clientauthentifizierung erforderlich. Der SSL-Zugriff mit Clientauthentifizierung wird durch den mit IBM Security Directory Integrator implementierten Beispielschlüsselspeicher und -Truststore bereitgestellt.

Die Server-API selbst ist in der Java-API-Dokumentation von IBM Security Directory Integrator (*tdi-installationsverzeichnis/docs/api*) dokumentiert. Sie können einen Browser zum Anzeigen dieser Dokumentation starten, indem Sie im Konfigurationseditor die Optionen **Hilfe -> Willkommen -> Javadocs** auswählen. Das in diesem Kontext relevante Paket ist `com.ibm.di.api`. Lesen Sie außerdem den Abschnitt zur Verwendung der Server-API im Abschnitt *Referenzinformationen* im IBM Knowledge Center for IBM Security Directory Integrator.

Der Konfigurationseditor verwendet die ferne API, um mit dem Server zu kommunizieren, den Sie für den Testbetrieb Ihrer Lösungen verwenden. Falls dieser IBM Security Directory Integrator-Server auf derselben Maschine ausgeführt wird, wird er häufig als "lokaler Entwicklungsserver" bezeichnet. Bei Konfigurationen, in denen der Konfigurationseditor von der Implementierungsplattform nicht unterstützt wird, können Sie den Entwicklungsserver auf dem Implementierungsserver und den Konfigurationseditor auf einer unterstützten Plattform wie Windows ausführen (diese Art der Ausführung wird als "ferner Konfigurationseditor" bezeichnet). Diese Strategie stellt für ferne und lokale Konfigurationsdateien eine einheitliche Schnittstelle bereit. Einige Aspekte des Konfigurationseditors bezüglich der Kommunikation mit einem fernen Implementierungsserver sind im Abschnitt „Fernes Konfigurationseditor verwenden“ auf Seite 160 erläutert.

Die Server-API wird durch eine Reihe von Servereigenschaften konfiguriert (siehe „Server-API konfigurieren“). Diese Eigenschaften werden in der Konfigurationsdatei `global.properties` des IBM Security Directory Integrator-Servers angegeben. Manche Eigenschaften verweisen ihrerseits auf zusätzliche Konfigurationsdateien und Schlüsselspeicherdateien.

Die Server-API bietet eine Reihe von sicherheitsrelevanten Funktionen (die sowohl für auf IBM Security Directory Integrator-Lösungen basierende Clients als auch für andere Clientanwendungen von Bedeutung sind). Die Aspekte des Zugriffsschutzes für die Server-API sind in den drei folgenden Abschnitten erläutert:

1. „SSL-Fernzugriff auf Server-API“ auf Seite 124 (Schutz des Transportkanals zu einem fernen IBM Security Directory Integrator-Server)
2. „Server-API-Authentifizierung“ auf Seite 126 (Verarbeitung der Clientauthentifizierung für einen IBM Security Directory Integrator-Server)
3. „Server-API-Autorisierung“ auf Seite 138 (Verarbeitung der Clientautorisierung für einen IBM Security Directory Integrator-Server, also die nach der Authentifizierung zulässigen Aktionen des Clients)

Server-API konfigurieren

Nachstehend finden Sie eine Liste der Eigenschaften, mit denen sich die Server-API konfigurieren lässt.

In diesem Zusammenhang sind die folgenden Eigenschaften relevant:

Eigenschaft	Standardwert	Beschreibung
<code>api.on</code>	true	Wenn diese Eigenschaft auf true gesetzt ist, wird die Server-API beim Start initialisiert und kann verwendet werden. Andernfalls wird die Server-API nicht initialisiert und kann nicht verwendet werden. Alle anderen Eigenschaften, deren Namen mit "api." beginnen, werden nur dann berücksichtigt, wenn die Eigenschaft <code>api.on</code> auf true gesetzt ist.
<code>api.audit.on</code>	false	Wenn diese Eigenschaft auf true gesetzt ist, ist die Prüffunktion aktiviert. Bei der Einstellung true wird selbst dann an jedem Prüfpunkt ein Prüfeintrag erstellt, wenn die Prüfbenachrichtigungen unterdrückt werden.
<code>api.user.registry</code>	<code>serverapi/registry.txt</code>	Wenn diese Eigenschaft konfiguriert ist, gibt sie den Dateinamen der Serverbenutzerregistry an.

Eigenschaft	Standardwert	Beschreibung
<i>api.user.registry.encryption.on</i>	false	Wenn diese Eigenschaft auf true gesetzt ist, entschlüsselt die Server-API beim Start die Datei der Serverbenutzerregistry.
<i>api.remote.on</i>	true	Wenn diese Eigenschaft auf true gesetzt ist, wird der ferne RMI-Teil der Server-API initialisiert und kann verwendet werden. Andernfalls wird der ferne RMI-Teil der Server-API nicht initialisiert und kann nicht verwendet werden.
<i>api.remote.ssl.on</i>	true	Wenn diese Eigenschaft auf true gesetzt ist, wird für RMI-Verbindungen der Server-API und ihrer JMX-Schicht SSL mit Client- und Serverauthentifizierung verwendet. Die Server-API verwendet dann das Serverzertifikat und den privaten Schlüssel (angegeben in den Eigenschaften <i>api.keystore</i> und <i>api.key.alias</i>) für SSL-Verbindungen. RMI-Clients müssen dieses Zertifikat anerkennen. Ist diese Eigenschaft auf false gesetzt, wird für Clientverbindungen SSL nicht verwendet und es findet keine Authentifizierung und Autorisierung statt. Verbindungen vom lokalen Host und von den in der Eigenschaft <i>api.remote.nonssl.hosts</i> aufgelisteten Hosts werden akzeptiert. Ist die Eigenschaft <i>api.remote.nonssl.hosts</i> leer, werden nur Verbindungen vom lokalen Host akzeptiert.
<i>api.remote.ssl.client.auth.on</i>	true	Wenn diese Eigenschaft auf true gesetzt ist, ist die SSL-Clientauthentifizierung für die ferne Server-API aktiviert.
<i>api.remote.naming.port</i>	1099	Wenn diese Eigenschaft angegeben ist, enthält sie den Port, an dem die RMI-Registry für Anforderungen empfangsbereit ist.
<i>api.remote.server.ports</i>	8700-8900	Diese Eigenschaft gibt den Bereich der Ports an, die von den verschiedenen RMI-Services verwendet werden können. Der Wert wird in Form einer durch Kommas getrennten Liste angegeben, in der durch Bindestriche angegebene Bereiche zulässig sind (Beispiel: 1, 3-5, 10). Portnummern müssen größer als 0 und kleiner als 65536 sein. Der Server verwendet diese Ports, um zusätzlich zu der Empfangsbereitschaft an den durch andere Eigenschaften definierten Ports für eingehende RMI-Serviceanforderungen empfangsbereit zu sein. Für abgehende RMI-Serviceanforderungen können wahlfreie Portnummern verwendet werden.
<i>com.ibm.di.default.bind.address</i>	*	Die Standardbindungsadresse für den gesamten IBM Security Directory Integrator-Server, also die Komponenten und die Server-API. * bedeutet eine Bindung an alle verfügbaren Netzchnittstellen. Ein fehlender oder ungültiger Wert bewirkt ebenfalls eine Bindung an alle Schnittstellen. Als Wert sollte nur eine einzige IP-Adresse angegeben werden. Diese Eigenschaft betrifft alle Server-Connectors, die einen Server-Socket erstellen.
<i>api.remote.bind.address</i>	*	Die Bindungsadresse für die RMI-Server-API. Der Wert dieser Eigenschaft setzt die Eigenschaft "com.ibm.di.default.bind.address" außer Kraft. * bedeutet, dass eine Verbindung zu allen verfügbaren Netzchnittstellen hergestellt wird. Ein fehlender oder leerer Wert führt dazu, dass wieder die Eigenschaft "com.ibm.di.default.bind.address" gültig ist. Als Wert sollte nur eine einzige IP-Adresse angegeben werden.
<i>api.truststore</i>	testserver.jks	Wenn diese Eigenschaft angegeben ist, definiert sie die Schlüsselspeicherdatei, die die öffentlichen Zertifikate aller fernen Benutzer der Server-API enthält.
{protect}- <i>api.truststore.pass</i>	server (standardmäßig verschlüsselt)	Wenn diese Eigenschaft angegeben ist, enthält sie das Kennwort für die Schlüsselspeicherdatei, die in der Eigenschaft <i>api.remote.server.truststore</i> benannt ist.
<i>api.remote.nonssl.hosts</i>		Wenn diese Eigenschaft angegeben ist, enthält sie eine Liste der IP-Adressen, von denen Verbindungen ohne SSL akzeptiert werden (Hostnamen sind nicht zulässig). Verwenden Sie als Begrenzungszeichen zwischen den IP-Adressen ein Leerzeichen, ein Komma oder ein Semikolon. Diese Eigenschaft wird nur dann berücksichtigt, wenn die Eigenschaft <i>api.remote.ssl.on</i> auf false gesetzt ist.
<i>api.jmx.on</i>	false	Wenn diese Eigenschaft auf true gesetzt ist, wird die JMX-Schicht der Server-API beim Start initialisiert und kann verwendet werden. Andernfalls wird die JMX-Schicht nicht initialisiert und kann nicht verwendet werden.
<i>api.jmx.remote.on</i>	false	Wenn diese Eigenschaft auf true gesetzt ist, wird die ferne JMX-Schnittstelle (wie durch JSR160 definiert) initialisiert und kann verwendet werden. Andernfalls wird die ferne JMX-Schnittstelle nicht initialisiert und kann nicht verwendet werden.

Eigenschaft	Standardwert	Beschreibung
<i>api.config.folder</i>	configs	Wenn diese Eigenschaft auf den Wert <i>tdi-stammverzeichnis/configs</i> gesetzt ist, können nur über die Server-API nur diejenigen Konfigurationsdateien bearbeitet werden, die sich in diesem Ordner befinden.
<i>api.config.lock.timeout</i>	0	Wenn diese Eigenschaft auf 0 gesetzt ist, wird kein Zeitlimit angewendet.
<i>api.custom.method.invoke.on</i>	false	Die Fähigkeit zur Verwendung von Methoden <code>Session.invokeCustom()</code> kann aktiviert oder inaktiviert werden. Die Standardeinstellung ist "false" (= inaktiviert). Ist für diese Eigenschaft der Wert true angegeben, können Benutzer diese Methoden verwenden. Beim Wert "false" wird eine Ausnahmebedingung ausgelöst.
<i>api.custom.method.invoke.allowed.classes</i>		Falls diese Eigenschaft angegeben ist, enthält sie eine Liste der Klassen, die durch die Methoden der Server-API für einen angepassten Methodenaufruf (<code>Session.invokeCustom(...)</code>) direkt aufgerufen werden können. Diese Eigenschaft wird nur dann berücksichtigt, wenn die Eigenschaft <i>api.custom.method.invoke.on</i> auf den Wert true gesetzt ist. Die Klassen in dieser Liste müssen durch ein Leerzeichen, ein Komma oder ein Semikolon voneinander abgegrenzt werden.
<i>api.custom.authentication.</i>	[ldap]	Wenn diese Eigenschaft angegeben ist, verweist sie auf eine JavaScript-Textdatei, die angepassten Authentifizierungscode enthält. Um den integrierten LDAP/JAAS-Authentifizierungsmechanismus zu aktivieren, setzen Sie diese Eigenschaft auf [ldap]/[jaas].
<i>com.ibm.di.server.id</i>		Wenn diese Eigenschaft angegeben ist, enthält sie die Server-ID. Ordnen Sie für jeden Server in der Gruppe der Server, die über dieselbe IP-Adresse und denselben Port ausgeführt werden, einen eindeutigen Wert zu.
<i>api.config.load.timeout</i>	2	Wenn diese Eigenschaft angegeben ist, enthält sie den Zeitlimitwert für das Laden der Server-API-Konfiguration (der Wert wird in Minuten angegeben). Diese Eigenschaft wurde in IBM Security Directory Integrator v7.0 hinzugefügt.
<i>api.notification.suppress</i>	di.server.api.authenticate di.server.api.authorize.*	Wenn diese Eigenschaft angegeben ist, enthält sie eine Liste der Serverbenachrichtigungstypen, die unterdrückt werden sollen. Benachrichtigungen unterdrückten Typs werden durch das Benachrichtigungsframework nicht weitergegeben. Die Benachrichtigungstypen in der Liste werden durch Leerzeichen voneinander getrennt. Die Verwendung von Platzhalterzeichen ist zulässig. Beispiel: <code>api.notification.suppress=di.al.* di.ci.start</code> Das obige Beispiel unterdrückt alle die fertigungslinienbezogenen Benachrichtigungen sowie Benachrichtigungen über das Starten einer Konfigurationsinstanz. Falls die Eigenschaft fehlt oder leer ist, werden keine Benachrichtigungen unterdrückt. Diese Eigenschaft wurde in IBM Security Directory Integrator v7.0 hinzugefügt.
<i>api.client.ssl.custom.properties.on</i>	true	Diese Eigenschaft aktiviert angepasste SSL-Eigenschaften für Server-API-Clients. Wenn sie auf "true" gesetzt ist, werden von den Server-API-Clients Eigenschaften "api.client.*" verwendet. Andernfalls werden die Standardeigenschaften "javax.net.ssl.*" verwendet. Diese Eigenschaft wurde in IBM Security Directory Integrator v7.1 hinzugefügt.
<i>api.client.keystore</i>	serverapi/testadmin.jks	Diese Eigenschaft gibt den Schlüsselspeicher für Server-API-Clients an.
<i>api.client.keystore.pass</i>	administrator	Diese Eigenschaft gibt das Kennwort für den Schlüsselspeicher an, der in der Eigenschaft "api.client.keystore" benannt ist.
<i>api.client.keystore.type</i>	jks	Diese Eigenschaft gibt den Typ der Schlüsselspeicherdatei an, die in der Eigenschaft "api.client.keystore" benannt ist, und ist optional. Wird diese Eigenschaft nicht angegeben, wird das Standardschlüsselspeicherformat für die JVM verwendet.
<i>api.client.key.pass</i>	administrator	Diese Eigenschaft gibt das Kennwort für den privaten Schlüssel an. Der Schlüssel befindet sich in dem Schlüsselspeicher, der in der Eigenschaft "api.client.keystore" definiert ist.
<i>api.client.truststore</i>	serverapi/testadmin.jks	Diese Eigenschaft gibt den Truststore für Server-API-Clients an.
<i>api.client.truststore.pass</i>	administrator	Diese Eigenschaft gibt das Kennwort für den Truststore an, der in der Eigenschaft "api.client.truststore" benannt ist.

Eigenschaft	Standardwert	Beschreibung
<code>api.client.truststore.type</code>	jks	Diese Eigenschaft gibt den Typ der Schlüsselspeicherdatei an, die in der Eigenschaft "api.client.truststore" benannt ist, und ist optional. Wird diese Eigenschaft nicht angegeben, wird das Standardschlüsselspeicherformat für die JVM verwendet.

Anmerkung: Unabhängig davon, ob der Client ein Java-Programm oder eine andere Instanz des IBM Security Directory Integrator-Servers ist, sind die Java-Systemeigenschaften, die die Server-API für ihre Konfiguration verwendet, identisch. Es ist jedoch zu beachten, dass diese Java-Systemeigenschaften möglicherweise unterschiedlich festgelegt werden. In IBM Security Directory Integrator werden diese Eigenschaften normalerweise durch eine Bearbeitung der Datei `global.properties` oder `solution.properties` festgelegt. In einem Java-Programm können sie hingegen entweder mit dem Java-Befehlszeilenswitch "-D" in der Befehlszeile oder durch die Verwendung von Java-Code im Java-Programm mit der Java-Standardmethode `java.lang.System.setProperty(key,value)` festgelegt werden.

Zugriff auf ferne Server-API in einem virtuellen privaten Netz

Nachstehend erhalten Sie Informationen zum Zugriff auf eine ferne Server-API über ein Client-VPN.

Wenn in einem virtuellen privaten Netz (VPN) von einem Client auf die ferne Server-API zugegriffen wird, ordnet das VPN dem Computer des Server-API-Clients eine IP-Adresse zu. Diese vom VPN zugeordnete IP-Adresse muss in einer RMI-Java-Systemeigenschaft angegeben werden. Falls es sich bei dem Server-API-Client um den fernen Konfigurationseditor handelt, kann diese Eigenschaft in der Datei `global.properties` oder `solution.properties` durch das Hinzufügen der folgenden Zeile zur Eigenschaftendatei festgelegt werden:

```
java.rmi.server.hostname=<ip-adresse>
```

Hierbei steht *ip-adresse* für die durch das VPN zugeordnete IP-Adresse.

Handelt es sich bei dem Server-API-Client um ein angepasstes Java-Programm, kann diese Eigenschaft folgendermaßen innerhalb des Java-Codes festgelegt werden:

```
java.lang.System.setProperty("java.rmi.server.hostname", "ip-adresse");
```

Hierbei steht *ip-adresse* für die durch das VPN zugeordnete IP-Adresse.

Bitte beachten Sie, dass die RMI-Java-Systemeigenschaft vor dem RMI-Code festgelegt werden muss, der sich auf die Server-API bezieht.

Zugriffsoptionen für die Server-API

Nachstehend erhalten Sie Informationen zu den verschiedenen Zugriffsoptionen für die Server-API.

Die Server-API kann auf mehreren Wegen verwendet werden:

- Der Zugriff auf die Server-API kann aus dem fernen Konfigurationseditor über eine Netzverbindung erfolgen.
- Der Zugriff auf die Server-API kann aus IBM Security Directory Integrator-Komponenten erfolgen, die in einem fernen IBM Security Directory Integrator-Server ausgeführt werden (ferner Server-API-Zugriff). Beispiele für solche Komponenten:
 - Systemwarteschlangenconnector

- Serverbenachrichtigungsconnector
usw.
- Der Zugriff auf die Server-API kann innerhalb derselben Java Virtual Machine des IBM Security Directory Integrator-Servers erfolgen (lokaler Server-API-Zugriff). In diesem Fall kann die Server-API zusätzlich zu den oben genannten Optionen aus JavaScript heraus in Hooks oder aus der Scriptkomponente heraus erreicht werden.
- Der Zugriff auf die Server-API kann aus Java-Anwendungen heraus erfolgen, die nicht zu IBM Security Directory Integrator gehören. Für diese Funktionsweise bestehen die folgenden Bedingungen:
 - Java 7.0.4 oder höher ist auf der Clientseite erforderlich.
 - Die folgenden JAR-Dateien müssen in der CLASSPATH-Angabe der fernen Seite enthalten sein:
 - jars/common/diserverapi.jar
 - jars/common/diserverapirmi.jar
 - jars/3rdparty/others/log4j-1.2.16.jar
 - jars/common/miconfig.jar
 - jars/common/miserver.jar
 - jars/common/mmconfig.jar
 - jars/common/tdiresource.jar
 - jars/3rdparty/IBM/icu4j-50_1_1.jar
 - jars/3rdparty/IBM/jlog.jar

Sie können diese JAR-Dateien aus der IBM Security Directory Integrator-Installation kopieren.
- Falls angepasste, nicht aus IBM Security Directory Integrator stammende Objekte in der Lösung verwendet werden, die mit der Server-API implementiert wird (z. B. Attributwerte eines Eintrags, der über die Verbindung übertragen wird), müssen die entsprechenden Java-Klassen auf der Clientseite ebenfalls verfügbar sein. Diese Klassen müssen serialisierbar und in der CLASSPATH-Angabe der Client-JVM enthalten sein.

SSL-Fernzugriff auf Server-API

Die ferne Schnittstelle der Server-API kann SSL verwenden. Nachstehend erhalten Sie Informationen zum Fernzugriff.

Die Server-API stellt zwei Gruppen von Schnittstellen bereit, nämlich lokale Schnittstellen und ferne Schnittstellen. Lediglich die fernen Schnittstellen verwenden SSL. Die lokalen Schnittstellen verwenden SSL nicht, da der Zugriff innerhalb der Grenzen der Java Virtual Machine erfolgt. In einem Zugriffsszenario für die Server-API kann IBM Security Directory Integrator als Server, als Client sowie als Server und Client agieren. Wenn SSL zusammen mit der Server-API verwendet wird, müssen ein Schlüsselspeicher und ein Truststore konfiguriert werden. Für die Konfiguration dieser Elemente gibt es zwei Optionen. Welche Option verwendet wird, richtet sich danach, ob und mit welchem Wert die Java-Systemeigenschaft `api.client.ssl.custom.properties.on` vorhanden ist.

Spezielle SSL-Eigenschaften der Server-API verwenden

Verwenden Sie die aufgeführten Eigenschaften zum Konfigurieren von SSL.

Wenn die Java-Systemeigenschaft "api.client.ssl.custom.properties.on" auf *true* gesetzt ist, wird SSL durch die folgenden Java-Systemeigenschaften von IBM Security Directory Integrator konfiguriert, die speziell für die Server-API gelten:

- **api.client.keystore** – Diese Eigenschaft gibt die Schlüsselspeicherdatei an, die das Clientzertifikat enthält.
- **api.client.keystore.pass** – Diese Eigenschaft gibt das Kennwort der Schlüsselspeicherdatei an, die in der Eigenschaft "api.client.keystore" definiert ist.
- **api.client.keystore.type** – Diese Eigenschaft gibt den Typ der Schlüsselspeicherdatei an, die in der Eigenschaft "api.client.keystore" benannt ist, und ist optional. Wird diese Eigenschaft nicht angegeben, wird das Standardschlüsselspeicherformat für die JVM verwendet.
- **api.client.key.pass** – Diese Eigenschaft gibt das Kennwort des privaten Schlüssels an, der in der Schlüsselspeicherdatei gespeichert ist, die in der Eigenschaft "api.client.keystore" definiert ist. Falls diese Eigenschaft fehlt, wird stattdessen das in der Eigenschaft **api.client.keystore.pass** angegebene Kennwort verwendet.
- **api.client.truststore** – Diese Eigenschaft gibt die Schlüsselspeicherdatei an, die das öffentliche Zertifikat des IBM Security Directory Integrator-Servers enthält.
- **api.client.truststore.pass** – Diese Eigenschaft gibt das Kennwort der Schlüsselspeicherdatei an, die in der Eigenschaft "api.client.truststore" definiert ist.
- **api.client.truststore.type** – Diese Eigenschaft gibt den Typ der Schlüsselspeicherdatei an, die in der Eigenschaft "api.client.truststore" benannt ist, und ist optional. Wird diese Eigenschaft nicht angegeben, wird das Standardschlüsselspeicherformat für die JVM verwendet.

Verwenden Sie die speziellen SSL-Eigenschaften für die Server-API, wenn Ihre Clientanwendung die Java-Standard-eigenschaften für SSL verwendet. Bei den Java-Standard-eigenschaften für SSL handelt es sich um Eigenschaften, mit denen ein anderer SSL-Kanal konfiguriert wird, der durch dieselbe Anwendung genutzt wird.

Sie können diese Eigenschaften als JVM-Argumente in der Befehlszeile angeben.
Beispiel:

```
java MyTDIServerAPIClientApp
-Dapi.client.ssl.custom.properties.on=true
-Dapi.client.truststore=C:\TDI\serverapi\testadmin.jks
-Dapi.client.truststore.pass=administrator
-Dapi.client.keystore=C:\TDI\serverapi\testadmin.jks
-Dapi.client.keystore.pass=administrator
```

Dieses Beispiel bezieht sich auf die Schlüsselspeicherbeispieldatei "testadmin.jks", die mit IBM Security Directory Integrator ausgeliefert wird. Beachten Sie, dass sie sowohl den privaten Schlüssel des Clients als auch den öffentlichen Schlüssel des IBM Security Directory Integrator-Servers enthält und somit als Schlüsselspeicher und auch als Truststore eingesetzt wird.

Sie können diese Eigenschaften in der Datei `global.properties` oder `solution.properties` angeben, wenn es sich bei dem Client um einen IBM Security Directory Integrator-Server handelt.

Java-Standard-systemeigenschaften für SSL verwenden

Verwenden Sie die aufgeführten JVM-Befehle zum Konfigurieren von SSL.

Wenn die Java-Systemeigenschaft `api.client.ssl.custom.properties.on` fehlt oder auf die Einstellung "false" gesetzt ist, werden zur Konfiguration des SSL-Kanals die JSSE-Standard-systemeigenschaften verwendet. Beim Konfigurieren des Schlüsselspeichers und des Truststores, die von der Clientanwendung verwendet werden, ist die JSSE-Standard-prozedur zu befolgen.

Sie können diese Eigenschaften als JVM-Argumente in der Befehlszeile angeben.
Beispiel:

```
java MyTDIServerAPIClientApp
-Djavax.net.ssl.keyStore=C:\TDI\serverapi\testadmin.jks
-Djavax.net.ssl.keyStorePassword=administrator
-Djavax.net.ssl.trustStore=C:\TDI\serverapi\testadmin.jks
-Djavax.net.ssl.trustStorePassword=administrator
```

Auch diese Eigenschaften können Sie in der Datei `global.properties` oder `solution.properties` angeben, wenn es sich bei dem Client um einen IBM Security Directory Integrator-Server handelt.

Server-API-Authentifizierung

Beim Versuch, eine Server-API-Sitzung einzurichten, können Sie die Server-API-Authentifizierung verwenden.

Die Server-API-Authentifizierung kommt normalerweise in einem Kontext zum Tragen, in dem ein Client der fernen Server-API eine Server-API-Sitzung aufbaut. In diesem Szenario ist die Grundlage der Logik für die Server-API-Authentifizierung dargestellt, da die Server-API verschiedene Arten der Clientauthentifizierung ermöglicht. Vor einer Erläuterung der unterschiedlichen Authentifizierungsmechanismen wird zunächst das Szenario beschrieben, in dem ein lokaler Client eine Sitzung mit der lokalen Server-API aufbaut.

Lokale Clientsitzung

Mit den hier aufgeführten Anweisungen können Sie mit einem Client lokal arbeiten. Nachstehend erhalten Sie Informationen zu den Optionen.

Eine lokale Clientsitzung ist eine Sitzung, die durch einen Client aufgebaut wird, der in derselben Java Virtual Machine wie der IBM Security Directory Integrator-Server ausgeführt wird. Beispiele für solche Sitzungen sind lokale Sitzungen für den Zugriff auf die lokale Server-API, die aus JavaScript-Code in Hooks oder in einer Scriptkomponente oder aus Connectors und Funktionskomponenten heraus aufgebaut werden, die als Teil einer Fertigungslinie in demselben IBM Security Directory Integrator-Server ausgeführt werden, usw. Wenn ein lokaler Client eine lokale Server-API-Sitzung aufbaut, kann er zwei Optionen verwenden:

- Keine Angabe eines Paares aus Benutzername und Kennwort – In diesem Fall wird die lokale Server-API-Sitzung aufgebaut und der Client wird mit der Rolle "admin" autorisiert. Weitere Informationen zu Server-API-Rollen finden Sie unter „Server-API-Autorisierung“ auf Seite 138.
- Angabe eines Paares aus Benutzername und Kennwort – In diesem Fall wird die Server-API-Sitzung erst aufgebaut, nachdem der im Paar aus Benutzername und Kennwort angegebene Benutzername gemäß der Autorisierungslogik der Server-API, die im Abschnitt „Server-API-Autorisierung“ auf Seite 138 beschrieben ist, autorisiert wurde. Diese Option wird normalerweise verwendet, wenn für die Authentifizierung eine bestimmte Benutzer-ID benötigt wird, beispielsweise für Demos, die Prototyperstellung usw.

Ferne Clientsitzung

Mit den hier aufgeführten Anweisungen können Sie mit einem Client über Fernzugriff arbeiten. Nachstehend erhalten Sie Informationen zu den Authentifizierungstypen.

Eine ferne Clientsitzung ist eine Sitzung, die durch einen Client aufgebaut wird, der nicht in derselben Java Virtual Machine wie der IBM Security Directory Integrator-Server ausgeführt wird. Beispiele für solche Sitzungen sind Sitzungen für den

Zugriff auf eine ferne Server-API, die aus dem Konfigurationseditor heraus aufgebaut werden, oder eine Java-Anwendung, die auf einen IBM Security Directory Integrator-Server zugreifen will. Für diese Zugriffsarten gibt es die folgenden Methoden zur Authentifizierung beim IBM Security Directory Integrator-Server:

JAAS-Authentifizierung

Mithilfe von JAAS-Authentifizierung können Sie die Zugriffssteuerung eines Benutzers auswerten. Nachstehend erhalten Sie Informationen zum Konfigurieren der Eigenschaften.

Die Authentifizierung mit Java Authentication and Authorization Service (JAAS) wird als Authentifizierungsmodul für Server-APIs von IBM Security Directory Integrator unterstützt. JAAS besteht aus einer Reihe von APIs, die Services in die Lage versetzen, Authentifizierungen vorzunehmen und für Benutzer eine Zugriffssteuerung durchzusetzen. Die JAAS-Authentifizierung wird durch die Server-API von IBM Security Directory Integrator unterstützt. In den Clients der IBM Security Directory Integrator-Server-API (z. B. CLI und AMC) sind keine Änderungen erforderlich, damit das JAAS-Authentifizierungsmodul verwendet werden kann.

Zur Verwendung der JAAS-Authentifizierung müssen Sie die entsprechenden Eigenschaften in der Datei `global.properties` oder `solution.properties` konfigurieren. Des Weiteren muss die JAAS-Anmeldung installiert sein.

SSL-basierte Authentifizierung

Sie können die zweistufige Prüfung für die Berechtigungsnachweise des Clients über die SSL-basierte Authentifizierung verwenden.

Dies ist der einzige Authentifizierungsmechanismus, der in IBM Security Directory Integrator 6.0 verfügbar ist. Der SSL-basierten Authentifizierung liegt eine zweistufige Prüfung für die Berechtigungsnachweise des Clients zugrunde.

1. Zunächst prüft der IBM Security Directory Integrator-Server, ob ein Client (repräsentiert durch sein SSL-Zertifikat) für den Zugriff auf den IBM Security Directory Integrator-Server berechtigt ist. Hierzu überprüft er, ob das SSL-Zertifikat des Clients im Truststore des IBM Security Directory Integrator-Servers enthalten ist, ob also der IBM Security Directory Integrator-Server diesem Client vertraut. Die Überprüfung, ob das Zertifikat des Clients im Truststore des Servers enthalten ist, ist Bestandteil der Sequenz für den SSL-Handshake.

Achtung: Ein Beispiel für ein Clientzertifikat, das dem Beispiel des Serverzertifikats in der Datei `testserver.jks` entspricht, wird in der Datei `serverapi/testadmin.jks` bereitgestellt (das Kennwort des Zertifikats lautet "administrator"). Wie bei allen Standardsicherheitsparametern sollten Sie nicht diese Beispiele verwenden, sondern eigene Client/Server-Zertifikate erstellen und diese in den Eigenschaftendateien angeben. Informationen hierzu finden Sie unter „Zertifikate für Web-Service-Suite von IBM Security Directory Integrator“ auf Seite 167.

Der Truststore ist in der Datei enthalten, die durch die Eigenschaft `api.truststore` angegeben wird.

2. Falls die Truststoreprüfung erfolgreich verläuft, prüft der Server, ob der definierte Name im SSL-Zertifikat des Clients mit einer Benutzer-ID in der Benutzerregistry der Server-API übereinstimmt (siehe „Benutzerregistry der Server-API“ auf Seite 140). Falls der definierte Name im Clientzertifikat nicht mit einer der Benutzer-IDs in der Benutzerregistry der Server-API übereinstimmt, wird die Verbindungsanforderung des Clients verweigert. Dieser zweite Schritt könnte ebenfalls als Bestandteil der Autorisierungssequenz betrachtet werden.

Der SSL-basierte Authentifizierungsmechanismus kann in IBM Security Directory Integrator inaktiviert werden. In der IBM Security Directory Integrator-Serverkonfigurationsdatei `global.properties` oder `solution.properties` ist eine zusätzliche Eigenschaft namens `api.remote.ssl.client.auth.on` verfügbar. Wenn diese Eigenschaft auf "true" gesetzt ist, erfordert der IBM Security Directory Integrator-Server die Clientauthentifizierung innerhalb des SSL-Handshakes (was dem Mechanismus zur SSL-basierten Authentifizierung aus IBM Security Directory Integrator 6.0 entspricht). Die SSL-Clientauthentifizierung für die Server-API von IBM Security Directory Integrator ist nicht davon abhängig, ob ein Paar aus Benutzername und Kennwort angegeben wird. Dies bedeutet, dass der Mechanismus für die SSL-basierte Authentifizierung aus IBM Security Directory Integrator 6.0 verwendet wird, falls kein Paar aus Benutzername und Kennwort bereitgestellt wird. Wird jedoch ein Paar aus Benutzername und Kennwort angegeben, muss der Client auch in diesem Fall sein SSL-Zertifikat zur Authentifizierung senden. Die Benutzer-ID für die Authentifizierung (und die in einem späteren Schritt erfolgende Autorisierung) wird jedoch aus dem angegebenen Benutzernamen entnommen.

Wenn die Eigenschaft `api.remote.ssl.client.auth.on` auf "false" gesetzt ist, kann die SSL-basierte Authentifizierung nicht verwendet werden. Ist diese Eigenschaft nicht angegeben, wird der Wert "false" vorausgesetzt.

Auf Benutzername/Kennwort basierende Authentifizierung

Sie können die auf Benutzername/Kennwort basierende Authentifizierung mithilfe eines Authentifizierungshooks ausführen.

Bei diesem Mechanismus muss ein Client beim Öffnen seiner Server-API-Verbindung zum IBM Security Directory Integrator-Server einen Benutzernamen und ein Kennwort angeben. Zum Konfigurieren dieser Authentifizierungsmethode wird ein Authentifizierungshook verwendet.

Authentifizierungshook

Dieser Hook ermöglicht die Bereitstellung von angepasstem JavaScript-Code, der eine auf Benutzername und Kennwort basierende Authentifizierung vornimmt. Mithilfe dieses Hooks können Paketersteller/Implementierer angepassten JavaScript-Code schreiben, der anhand eines angegebenen Paares aus Benutzername und Kennwort bestimmt, ob die Authentifizierung erfolgreich vorgenommen werden soll oder nicht.

Die Eigenschaft, die diese Authentifizierung über ein angepasstes JavaScript ermöglicht, wird in der IBM Security Directory Integrator-Serverkonfigurationsdatei `global.properties` oder `solution.properties` angegeben und heißt `api.custom.authentication`. Die Eigenschaft "api.custom.authentication" zeigt auf eine JavaScript-Textdatei auf der Platte, die angepassten Authentifizierungscode enthält. Falls diese Eigenschaft nicht angegeben ist, wird der SSL-basierte Authentifizierungsmechanismus aus IBM Security Directory Integrator 6.0 verwendet. Ist die Eigenschaft "api.custom.authentication" angegeben, wird der in der angegebenen Datei enthaltene JavaScript-Code für jede auf Benutzername und Kennwort basierende Authentifizierungsanforderung ausgeführt.

Das Authentifizierungsscript kann auf das vordefinierte Scriptobjekt "userdata" zugreifen. Dieses Objekt stellt die beiden folgenden öffentlichen Elemente bereit:

- **userdata.username:** Dieses Element enthält den Namen des Benutzers, der die Authentifizierung anfordert.
- **userdata.password:** Dieses Element enthält das vom Benutzer angegebene Kennwort.

Das Script kann alle benötigten Überprüfungen und Authentifizierungsaktionen ausführen. Im Objekt `ret` wird zurückgegeben, ob die Authentifizierung erfolgreich war:

- Ist `ret.auth = true` gesetzt, war die Authentifizierung erfolgreich.
- Ist `ret.auth = false` gesetzt, war die Authentifizierung nicht erfolgreich. In diesem Fall kann das Authentifizierungsscript weitere Informationen über den Grund bereitstellen, aus dem die Authentifizierung fehlschlug. Hierzu wird das Attribut `ret.errordescr` (z. B. `ret.errordescr = "Invalid user name"`) und das Attribut `ret.errorcode` (z. B. `ret.errorcode = 1`) verwendet.

Die Felder für die Beschreibung und den Fehlercode werden durch die Ausnahmebedingung "AuthenticationException" bereitgestellt, die bei einer nicht erfolgreichen Authentifizierung durch die Server-API ausgelöst wird.

Das Authentifizierungsscript hat Zugriff auf das Scriptobjekt "main". Es kann zur Protokollierung von angepassten Nachrichten in der IBM Security Directory Integrator-Serverprotokolldatei verwendet werden (z. B. `main.logmsg("Authentication failed for user : " + userdata.username)`).

Beispiel für Authentifizierungshook

Ein Beispiel für eine JavaScript-Datei mit einem Authentifizierungshook ist im Verzeichnis `tdi-installationsverzeichnis/examples` verfügbar. Es veranschaulicht, wie der JavaScript-Code eines Authentifizierungshooks aussehen könnte. Dieses JavaScript-Beispiel kann auch als Grundlage für einen echten IBM Security Directory Integrator-Authentifizierungshook verwendet werden. Das JavaScript-Beispiel zeigt, wie ein Authentifizierungshook einen LDAP-Server (IBM Security Directory Server, Active Directory usw.) zur Authentifizierung von Clientanforderungen einsetzen kann.

Die JavaScript -Datei heißt "ldap_auth.js". Sie befindet sich im Ordner `examples/auth_ldap` des IBM Security Directory Integrator-Servers. Um dieses Beispiel für einen LDAP-Authentifizierungsmechanismus zu implementieren, können Sie diese Datei in den IBM Security Directory Integrator-Lösungsordner kopieren und die Eigenschaft `api.custom.authentication=ldap_auth.js` in der Datei `global.properties` oder `solution.properties` angeben. Der JavaScript-Code in "ldap_auth.js" versucht, mit den angegebenen Werten für Benutzername und Kennwort eine Bindung an einen LDAP-Server herzustellen. Falls die Bindeoperation erfolgreich verläuft, gibt das Script eine erfolgreiche Authentifizierung an. Andernfalls wird die Authentifizierung abgelehnt. Die Details für das Herstellen der Verbindung zum LDAP-Server (z. B. die Server-URL) sind im Script "ldap_auth.js" angegeben. Dies bedeutet, dass Sie vor Verwendung des Scripts diese Datei bearbeiten und die richtigen Verbindungsparameter festlegen müssen. Das Beispielscript "ldap_auth.js" hat den folgenden Inhalt:

```
env = new Packages.java.util.Hashtable();
env.put("java.naming.factory.initial", "com.sun.jndi.ldap.LdapCtxFactory");
env.put("java.naming.provider.url", "ldap://192.168.113.54:389");
env.put("java.naming.security.principal", userdata.username);
env.put("java.naming.security.credentials", userdata.password);
env.put(Packages.javax.naming.Context.SECURITY_AUTHENTICATION, "simple");

main.logmsg("Authentication request for user: " + userdata.username);

try
{
    mCtx = new Packages.javax.naming.directory.InitialDirContext(env);
    ret.auth = true;
}
catch(e)
{
```

```
ret.auth = false;
ret.errordescr = e.toString();
// ret.errorcode = "49";
}
```

Unterstützung der LDAP-Authentifizierung

Die IBM Security Directory Integrator-Server-API unterstützt die LDAP-Authentifizierung. Dies versetzt Sie in die Lage, Ihre vorhandene LDAP-Infrastruktur zu nutzen, die bereits Benutzer-IDs und Kennwörter enthält.

Konfiguration der LDAP-Authentifizierung:

Sie können die LDAP-Authentifizierung konfigurieren, indem Sie die aufgeführten Eigenschaften bearbeiten.

Zur Verwendung der LDAP-Authentifizierung müssen die entsprechenden Eigenschaften in der Datei `global.properties` oder `solution.properties` konfiguriert werden. Die folgende Liste enthält diese Eigenschaften zusammen mit einer jeweiligen Beschreibung:

api.custom.authentication

Dies ist dieselbe Eigenschaft, die für die Authentifizierung mit Benutzername und Kennwort verwendet wird. Weitere Informationen zur Authentifizierung mit Benutzername und Kennwort enthält der Abschnitt "Auf Benutzername und Kennwort basierende Authentifizierung". Diese Eigenschaft zeigt auf eine JavaScript-Textdatei auf der Platte, die angepassten Authentifizierungscode enthält. Es ist zulässig, dass diese Eigenschaft vom Benutzer nicht angegeben wird. In diesem Fall kann nur der SSL-basierte Authentifizierungsmechanismus von IBM Security Directory Integrator 6.0 verwendet werden. Die Authentifizierung über den Benutzernamen und das Kennwort von IBM Security Directory Integrator Version 7.2 funktioniert nicht. Setzen Sie diese Eigenschaft auf "[ldap]", um den in IBM Security Directory Integrator Version 7.2 integrierten LDAP-Authentifizierungsmechanismus zu aktivieren (also "api.custom.authentication=[ldap]"). Alle Eigenschaften, die mit der Zeichenfolge `api.custom.authentication.ldap` beginnen, werden nur dann berücksichtigt, wenn die Eigenschaft `api.custom.authentication` auf `[ldap]` gesetzt ist.

api.custom.authentication.ldap.critical

Dieser Parameter gibt das Verhalten der Server-API an, wenn das LDAP-Authentifizierungsmodul beim Start nicht initialisiert werden kann. Falls dieser Parameter auf "true" gesetzt ist, schlägt die Initialisierung der Server-API fehl und die Server-API wird nicht gestartet.

Fehlt dieser Parameter oder ist er auf "false" gesetzt, protokolliert die Server-API zwar den Fehler für die Initialisierung der LDAP-Authentifizierung, wird aber gestartet. Bei jedem Empfang einer Authentifizierungsanforderung durch die Server-API wird solange versucht, das LDAP-Authentifizierungsmodul zu initialisieren, bis das LDAP-Authentifizierungsmodul initialisiert wurde.

api.custom.authentication.ldap.hostname

Diese Eigenschaft gibt den Hostnamen des LDAP-Servers an. Falls die angepasste LDAP-Authentifizierung verwendet wird, ist diese Eigenschaft erforderlich.

api.custom.authentication.ldap.port

Diese Eigenschaft gibt die Portnummer des LDAP-Servers an (z. B. 389 ohne SSL oder 636 mit SSL). Falls die angepasste LDAP-Authentifizierung verwendet wird, ist diese Eigenschaft erforderlich.

api.custom.authentication.ldap.ssl

Diese Eigenschaft gibt an, ob bei der Kommunikation mit dem LDAP-Server SSL verwendet wird. Ist sie auf "true" gesetzt, wird SSL verwendet; andernfalls wird SSL nicht verwendet.

api.custom.authentication.ldap.searchbase

Diese Eigenschaft gibt die LDAP-Verzeichnisposition an, an der Benutzersuchen ausgeführt werden. Wenn sie nicht angegeben ist, werden keine Benutzersuchen ausgeführt.

api.custom.authentication.ldap.adminidn

Diese Eigenschaft gibt den definierten Namen eines LDAP-Serveradministrators an, der für Benutzersuchen verwendet wird. Ist sie nicht angegeben, wird für Benutzersuchen das anonyme Binden verwendet.

api.custom.authentication.ldap.adminpassword

Diese Eigenschaft enthält das Kennwort für den definierten Namen des LDAP-Serveradministrators.

api.custom.authentication.ldap.userattribute

Diese Eigenschaft enthält das Benutzer-ID-Attribut, das bei Suchvorgängen verwendet werden soll. Ist sie nicht angegeben, werden keine Benutzersuchen ausgeführt. Beispiel für die Definition dieser Eigenschaft:
api.custom.authentication.ldap.userattribute=cn

Falls eine erforderliche Eigenschaft fehlt, wird bei der Initialisierung eine Ausnahmebedingung ausgelöst.

Fehlt entweder der Wert von **api.custom.authentication.ldap.searchbase** oder der Wert von **api.custom.authentication.ldap.userattribute**, wird kein Suchkontext initialisiert und es werden während der eigentlichen Benutzerauthentifizierung keine Suchvorgänge ausgeführt. (Wenn keine Suchvorgänge ausgeführt werden, bedeutet dies, dass die Bindung an den LDAP-Server direkt unter Verwendung des zur Authentifizierung angegebenen Benutzernamens und Kennworts versucht wird.)

Wenn die Eigenschaft **api.custom.authentication.ldap.adminidn** angegeben ist, wird ein Suchkontext unter Verwendung einer "einfachen" Authentifizierung erstellt. Falls während der Initialisierung des Suchkontextes ein Fehler auftritt, schlägt die Initialisierung des LDAP-Authentifizierungsmoduls fehl und eine Ausnahmebedingung wird ausgelöst.

Wenn die Eigenschaft **api.custom.authentication.ldap.adminidn** nicht angegeben ist, wird unter Verwendung einer "anonymen" JNDI-Bindung ein Suchkontext erstellt.

Anmerkung: Falls der Suchkontext nicht unter Verwendung der Eigenschaft **api.custom.authentication.ldap.adminidn** initialisiert werden kann, schlägt die Authentifizierung sofort fehl. In diesem Fall wird nicht versucht, ein anonymes Binden auszuführen.

LDAP-Authentifizierungslogik:

Verwenden Sie die aufgeführten Pfade, um die Berechtigungsnachweise für die LDAP-Authentifizierung zu authentifizieren.

Bei jedem Versuch, einen Benutzer zu authentifizieren, werden an das LDAP-Authentifizierungsmodul der Benutzername und das Kennwort des zu authentifizierenden Benutzers übergeben. Die folgenden Authentifizierungspfade sind möglich:

- Beide Eigenschaften **api.custom.authentication.ldap.searchbase** und **api.custom.authentication.ldap.userattribute properties** sind angegeben:
 - Falls der für die Authentifizierung angegebene Benutzername mit dem Wert der Eigenschaft **api.custom.authentication.ldap.searchbase** endet, wird davon ausgegangen, dass ein vollständiger definierter Name angegeben wurde, und es wird keine Benutzersuche ausgeführt. Es wird direkt versucht, mit dem für die Authentifizierung angegebenen Benutzernamen und Kennwort eine Bindung an den LDAP-Server vorzunehmen. Falls die Bindung erfolgreich erstellt werden kann, gilt die Authentifizierung als erfolgreich, andernfalls gilt sie als fehlgeschlagen.
 - Endet der Benutzername nicht mit dem Wert der Eigenschaft **api.custom.authentication.ldap.searchbase**, wird für den bei der Initialisierung erstellten Suchkontext eine Suche ausgeführt, deren Suchbereich eine Unterverzeichnisstruktur ist. Die Suchabfrage lautet "(*<ldap-benutzer-id-attribut>=<benutzername>*)". Hierbei steht *ldap-benutzer-id-attribut* für den Wert der Eigenschaft **api.custom.authentication.ldap.userattribute** und *benutzername* für den zur Authentifizierung angegebenen Benutzernamen. Falls genau ein Suchergebnis zurückgegeben wird, wird eine Bindung an den LDAP-Server vorgenommen. Hierzu werden der definierte Name des zurückgegebenen Eintrags und das für die Authentifizierung angegebene Kennwort verwendet. Die Authentifizierung ist nur dann erfolgreich, wenn die Bindung an den LDAP-Server erfolgreich vorgenommen werden kann. In allen anderen Fällen gilt die Authentifizierung als fehlgeschlagen. Werden mehrere Suchergebnisse zurückgegeben, schlägt die Authentifizierung fehl.
- Mindestens eine der Eigenschaften **api.custom.authentication.ldap.searchbase** oder **api.custom.authentication.ldap.userattribute** ist nicht angegeben:
In diesem Fall werden keine Suchvorgänge ausgeführt und es wird direkt versucht, mit dem für die Authentifizierung angegebenen Benutzernamen und Kennwort eine Bindung an den LDAP-Server vorzunehmen. Falls die Bindung erfolgreich erstellt werden kann, gilt die Authentifizierung als erfolgreich, andernfalls gilt sie als fehlgeschlagen.

Unterstützung von LDAP-Gruppen:

Sie können Berechtigungen für eine Gruppe in der Benutzerregistry auf dieselbe Weise wie für einen Benutzer festlegen. Die aufgeführten Eigenschaften dienen zur Unterscheidung zwischen Benutzern und Gruppen.

Zur Vereinfachung der Verwaltung lässt IBM Security Directory Integrator zu, dass Berechtigungen für Gruppen genauso konfiguriert werden, wie sie für Benutzer konfiguriert werden. Sie können Berechtigungen in der Benutzerregistry mit derselben Syntax festlegen, die Sie für einen Benutzer verwenden würden. Tatsächlich unterscheidet die Benutzerregistry nicht, ob es sich bei einer Sicherheitsentität um eine Gruppe oder einen Benutzer handelt. Die Unterscheidung zwischen Benutzern und Gruppen wird während des Authentifizierungsprozesses getroffen.

Die Gruppenzugehörigkeit ist im LDAP-Verzeichnis konfiguriert, mit dessen Hilfe IBM Security Directory Integrator Benutzer authentifiziert. Falls ein Benutzer zu einer LDAP-Gruppe gehört, übernimmt der Benutzer bei seiner Authentifizierung automatisch alle Berechtigungen für diese Gruppe. Die Gruppenunterstützung ist standardmäßig inaktiviert und muss daher von Ihnen aktiviert werden.

Die folgenden Systemeigenschaften sind mit der Unterstützung von LDAP-Gruppen verbunden:

api.custom.authentication.ldap.groupsupport

Bei dieser optionalen Eigenschaft handelt es sich um ein boolesches Flag. Falls sie fehlt, wird der Standardwert "false" verwendet. Sie gibt an, ob die Gruppenzugehörigkeit bei der Authentifizierung von Benutzern aufgelöst wird. Falls die Gruppenzugehörigkeit aufgelöst wird, wird sie bei der Autorisierung berücksichtigt.

api.custom.authentication.ldap.usermembershipattribute

Diese Eigenschaft ist nur dann erforderlich, wenn die Eigenschaft **api.custom.authentication.ldap.groupsupport** auf "true" gesetzt ist. Sie gibt den Namen des Attributs eines Benutzers in LDAP an, das eine Liste der Gruppen enthält, zu denen der Benutzer gehört.

api.custom.authentication.ldap.usermembershipattributecontent

Diese Eigenschaft ist nur dann erforderlich, wenn die Eigenschaft **api.custom.authentication.ldap.groupsupport** auf "true" gesetzt ist. Sie gibt an, wie Gruppen im Zugehörigkeitsattribut eines Benutzers benannt sind. Enthält das Zugehörigkeitsattribut des Benutzers beispielsweise Werte, die den Attributen "objectSID" von Gruppen entsprechen, setzen Sie diese Eigenschaft auf den Wert "objectSID". Enthält das Zugehörigkeitsattribut des Benutzers definierte Namen von Gruppen, setzen Sie diese Eigenschaft auf "dn".

api.custom.authentication.ldap.groupnameattribute

Diese Eigenschaft ist nur dann erforderlich, wenn die Eigenschaft **api.custom.authentication.ldap.groupsupport** auf "true" gesetzt ist. Sie gibt den Namen des Attributs einer Gruppe in LDAP an, das dem Verfahren entspricht, mit dem die Gruppe in der IBM Security Directory Integrator-Benutzerregistry benannt wird. Falls LDAP-Gruppen in der IBM Security Directory Integrator-Registry beispielsweise durch ihren allgemeinen Namen angegeben werden, setzen Sie diese Eigenschaft auf den Wert "cn". Enthält die Benutzerregistry die definierten Namen der Gruppen, setzen Sie diese Eigenschaft auf den Wert "dn".

api.custom.authentication.ldap.groupsearchbase

Diese Eigenschaft ist nur dann erforderlich, wenn die Eigenschaft **api.custom.authentication.ldap.groupsupport** auf "true" gesetzt ist. Sie stellt den LDAP-Verzeichniskontext dar, in dem nach Gruppen gesucht wird.

api.custom.authentication.ldap.binaryattributes

Dies ist eine optionale Eigenschaft, die eine Liste von durch Leerzeichen getrennten Attributnamen enthält. Sie gibt Attribute mit einer Syntax ohne Zeichenfolge an.

Beispiel für Active Directory

Das folgende Beispiel zeigt, wie die Gruppenunterstützung so konfiguriert wird, dass sie bei einem **Active Directory**-Server verwendet werden kann:

```
api.custom.authentication.ldap.groupsupport=true
api.custom.authentication.ldap.usermembershipattribute=tokenGroups
api.custom.authentication.ldap.usermembershipattributecontent=objectSID
api.custom.authentication.ldap.groupnameattribute=SAMAccountName
api.custom.authentication.ldap.groupsearchbase=DC=mytestadsrver,DC=com
api.custom.authentication.ldap.binaryattributes=objectSID tokenGroups
```

Das Attribut "tokenGroups" ist ein berechnetes Attribut, das für alle Benutzer in Active Directory vorhanden ist.

Es enthält eine Sammlung der Sicherheits-IDs (SID) für alle Sicherheitsgruppen, zu denen ein Benutzer gehört.

Diese Sammlung enthält ausschließlich Sicherheitsgruppen (für E-Mail verwendete Verteilergruppen sind nicht enthalten) und sie enthält alle Sicherheitsgruppen inklusive verschachtelter Gruppen und Primärgruppen.

Die Sicherheits-IDs sind binäre Attribute und müssen daher in der Eigenschaft `api.custom.authentication.ldap.binaryattributes` festgelegt werden.

Im obigen Beispiel werden Gruppen nach ihrem LDAP-Attribut "sAMAccountName" in der IBM Security Directory Integrator-Benutzerregistry benannt.

IBM Security Directory Server - Beispiel

Das folgende Beispiel zeigt, wie die Gruppenunterstützung so konfiguriert wird, dass sie mit IBM Security Directory Server verwendet werden kann:

```
api.custom.authentication.ldap.groupsupport=true
api.custom.authentication.ldap.usermembershipattribute=ibm-allGroups
api.custom.authentication.ldap.usermembershipattributecontent=dn
api.custom.authentication.ldap.groupnameattribute=dn
api.custom.authentication.ldap.groupsearchbase=ou=mytestou,c=mytestcountry
```

Für einen gegebenen Benutzereintrag listet das aktive Attribut "ibm-allGroups" alle statischen, dynamischen und verschachtelten Gruppen auf, zu denen dieser Benutzer gehört.

Anmerkung:

1. IBM Security Directory Integrator ermittelt die Gruppenzugehörigkeit durch eine direkte Prüfung des LDAP-Benutzereintrags (statt die Zugehörigkeit durch ein Durchsuchen aller Gruppen indirekt festzustellen). Damit dieser Ansatz funktioniert, muss der Benutzereintrag ein Attribut aufweisen, in dem die Gruppen aufgelistet sind, zu denen der Benutzer gehört. Die Gruppenunterstützung kann nur bei LDAP-Servern verwendet werden, die bei jedem Benutzereintrag ein solches Zugehörigkeitsattribut unterstützen.
2. Falls Sie die Gruppenzugehörigkeit eines Benutzers modifizieren, hat dies auf bestehende Server-API-Sitzungen keinen Einfluss. Sitzungen, die nach der Modifizierung aufgebaut werden, berücksichtigen dies jedoch.
3. Die Gruppenunterstützung steht gegenwärtig nur für die LDAP-Authentifizierung zur Verfügung. Für die JAAS-Authentifizierung oder die Authentifizierung mit angepasstem JavaScript gibt es keine Gruppenunterstützung.
4. Wenn in der Server-API die SSL-Clientauthentifizierung aktiviert ist, müssen Clients, die keinen Benutzernamen angeben, anhand des Eigners des SSL-Clientzertifikats authentifiziert und autorisiert werden. Falls die LDAP-Authentifizierung mit Gruppenunterstützung (zusammen mit der SSL-Clientauthentifizierung) ebenfalls aktiviert ist, wird die Gruppenzugehörigkeit für den Eigner des SSL-Clientzertifikats aufgelöst.

Hostbasierte Authentifizierung

Konfigurieren Sie die hostbasierten Eigenschaften, um die hostbasierte Authentifizierung zu verwenden. Nachstehend erhalten Sie Informationen dazu.

Die hostbasierte Authentifizierung wird verwendet, wenn SSL durch Angabe von `api.remote.ssl.on=false` in den Dateien `global.properties` oder `solution.properties` inaktiviert ist. Die hostbasierte Authentifizierung wird unter Verwendung der Eigenschaft `api.remote.nonssl.hosts` konfiguriert. Diese Eigenschaft gibt die Liste der Host-IP-Adressen an, von denen aus Clients der fernen Server-API die Server-API ohne Angabe eines Benutzernamens und eines Kennworts verwenden können.

Diese Hostliste besteht aus einer Liste von IP-Adressen (Hostnamen werden nicht akzeptiert), zwischen denen als Begrenzungszeichen ein Leerzeichen, ein Komma oder ein Semikolon verwendet wird. Beispiel für den Wert dieser Eigenschaft:

```
api.remote.nonssl.hosts=192.168.111.222, 192.168.112.158
```

Wenn ein Client, der die hostbasierte Authentifizierung verwendet, erfolgreich authentifiziert wurde, wird ihm die Administratorberechtigung erteilt. Aus diesem Grund müssen Sie beim Hinzufügen von IP-Adressen zu dieser Liste äußerst sorgfältig vorgehen. Eine Verwendung der hostbasierten Authentifizierung in einer Produktionsumgebung ist aufgrund der mit ihr verbundenen Sicherheitsprobleme nicht empfehlenswert. Normalerweise wird die hostbasierte Authentifizierung bei der Entwicklung einer Lösung oder bei der Ausführung eines Demos eingesetzt.

Zusammenfassung der Optionen für die Server-API-Authentifizierung

Es gibt mehrere Optionen für die Server-API-Authentifizierung. Sie sind in der nachstehenden Liste zusammengefasst.

Die folgenden Authentifizierungsoptionen sind verfügbar:

SSL-basierte Authentifizierung (in IBM Security Directory Integrator 6.0 verfügbarer Mechanismus)

Diese Option funktioniert nur, wenn die Eigenschaft `api.remote.ssl.client.auth.on=true` angegeben ist (außerdem muss `api.on=true`, `api.remote.on=true`, `api.remote.ssl.on=true` angegeben sein). Der Benutzer wird mit der Berechtigung autorisiert, die der Benutzer-ID des SSL-Zertifikats in der Benutzerregistry der Server-API zugeordnet ist.

Anmerkung: Wenn SSL eingesetzt wird und die ferne Clientanwendung Listener-Objekte der Server-API verwendet, muss die Clientanwendung ein eigenes Zertifikat besitzen, das vom IBM Security Directory Integrator-Server anerkannt wird (analog zur Konfiguration für die SSL-Clientauthentifizierung). Gibt es kein vom IBM Security Directory Integrator-Server anerkanntes Clientzertifikat, funktionieren die Listener-Objekte nicht und die ferne Clientanwendung kann keine Benachrichtigungen vom IBM Security Directory Integrator-Server empfangen.

Auf Benutzername/Kennwort basierende Authentifizierung

Diese Option kann nur verwendet werden, wenn in der Eigenschaft `api.custom.authentication` eine JavaScript-Authentifizierungsdatei angegeben ist. Diese Authentifizierungsmethode funktioniert unabhängig davon, ob SSL und ob die SSL-Clientauthentifizierung verwendet wird. Der Benutzer wird mit der Berechtigung autorisiert, die dem Benutzer (angegeben mit dem Wert für `username`) in der Benutzerregistry der Server-API zugeordnet ist (siehe hierzu auch „Benutzerregistry der Server-API“ auf Seite 140).

LDAP-Authentifizierung

Diese Option ist unter „Unterstützung der LDAP-Authentifizierung“ auf Seite 130 beschrieben. Sie ist von einer Reihe von Einstellungen für `api.custom.authentication` in der Datei `global.properties` oder `solution.properties` abhängig.

Hostbasierte Authentifizierung

Diese Option kann nur verwendet werden, wenn die Eigenschaft `api.remote.ssl.on=false` angegeben ist. Bei ihrer Verwendung wird das Öffnen von Server-API-Sitzungen ohne Benutzername und Kennwort von allen Hosts, die in der Eigenschaft `api.remote.nonssl.hosts` angegeben sind, erfolgreich authentifiziert und den Sitzungen wird eine Administratorberechtigung

gung erteilt. Die Eigenschaft *api.remote.nonssl.hosts* kann in der Datei *global.properties* oder *solution.properties* angegeben werden.

JMX-Schicht der Server-API

Die JMX-Schicht der Server-API unterstützt die Authentifizierung über Benutzername und Kennwort nicht. Authentifizieren Sie die JMX-Schicht mit den hier aufgeführten Schritten.

Die ferne JMX-Schicht der Server-API unterstützt die auf Benutzername und Kennwort basierende Authentifizierung nicht. Sie ignoriert die Eigenschaften *api.custom.authentication*. Unabhängig davon, welchen Wert diese Eigenschaften aufweisen und ob die angepasste Authentifizierung für die Server-API aktiviert ist oder nicht, führt die ferne JMX-Schicht die folgende Authentifizierung durch:

- Falls SSL und auch die SSL-Clientauthentifizierung aktiviert ist, führt die ferne JMX-Schicht eine SSL-basierte Authentifizierung durch (wie in IBM Security Directory Integrator 6.0).
- Falls SSL aktiviert, die SSL-Clientauthentifizierung jedoch inaktiviert ist, kann die ferne JMX-Schicht nicht verwendet werden.
- Falls SSL inaktiviert ist, wird der ferne JMX-Client nur dann erfolgreich authentifiziert, wenn sein Host in der Eigenschaft *api.remote.nonssl.hosts* angegeben ist, also von der hostbasierten Authentifizierung ausgegangen wird. In diesem Fall wird dem Client die Administratorberechtigung erteilt.

Zusammenfassend lässt sich festhalten, dass die JMX-Schicht der Server-API die Authentifizierung über Benutzername und Kennwort nicht unterstützt.

Beispiele für Konfiguration der Server-API-Authentifizierung

Sie können die aufgeführte Liste mit Beispielen für eine Server-API-Authentifizierung durchsehen. Dies hilft beim Konfigurieren des Servers.

Nachfolgend sind einige Beispiele für die Konfiguration der Authentifizierung dargestellt.

1. Konfiguration ohne SSL und angepasste Authentifizierung:

```
api.remote.ssl.on=false
api.remote.nonssl.hosts=192.168.113.51, 192.168.113.52
api.custom.authentication=ldap_auth.js
```

SSL wird nicht verwendet.

- Authentifizierungsanforderungen ohne Angabe von Benutzername und Kennwort sind nur dann erfolgreich, wenn sie über "localhost" bzw. die Adresse "192.168.113.51" oder "192.168.113.52" aufgerufen werden.
- Authentifizierungsanforderungen mit angegebenem Benutzernamen und Kennwort sind nur dann erfolgreich, wenn das Script "ldap_auth.js" den Benutzer, der mit den Parametern für den Benutzernamen und das Kennwort angegeben wurde, erfolgreich authentifiziert.
- Ferne JMX-Clients werden nur dann authentifiziert, wenn die Anforderung von "localhost" bzw. der Adresse "192.168.113.51" oder "192.168.113.52" stammt.

2. SSL (ohne Clientauthentifizierung) und angepasste Authentifizierung:

```
api.remote.ssl.on=true
api.remote.ssl.client.auth.on=false
api.custom.authentication=ldap_auth.js
```

SSL wird für die Kommunikation mit der fernen Server-API verwendet.

- Authentifizierungsanforderungen ohne Angabe von Benutzername und Kennwort schlagen fehl, weil weder die SSL-Clientauthentifizierung noch die hostbasierte Authentifizierung aktiviert ist.
 - Authentifizierungsanforderungen mit angegebenem Benutzernamen und Kennwort sind nur dann erfolgreich, wenn das Script "ldap_auth.js" den Benutzer, der mit den Parametern für den Benutzernamen und das Kennwort angegeben wurde, erfolgreich authentifiziert.
 - Die hostbasierte Authentifizierung ist in diesem Fall unabhängig vom Wert des Parameters "api.remote.nonssl.hosts" nicht verfügbar, da "api.remote.ssl.on" auf "true" gesetzt ist.
 - Die ferne JMX-Schicht ist nicht zugänglich. Dies liegt daran, dass SSL aktiviert ist, die SSL-Clientauthentifizierung jedoch nicht verwendet wird.
3. SSL mit Clientauthentifizierung und angepasste Authentifizierung:

```
api.remote.ssl.on=true
api.remote.ssl.client.auth.on=true
api.custom.authentication=ldap_auth.js
```

SSL wird für die Kommunikation mit der fernen Server-API verwendet. Der Server erfordert die SSL-Clientauthentifizierung.

- Authentifizierungsanforderungen ohne Angabe von Benutzername und Kennwort sind erfolgreich, wenn das SSL-Zertifikat des Clients im Truststore des Servers vorhanden ist (oder anhand der Zertifikate im Truststore überprüft werden kann).
 - Authentifizierungsanforderungen mit angegebenem Benutzernamen und Kennwort sind nur dann erfolgreich, wenn die SSL-Clientauthentifizierung erfolgreich ist (das SSL-Zertifikat des Clients also im Truststore des Servers vorhanden ist) und das Script "ldap_auth.js" den Benutzer, der mit den Parametern für den Benutzernamen und das Kennwort angegeben wurde, erfolgreich authentifiziert. In diesem Fall wird die Autorisierung basierend auf dem Parameter für den Benutzernamen (aus dem angegebenen Benutzernamen und Kennwort) und nicht anhand der Benutzeridentität aus dem SSL-Clientzertifikat vorgenommen.
 - Die hostbasierte Authentifizierung ist in diesem Fall unabhängig vom Wert des Parameters *api.remote.nonssl.hosts* nicht verfügbar, da *api.remote.ssl.on* auf "true" gesetzt ist.
 - Ferne JMX-Clients werden authentifiziert, wenn das SSL-Zertifikat des Clients im Truststore des Servers vorhanden ist (oder anhand der Zertifikate im Truststore überprüft werden kann).
4. SSL mit Clientauthentifizierung und ohne angepasste Authentifizierung:

```
api.remote.ssl.on=true
api.remote.ssl.client.auth.on=true
api.custom.authentication=
```

(Alternativ kann es sein, dass die Eigenschaft *api.custom.authentication* gänzlich fehlt.)

SSL wird für die Kommunikation mit der fernen Server-API verwendet. Der Server erfordert die SSL-Clientauthentifizierung.

- Authentifizierungsanforderungen ohne Angabe von Benutzername und Kennwort sind erfolgreich, wenn das SSL-Zertifikat des Clients im Truststore des Servers vorhanden ist (oder anhand der Zertifikate im Truststore überprüft werden kann).
- Authentifizierungsanforderungen mit angegebenem Benutzernamen und Kennwort sind nicht erfolgreich, weil die angepasste Authentifizierung nicht konfiguriert ist.

- Die hostbasierte Authentifizierung ist in diesem Fall unabhängig vom Wert des Parameters *api.remote.nonssl.hosts* nicht verfügbar, da *api.remote.ssl.on* auf "true" gesetzt ist.
- Ferne JMX-Clients werden nur dann erfolgreich authentifiziert, wenn das SSL-Zertifikat des Clients im Truststore des Servers vorhanden ist.

Server-API-Autorisierung

Mit den hier aufgeführten Anweisungen können Sie einem Benutzer eine Autorisierung zuweisen.

Nachdem die Anforderung eines Clients für eine Server-API-Sitzung authentifiziert wurde, muss sie autorisiert werden.

Benutzer der fernen API können verschiedene Rollen zugeordnet werden. Eine Rolle definiert die Liste der Server-API-Aufrufe, die vom Benutzer ausgeführt werden können, und auch den Kontext, in dem diese Aufrufe ausgeführt werden können. Eine Methode der Server-API kann ausgeführt werden, wenn dem Benutzer mindestens eine Rolle zugeordnet ist, die die Ausführung dieser Methode in dem Kontext zulässt, in dem der Benutzer die Methode auszuführen versucht. Beispielsweise kann eine Rolle die Benutzerberechtigung lediglich für die Ausführung bestimmter Fertigungslinien aus einer bestimmten Konfiguration erteilen. Details über die Erstellung der Datei, in der diese Benutzerberechtigungen definiert sind, finden Sie unter „Benutzerregistry der Server-API“ auf Seite 140.

Die Autorisierung basiert auf der Benutzer-ID. Abhängig vom verwendeten Authentifizierungsmechanismus werden zum Abrufen der Benutzer-ID unterschiedliche Verfahren verwendet:

- SSL-basierte Authentifizierung - Die Benutzer-ID ist der definierte Name (DN) im SSL-Zertifikat des Clients.
- Auf Benutzername oder Kennwort basierende Authentifizierung - Die Benutzer-ID ist der Benutzername, der im Paar aus Benutzernamen und Kennwort angegeben ist.
- Hostbasierte Authentifizierung - Bei Verwendung dieses Authentifizierungsmechanismus kann keine Benutzer-ID aus dem Client abgerufen werden. In diesem Fall wird die Clientsitzung mit der Rolle *admin* autorisiert.

Berechtigungsrollen

Sie können einen Benutzer mit mehreren Rollen autorisieren. Nachstehend finden Sie eine Liste mit den Rollen, die für das Sicherheitsmodell der Server-API gelten:

Benutzern der fernen API werden unterschiedliche Rollen zugeordnet. Eine Rolle definiert die Liste der Server-API-Aufrufe, die vom Benutzer ausgeführt werden können, und auch den Kontext, in dem diese Aufrufe erfolgen werden können. Beispielsweise kann eine Rolle die Benutzerberechtigung lediglich für den Aufruf bestimmter Fertigungslinien aus einer bestimmten Konfiguration erteilen.

Einem Benutzer können mehrere Rollen zugeordnet sein. Hierbei ist auch die mehrmalige Zuordnung derselben Rolle mit unterschiedlichen Parametern möglich. Eine Methode der Server-API kann aufgerufen werden, wenn dem Benutzer mindestens eine Rolle zugeordnet ist, die die Ausführung dieser Methode in dem Kontext zulässt, in dem der Benutzer die Methode auszuführen versucht.

Es gibt keine Semantik für eine Zurückweisung. Dies bedeutet, dass Aktionen nicht explizit ausgeschlossen werden können. Die folgenden Rollen gelten für das Sicherheitsmodell der Server-API:

<p>Rolle <i>Lesen</i>: read [list_of(configuration)]</p>	<p>Die Rolle <i>Lesen</i> ermöglicht es dem Benutzer, Daten aus der bzw. den Konfiguration(en) des Servers zu lesen.</p> <p>Falls keine Liste mit Konfigurationen angegeben oder die Liste leer ist, darf der Benutzer Daten aus keiner Konfiguration lesen.</p> <p>Für die Liste der Konfigurationen kann der Sonderwert * (Stern) angegeben werden. Dies bedeutet, dass der Benutzer (über Server-API-Aufrufe) alle Konfigurationen lesen darf, die gegenwärtig durch den Server geladen sind.</p> <p>Wenn die Liste der Konfigurationen angegeben bzw. nicht leer ist und der Sonderwert * nicht angegeben ist, darf der Benutzer ausschließlich die angegebenen Konfigurationen lesen.</p> <p>Die Rolle <i>Lesen</i> beinhaltet keine Berechtigung zum Starten von Prozessen (Fertigungslinien) oder zum Anwenden von Änderungen auf den Server und dessen Konfigurationen. Beispiel:</p> <pre>[ROLE]:read [CONFIG]:*</pre>
<p>Rolle <i>Ausführen</i>: execute [list_of(configuration [list_of(AssemblyLines)])]</p>	<p>Die Rolle <i>Ausführen</i> erteilt dem Benutzer die Berechtigung zum Ausführen von Fertigungslinien.</p> <p>Falls keine Liste mit Konfigurationen angegeben oder die Liste leer ist, darf der Benutzer keine Fertigungslinien aus keiner Konfiguration ausführen.</p> <p>Für die Liste der Konfigurationen kann der Sonderwert * (Stern) angegeben werden. Dies bedeutet, dass der Benutzer (über Server-API-Aufrufe) alle Fertigungslinien aus allen Konfigurationen ausführen darf.</p> <p>Wenn die Liste der Konfigurationen vorhanden und der Sonderwert * nicht angegeben ist, darf der Benutzer ausschließlich die Prozesse aus den in der Liste angegebenen Konfigurationen starten. Für jede in der Liste angegebene Konfiguration gilt Folgendes:</p> <ul style="list-style-type: none"> • Falls keine Liste mit Fertigungslinien vorhanden ist, darf der Benutzer keine Fertigungslinie aus dieser Konfiguration ausführen. • Falls für die Liste der Fertigungslinien der Sonderwert * (Stern) angegeben ist, darf der Benutzer alle Fertigungslinien aus dieser Konfiguration ausführen. • Falls die Liste der Fertigungslinien vorhanden und der Sonderwert * nicht angegeben ist, darf der Benutzer ausschließlich die in der Liste angegebenen Fertigungslinien ausführen. <p>Beispiel:</p> <pre>[ROLE]:execute [CONFIG]:C:/TDI/rs.xml [AL]:* [CONFIG]:C:/TDI/prototype.xml [AL]:TestAssemblyLine</pre>

Rolle <i>Administrator</i> : admin	<p>Die Rolle <i>Administrator</i> ermöglicht dem Benutzer das Ausführen aller Server-API-Aufrufe in jedem möglichen Kontext.</p> <p>Ein Benutzer mit der Rolle <i>Administrator</i> darf alle Konfigurationen lesen und modifizieren, neue Konfigurationen laden, Fertigungslinien ausführen sowie Serverparameter lesen und modifizieren.</p> <p>Beispiel: [ROLE]:admin</p> <p>Anmerkung:</p> <p>Die Rolle "Administrator" ist für die Verwendung des fernen Konfigurationseditors erforderlich. Lesen Sie hierzu auch die Angaben im Abschnitt „Fernes Konfigurationseditor verwenden“ auf Seite 160.</p>
---	--

Bei den in einem Tag [CONFIG] angegebenen Werten kann es sich entweder um die Namen von Konfigurationsdateien oder um Namen von Lösungen handeln, falls diese in der Konfigurationsdatei angegeben sind.

Benutzerregistry der Server-API

Die Benutzerregistrydatei verschlüsselt das Serverzertifikat. Nachstehend erhalten Sie Informationen zu ihrer Struktur.

Die Benutzerregistry (angegeben durch die Eigenschaft *api.user.registry* in der Datei *global.properties* oder *solution.properties*) ist eine Textdatei, in der die Informationen zu allen Benutzern der API und deren Rollen verwaltet werden. Diese Datei ist mit dem Zertifikat des Servers verschlüsselt, das in der Eigenschaft *api.key.alias* aus dem in der Eigenschaft *api.keystore* angegebenen Schlüsselspeicher festgelegt ist. Als Verschlüsselungsalgorithmus wird die asymmetrische RSA-Verschlüsselung oder -Entschlüsselung angewendet. Aus diesem Grund wird bei dem unter „Zertifikate für Web-Service-Suite von IBM Security Directory Integrator“ auf Seite 167 beschriebenen Prozess der RSA-Algorithmus angegeben. Dies ist der Standardalgorithmus des im Abschnitt „Verschlüsselungsdienstprogramm von IBM Security Directory Integrator“ auf Seite 153 beschriebenen und mit IBM Security Directory Integrator bereitgestellten Dienstprogramms, das Sie zu diesem Zweck verwenden können. Beim Start entschlüsselt die Server-API-Steuerkomponente diese Datei und liest sie in die eigenen Hauptspeicherstrukturen ein.

Anmerkung:

1. Die gesamte Benutzerregistrydatei wird wie vorhanden blockweise und direkt mit dem RSA-Algorithmus und dem öffentlichen Schlüssel des Servers verschlüsselt. Es wird weder eine digitale Signatur noch eines der Hashverfahren verwendet.
2. Die Autorisierung anhand der Benutzerregistry ist kein optionales Verfahren. Gegenwärtig wird für den IBM Security Directory Integrator-Server kein Konzept verwendet, das einen Autorisierungsmechanismus über Plug-ins nutzt.

Der Inhalt der Textdatei mit der Identitätsregistry ist folgendermaßen strukturiert:

```
[USER]
[ID]:<benutzer-id>
[ROLE]:<rollen-id>
  [CONFIG]:<konfigurations-id>
    [AL]:<name_der_fertigungslinie>
    [AL]:<name_der_fertigungslinie>
    ...
  [CONFIG]:<konfigurations-id>
  ...
```

```

[ROLE]:<rollen-id>
...
[ROLE]:<rollen-id>
...
[ENDUSER]

[USER]
[ID]:<benutzer-id>
[ROLE]:<rollen-id>
...
[ENDUSER]
...

```

Jeder Tag darf sich nur über eine einzige Zeile erstrecken und für jeden Tag muss eine separate Zeile verwendet werden. Tabulator- und Leerzeichen werden nicht berücksichtigt. Leere Zeilen können an beliebiger Stelle verwendet werden. Die Tags in der Identitätsregistrydatei und deren Argumente lauten wie folgt:

Tag	Argument
[USER]	Dieser Tag verwendet keine Argumente und dient als Öffnungstag für die nachfolgenden Tags. Ein Paar aus den Tags [USER] und [ENDUSER] (jeweils in einer separaten Zeile) stellt die Definition eines einzelnen Benutzers in der Registrydatei bereit. Es kann mehrere Paare dieses Typs geben. Jedes Paar gibt einen Benutzer der Server-API an.
[ID]:<benutzer-id>	Dieser Tag ist der erste Tag nach dem Tag [USER]. Sein Argument <benutzer-id> gibt die eindeutige Kennung des Benutzers der Server-API an. Dieser ID-Wert ist der Wert 118 aus der Truststore-Datei. Der Tag und das Argument des Tags müssen in einer einzigen Zeile angegeben werden. Jedes Paar aus den Tags [USER] und [ENDUSER] darf nur einen einzigen Tag [ID]: enthalten.
[ROLE]:<rollen-id>	Dieser Tag gibt die Rolle für den Benutzer an. Mögliche Rollen sind read (Lesen), execute (Ausführen) oder admin (Administrator). Alle Angaben nach dem Tag [ROLE]: und seinem Argument sowie vor einem anderen Tag [ROLE]: oder einem Tag [ENDUSER] (je nachdem, welcher Tag zuerst auftritt) geben Details für diese Benutzerrolle an. Der Tag und das Argument des Tags müssen in einer einzigen Zeile angegeben werden. Jedes Paar aus den Tags [USER] und [ENDUSER] kann mehrere Tags [ROLE]: enthalten und somit mehrere Rollen für diesen Benutzer angeben.
[CONFIG]:<konfigurations-id>	Dieser Tag gibt die Kennung einer IBM Security Directory Integrator-Konfiguration in Form des absoluten Dateipfades der Konfiguration an. Relative Dateipfade werden nicht erkannt. Dieser Tag ist einem Tag [ROLE]: untergeordnet. Er gibt eine Konfiguration für die im Tag [ROLE]: angegebene Rolle an. Dieser Tag und sein Argument müssen in einer einzigen Zeile angegeben werden. Es kann mehrere Tags [CONFIG]: geben, die alle zum übergeordneten Tag [ROLE]: gehören. Falls einem Tag [ROLE]: kein Tag [CONFIG]: zugeordnet ist, ist die Liste der Konfigurationen für die entsprechende Rolle leer.
[AL]:<name_der_fertigungslinie>	Dieser Tag gibt den Namen einer Fertigungslinie an. Er ist einem Tag [CONFIG]: untergeordnet. Der Tag und sein Argument müssen in einer einzigen Zeile angegeben werden. Es kann mehrere Tags [AL]: geben, die alle zum übergeordneten Tag [CONFIG]: gehören. Falls einem Tag [CONFIG]: kein Tag [AL]: zugeordnet ist, ist die Liste der Fertigungslinien für die entsprechende Konfigurations-ID leer.

Der folgende Text ist ein Beispiel für eine Identitätsregistrydatei:

```

[USER]
[ID]:CN=Stan, OU=TDI, O=IBM, C=US
[ROLE]:admin
[ENDUSER]

[USER]
[ID]:CN=John, OU=TDI, O=IBM, C=US

```

```

[ROLE]:read
      [CONFIG]:*
[ROLE]:execute
      [CONFIG]:C:/TDI/rs.xml
      [AL]:*
      [CONFIG]:C:/TDI/prototype.xml
      [AL]:TestAssemblyLine
[ENDUSER]

[USER]
[ID]:CN=Peter, OU=TDI, O=IBM, C=US
[ROLE]:execute
      [CONFIG]:C:/TDI/rs.xml
      [AL]:*
[ENDUSER]

```

Diese Gruppe von Identitätsregistry-Einträgen impliziert die folgenden Bedingungen:

- Der Benutzer "Stan" ist gemäß dieser Registrydatei ein Administrator und darf alle Operationen der Server-API ausführen.
- Der Benutzer "John" darf alle in den Server geladenen Konfigurationen lesen, jedoch nur Prozesse aus zwei Konfigurationen ausführen:
 - Er darf alle Fertigungslinien aus der Konfiguration `rs.xml` ausführen.
 - Er darf aus der Konfiguration `prototype.xml` nur die Fertigungslinie namens "TestAssemblyLine" ausführen.
- Der Benutzer "Peter" darf lediglich alle Fertigungslinien aus der Konfiguration `rs.xml` ausführen.

Anmerkung: Mit dem Dienstprogramm **keytool** und/oder **Ikeyman** kann die Benutzer-ID aus der Truststore-Datei abgerufen werden. Die folgende Befehlszeile gibt alle Benutzer aus der Truststore-Datei aus:

```
keytool -v -list -keystore <truststore-datei> -storepass <truststore-kennwort>
```

Hierbei steht `<truststore-datei>` für die Schlüsselspeicherdatei, die die Zertifikate aller vertrauenswürdigen Benutzer enthält, und `<truststore-kennwort>` für das Kennwort dieser Schlüsselspeicherdatei. Die folgende Befehlszeile gibt für jedes Benutzerzertifikat einen Text ähnlich dem Folgenden aus:

```

Owner: CN=Stan, OU=TDI, O=IBM, C=US
Issuer: CN=Stan, OU=TDI, O=IBM, C=US
Serial number: 408f6a34
Valid from: 4/28/04 11:24 AM until: 7/27/04 11:24 AM
Certificate fingerprints:
    MD5:  F6:EF:81:8B:4C:0F:10:E4:A0:16:99:AB:42:29:70:8B
    SHA1: FE:37:62:8B:42:2F:54:F8:F6:F3:FC:A1:DD:7D:2A:51:9A:85:09:02

```

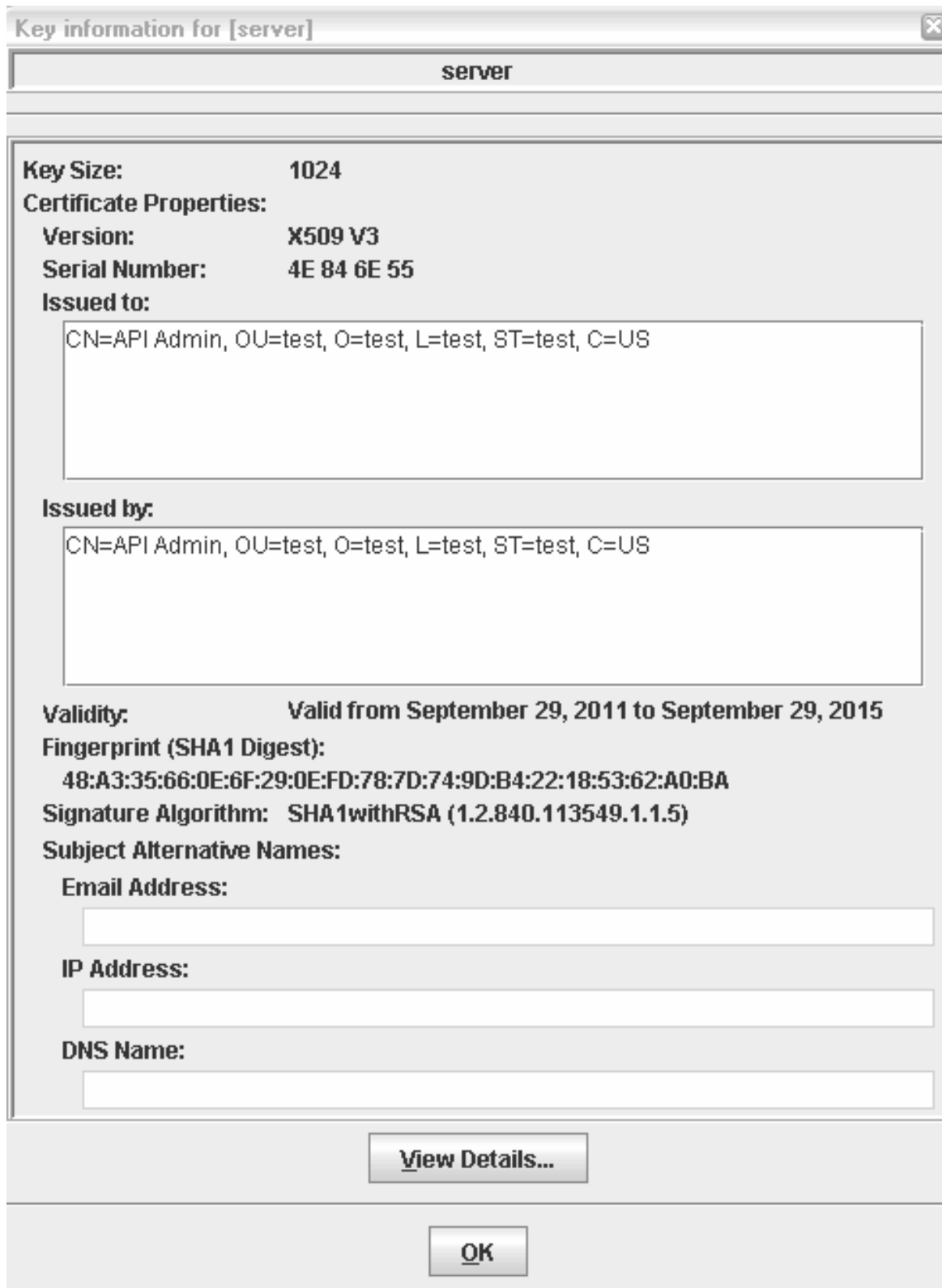
Der im Feld **Owner** vorhandene Wert **muss** wie gezeigt (also inklusive aller Leerzeichen und Kommas) als Wert für den Tag [ID]: in der Identitätsregistry angegeben werden. Bei diesem Beispiel sieht die Zeile für den ID-Tag folgendermaßen aus:

```
[ID]:CN=Stan, OU=TDI, O=IBM, C=US
```

Alternativ kann zum Abrufen der Benutzer-ID aus der Truststore-Datei auch das Tool "Ikeyman" wie folgt verwendet werden:

1. Starten Sie Ikeyman (oder wählen Sie in der Funktionsleiste das Symbol für den **Schlüsselmanager** aus).
2. Klicken Sie im Menü **Schlüsseldatenbankdatei** auf **Öffnen...**
3. Lesen Sie im Feld **Öffnen** die entsprechenden Werte fest und klicken Sie auf **OK**.
4. Geben Sie im Feld **Kennwort** das Kennwort für die Truststore-Datei ein.
5. Klicken Sie auf das relevante Zertifikat.

6. Klicken Sie auf die Schaltfläche **Anzeigen/Bearbeiten...**. Daraufhin wird ein Fenster geöffnet, das Informationen zum definierten Namen (also der Benutzer-ID) des Eintrags enthält.



Prüffunktionen des Servers

Sie können IBM Security Directory Integrator-Ereignisse prüfen. Für jedes Ereignis werden Benachrichtigungen erstellt. Nachstehend erhalten Sie Informationen zu Prüffunktionen.

Mit der IBM Security Directory Integrator-Prüfkomponente kann der IBM Security Directory Integrator-Server Ereignisse wie beispielsweise die Authentifizierung und Autorisierung in der Server-API prüfen.

Wenn Authentifizierungs- und Autorisierungsereignisse (auth*) stattfinden, werden Benachrichtigungen generiert. Die Prüfdaten werden als Paket in einem Eintrag zusammengefasst und als Benutzerdaten in der Benachrichtigung zur Verfügung gestellt. Der "Prüfservice" besteht aus einer separaten Prüfkongfiguration, die vom IBM Security Directory Integrator-Server automatisch geladen wird. Die Prüfkongfiguration enthält automatisch gestartete Prüffertigungslinien. Die Prüffertigungslinien iterieren am Benachrichtigungsconnector unter Verwendung von passenden Filtern. IBM Security Directory Integrator-Benutzer können sogar benutzerdefinierte Benachrichtigungen erstellen, wenn aus eigenem Code heraus Prüfereignisse erstellt werden sollen.

Die IBM Security Directory Integrator-Prüfung besteht aus zwei Hauptteilen:

- Verfahren zur Generierung der erforderlichen Prüfinformationen
- "Prüfservice" zur Verarbeitung vorhandener Prüfdaten

Die Generierung der erforderlichen Prüfinformationen wird durch die Erstellung von IBM Security Directory Integrator-Einträgen an jedem Prüfpunkt in der Server-API und durch das Broadcasting dieser in eine Benachrichtigung eingeschlossenen Einträge implementiert. Zu diesem Zweck enthält die Server-API eine neue Klasse (`com.ibm.di.api.APIAuditor`), die den Eintrag generiert, den Eintrag als Benutzerdaten an eine Benachrichtigung anhängt und ihn an alle relevanten Listener sendet.

Der "Prüfservice" ist der Hauptkonsument der Prüfbenachrichtigungen. Der Prüfservice ist eine Konfiguration aus mehreren, am Benachrichtigungsconnector iterierenden Fertigungslinien. Durch die Verwendung unterschiedlicher Filter kann eine Vielzahl von Benachrichtigungstypen registriert werden.

Geltungsbereich der Prüfung

Nur Ereignisse, die den aufgeführten Kriterien entsprechen, können für die Prüfung in Betracht gezogen werden.

Die Prüffunktion von IBM Security Directory Integrator verfolgt lediglich die von Benutzern ausgeführten Aktionen und nicht generell die Serverereignisse. Ein für die Ausführung einer Task (Fertigungslinie stoppen) autorisierter Benutzer ist nicht dasselbe wie die tatsächlich ausgeführte Task (Fertigungslinie wird beendet). Die Autorisierung ist ein Autorisierungsereignis und die Ausführung einer zulässigen Aktion (z. B. das Stoppen einer Fertigungslinie) ist ein Serverereignis. Wenn ein Benutzer das Stoppen einer Fertigungslinie anfordert und die Fertigungslinie beendet wird, entsteht ein Paar aus Autorisierungsereignis und Serverereignis. Ein Serverereignis kann zu anderen Zeitpunkten auch eigenständig auftreten, beispielsweise dann, wenn eine Fertigungslinie normal beendet wird. Nur Ereignisse, die eine direkte Benutzerinteraktion einschließen, werden geprüft. Dies begrenzt die Standardprüfpunkte für Authentifizierungs- und Autorisierungsereignisse innerhalb der Server-API. Fast jede durch die Server-API zugänglich gemachte Methode wird durch einen eigenen Autorisierungscode geschützt. Die Prüfkomponente versucht nicht, für alle Autorisierungsereignisse Benachrichtigungen zu senden, sondern

wählt eine angemessene Teilmenge der durch Autorisierung geschützten Server-API-Methoden aus. Der Auswahl liegt das Prinzip zugrunde, dass alle Ereignisse überwacht werden, die folgende Bedingungen erfüllen:

- Protokolle oder Tombstones werden gelöscht.
- IBM Security Directory Integrator-Entitäten wie Konfigurationen, Fertigungslinien und der Server werden gestartet oder gestoppt.
- Konfiguration der Konfigurationsinstanz wird ersetzt (Konfiguration der Konfigurationsinstanz oder Check-in-Konfiguration wird ersetzt).
- Änderung kritischer IBM Security Directory Integrator-Daten durch Benutzer ist zulässig (Festlegen externer Eigenschaften, Übergeben einer Nachricht an die Systemwarteschlange, Aufrufen von angepasstem Java-Code in der JVM von IBM Security Directory Integrator).

Benachrichtigungen unterdrücken

Nachstehend erhalten Sie Informationen zur Unterdrückung von Benachrichtigungen sowie zu den Befehlen und Methoden zum Generieren der unterdrückten Ereignistypen.

Die Server-API von IBM Security Directory Integrator erlaubt zur Leistungsverbesserung das Unterdrücken bestimmter Benachrichtigungstypen. Das Benachrichtigungsframework leitet unterdrückte Ereignisse nicht weiter. Falls Sie versuchen, ein Ereignis mit einem *unterdrückten* Typ zu übertragen, gibt die Server-API keinen Fehler aus. Das unterdrückte Ereignis kann jedoch keinen der registrierten Listener für Benachrichtigungen erreichen. Die Liste der unterdrückten Ereignistypen wird durch die folgende Systemeigenschaft konfiguriert:

```
api.notification.suppress
```

Standardmäßig sind alle Authentifizierungs- und Autorisierungsereignisse unterdrückt:

```
api.notification.suppress=di.server.api.authenticate di.server.api.authorize*
```

Die Ereignistypen in der Liste werden durch Leerzeichen voneinander getrennt. Platzhalterzeichen für Übereinstimmungen mit mehreren Ereignistypen sind zulässig. Falls die Eigenschaft für den Ereignistyp fehlt oder leer ist, werden keine Ereignisse unterdrückt. Zur Unterdrückung aller angepassten Benachrichtigungen können Sie Folgendes eingeben:

```
api.notification.suppress=user
```

Anmerkung: Die Unterdrückung gilt für den gesamten IBM Security Directory Integrator-Server und kann dazu führen, dass alle Arten von Benachrichtigungen unterdrückt werden. Es ist möglich, dass sogar integrierte Benachrichtigungen (beispielsweise über das Starten einer Fertigungslinie oder das Beenden des Servers) unterdrückt werden. Eine unsachgemäße Verwendung der Unterdrückungsfunktion kann die Arbeit von Komponenten beeinträchtigen, die für Benachrichtigungen empfangsbereit sind (z. B. Tombstone Manager oder der Serverbenachrichtigungsconnector).

Benachrichtigungen senden

Sie können jedem registrierten Listener eine Benachrichtigung zustellen. Nachstehend finden Sie eine Liste mit den Parametern für die Zustellung von Benachrichtigungen.

Zum Senden von Benachrichtigungen wird eine Methode in "com.ibm.di.api.API-Engine" verwendet:

```
public static void sendNotification (String type, String id, Object data, String configInstanceId)
```

Diese Methode erstellt ein DIEvent-Element. Hierdurch wird eine Benachrichtigung an jeden Listener zugestellt, der für den Empfang des jeweiligen Benachrichtigungstyps registriert ist. Es gibt folgende Parameter für die Zustellung von Benachrichtigungen:

Tabelle 18. Parameter für Zustellung von Benachrichtigungen

Parametername	Definition
type	Typ des Benachrichtigungsereignisses
id	ID des Benachrichtigungsereignisses
data	Benutzerdatenobjekt (UserData) des Benachrichtigungsereignisses in Form eines Java-Objekts mit zusätzlichen Informationen
configInstanceId	ID der Benachrichtigungskonfigurationsinstanz (ConfigInstance), an die die Benachrichtigung gebunden ist

Die Methode von "com.ibm.di.api.APIEngine" löst eine Ausnahmebedingung "DIException" aus, falls der Parameter "type" null ist. Aufrufe der Methode können auf folgenden Wegen erfolgen:

- Lokal aus der JVM des IBM Security Directory Integrator-Servers. Dieser Zugriffstyp schließt die Skripterstellung in Fertigungslinienhooks ein und verwendet auch die API aus den neuen Connectors, die in Java und im IBM Security Directory Integrator-Server implementiert sind.
- Über Fernzugriff per RMI (Remote Method Invocation) von einer anderen JVM aus (auf dem lokalen oder einem fernen Netzcomputer). Dieser Zugriffstyp verwendet Lösungen, die folgende Merkmale aufweisen:
 - Verbindungen werden über Fernzugriff zu IBM Security Directory Integrator hergestellt.
 - Prozesse werden in IBM Security Directory Integrator verwaltet.
 - Geschäftslogik wird aufbauend auf IBM Security Directory Integrator erstellt.
 - Es handelt sich um Anwendungen, die nur für IBM Security Directory Integrator dediziert sind.
 - Es handelt sich um Anwendungen, die IBM Security Directory Integrator für einige ihrer Zielsetzungen verwenden.

Sicherheit für IBM Security Directory Integrator-Serverinstanz

Sie können eine Serverinstanz mithilfe von Verschlüsselungsalgorithmen und verschiedenen Konfigurationsdateien einrichten. Nachstehend erhalten Sie Informationen dazu.

In diesem Abschnitt wird nicht auf die speziellen Aspekte des (IBM Security Directory Integrator-basierten oder anderweitigen) Clientzugriffs auf einen IBM Security Directory Integrator-Server eingegangen, die im Abschnitt „Ferne Server-API“ auf Seite 119 erläutert sind. Stattdessen werden der verwendete Verschlüsselungsalgorithmus sowie die verschiedenen Konfigurationsdateien behandelt, die zur Einrichtung einer Serverinstanz benötigt werden.

Der IBM Security Directory Integrator-Server erfordert einen Schlüsselspeicher, der sowohl seinen privaten Schlüssel als auch das zugehörige Zertifikat und den öffentlichen Schlüssel für die PKI-Verschlüsselung von Konfigurationsdateien, von Eigenschaften in Eigenschaftendateien, von Dateien der Serverbenutzerregistry und anderen Objekten sowie für die SSL-Kommunikation enthält.

Die Systemeigenschaften *api.keystore* und *api.key.alias* geben den Schlüsselspeicher und den Schlüsselaliasnamen für den Schlüssel/das Zertifikat des Servers im

Schlüsselspeicher an. Das Kennwort des Schlüsselspeichers und das Kennwort des eigentlichen Schlüssels (das nicht mit dem Schlüsselspeicherkey identisch ist) sind in der Stashdatei des Servers angegeben. Der Zugriff auf einen Schlüsselspeicher wird durch ein Kennwort geschützt, das bei der Erstellung des Schlüsselspeichers durch die Person definiert wird, die den Schlüsselspeicher erstellt, und das nur nach Angabe des aktuellen Kennworts geändert werden kann. Zusätzlich kann jeder private Schlüssel in einem Schlüsselspeicher durch ein eigenes Kennwort geschützt werden. Weitere Informationen zur Stashdatei des Servers enthält der Abschnitt „Stashdatei“.

Zur Verschlüsselung von Dateien und Eigenschaftswerten wird der RSA-Algorithmus verwendet. Er wird als Blockchiffrierung eingesetzt, bei der die Blockgröße anhand der Moduluskomponente des RSA-Schlüssels bestimmt wird. Die Verschlüsselung erfolgt im Modus von ECB (Electronic Codebook). Die PKCS#1-Auffüllung wird auf jeden Block separat angewendet. Bitte beachten Sie, dass dasselbe RSA-Schlüsselpaar, das für die Verschlüsselung von Dateien verwendet wird, auch bei der SSL-Kommunikation mit dem Server eingesetzt wird. IBM Security Directory Integrator verwendet die RSA-Implementierung aus dem Sicherheitsprovider "IBMJCE". Alle von diesem Provider unterstützten Schlüsselgrößen werden auch durch IBM Security Directory Integrator unterstützt. Ab IBM Security Directory Integrator Version 7.0 können zur Verschlüsselung auch Chiffrierwerte mit geheimen Schlüsseln verwendet werden. Standardmäßig wird aus Gründen der Abwärtskompatibilität RSA verwendet. Die Chiffrierwerte mit geheimen Schlüsseln sind jedoch schneller und sicherer als Chiffrierwerte mit öffentlichen Schlüsseln.

Zur Verschlüsselung von kennwortgeschützten Konfigurationsdateien werden DES- und AES-Algorithmen verwendet. Ein Verschlüsselungsschlüssel (DES oder AES) wird aus der binären UTF-8-Darstellung des Kennworts abgeleitet. Der abgeleitete Verschlüsselungsschlüssel umfasst 64 Bit bei DES und 128 Bit bei AES. Der ECB-Modus wird ohne Auffüllung verwendet.

DES/AES-Schlüssel werden aus Kennwörtern abgeleitet, wenn kennwortgeschützte Konfigurationsdateien verwendet werden. Abgesehen von den oben genannten Schlüsseln generiert IBM Security Directory Integrator keine Schlüssel. Vorhandene Schlüssel werden aus einem externen Schlüsselspeicher geladen. Die Schlüsselstellung und der Zugriff auf den Schlüsselspeicher erfolgen durch die Sicherheitsprovider "IBMJCE" und "IBMJSSE2". Alle Schlüsselgrößen und Algorithmen, die von diesen Providern unterstützt werden, können bei IBM Security Directory Integrator verwendet werden.

Stashdatei

In der Stashdatei werden das Kennwort des Schlüsselspeichers und das Kennwort des eigentlichen Schlüssels gespeichert. Nachstehend erhalten Sie Informationen zum Arbeiten mit einer Stashdatei.

Die Stashdatei enthält die Werte für die Schlüsselspeicherkeywörter des Servers, die mit AES128 und einem festen Schlüssel verschlüsselt sind. Die Stashdatei des Servers heißt "idisrv.sth" (der Name ist nicht konfigurierbar) und wird vom Server aus dem Lösungsordner geladen. Der IBM Security Directory Integrator-Ordner "bin" enthält ein Befehlszeilendienstprogramm für die Erstellung einer Stashdatei. Es heißt *createstash.bat* bzw. *createstash.sh*:

```
createstash <schlüsselspeicherkeywort> [<schlüsselkeywort>] [<sicherheitsproviderklasse>]
```

Hierbei steht *schlüsselspeicherkey* für das Kennwort der Schlüsselspeicherdatei, die in der Systemeigenschaft *api.keystore* angegeben ist, und *<schlüsselkey>* für das Kennwort des privaten Schlüssels des Servers, der in der Systemeigenschaft *api.key.alias* angegeben ist.

Der Wert für *schlüsselkey* ist ein optionaler Parameter, falls kein Wert für den Parameter *<sicherheitsproviderklasse>* angegeben ist. Ist kein Wert für *<schlüsselkey>* angegeben, wird davon ausgegangen, dass das Kennwort für den privaten Schlüssel des Servers mit dem Kennwort des Schlüsselspeichers identisch ist. Wenn der Parameter *<sicherheitsproviderklasse>* beim Dienstprogramm verwendet werden soll, müssen die beiden vorherigen Parameter (das Schlüsselspeicherkey und das Schlüsselkey) angegeben werden. Wird ein Sicherheitsprovider angegeben, wird dieser Provider für die Verschlüsselung verwendet.

Das Dienstprogramm erstellt im aktuellen Verzeichnis eine Datei namens "idisrv.sth" mit den angegebenen Kennwörtern.

Achtung: Im Produktpaket von IBM Security Directory Integrator wird ein Beispiel für eine Stashdatei mit dem Kennwort "server" ausgeliefert. Es wird dringend empfohlen, zur größeren Sicherheit mit dem oben genannten Dienstprogramm eine eigene Stashdatei zu generieren. Außerdem ist zu beachten, dass ein Zugriff auf die Stashdatei ausschließlich durch den eigentlichen IBM Security Directory Integrator-Server, der sie benötigt, möglich sein darf.

Sicherheitsmodi für den Server

Sie können den IBM Security Directory Integrator-Server in zwei Modi (**Standardmodus** und **gesicherter Modus**) ausführen. Nachstehend erhalten Sie Informationen dazu.

Standardmodus

Bei einer Ausführung im Standardmodus werden vom Server die auf Platte gespeicherten Konfigurationen nur dann mit PKI verschlüsselt, wenn ein bestimmter Server-API-Aufruf verwendet wird, der die PKI-Verschlüsselung erfordert. In diesem Modus kann der Server sowohl verschlüsselte als auch unverschlüsselte Konfigurationen lesen.

Gesicherter Modus

Bei einer Ausführung im gesicherten Modus verschlüsselt der Server alle auf Platte gespeicherten Konfigurationen mit der PKI-Verschlüsselung. In diesem Modus kann der Server nur verschlüsselte Konfigurationen lesen und laden. Wenn die Systemeigenschaft *com.ibm.di.server.securemode* auf "true" gesetzt ist, wird der Server im gesicherten Modus ausgeführt. (Eine Systemeigenschaft für die Verwendung des IBM Security Directory Integrator-Servers kann entweder in der Datei *global.properties* bzw. *solution.properties* hinzugefügt oder beim Starten des IBM Security Directory Integrator-Servers direkt in der Java-Befehlszeile angegeben werden: *-Dcom.ibm.di.server.securemode=true*).

Falls beim Starten des Servers in der Java-Befehlszeile die Befehlszeilenoption "-e" angegeben wird, wird der Server ungeachtet des Wertes für die Systemeigenschaft *com.ibm.di.server.securemode* im gesicherten Modus ausgeführt.

Anmerkung: Die kennwortbasierte Verschlüsselung von Konfigurationsdateien aus IBM Security Directory Integrator-Versionen vor 6.0 wird aus Gründen der Ab-

wärtskompatibilität unterstützt. Die kennwortbasierte Verschlüsselung wird verwendet, wenn der Benutzer beim Erstellen der Konfiguration ein Kennwort angibt. Die kennwortbasierte Verschlüsselung von Konfigurationen aus IBM Security Directory Integrator-Versionen vor 6.0 kann nicht mit der PKI-Verschlüsselung kombiniert werden. Falls Sie bei der Ausführung des Servers im gesicherten Modus ein Kennwort angeben, wird eine Fehlernachricht angezeigt.

Mit verschlüsselten IBM Security Directory Integrator-Konfigurationsdateien arbeiten

Sie können eine Verschlüsselungsumsetzung für die Konfigurationsdateien vornehmen. Nachstehend erhalten Sie Informationen zur Verwendung verschlüsselter Dateien und den dabei zu beachtenden Punkten.

Um die Vertraulichkeit von Daten zu gewährleisten, kann IBM Security Directory Integrator Konfigurationsdateien, Eigenschaftswerte in Eigenschaftendateien, Dateien der Serverbenutzerregistry sowie JavaScript-Dateien verschlüsseln.

Die Verschlüsselung in IBM Security Directory Integrator arbeitet mit einer Verschlüsselungsumsetzung, die einen Schlüssel oder ein Schlüsselpaar verwendet. Der Schlüssel bzw. das Schlüsselpaar muss in einer Schlüssel Speicherdatei bereitgestellt werden.

Die Verschlüsselungsumsetzung kann entweder eine Verschlüsselung mit öffentlichem Schlüssel oder eine Verschlüsselung mit geheimem Schlüssel sein. Standardmäßig verwendet IBM Security Directory Integrator die Verschlüsselung mit öffentlichem Schlüssel. (Die Option der Verschlüsselung mit geheimem Schlüssel wurde in IBM Security Directory Integrator Version 7.0 eingeführt. Zuvor wurde ausschließlich die Verschlüsselung mit öffentlichem Schlüssel unterstützt.)

Siehe:

Bei der Verschlüsselung mit öffentlichem Schlüssel wird ein Schlüsselpaar verwendet, das aus einem öffentlichen Schlüssel und einem privaten Schlüssel besteht. Der öffentliche Schlüssel wird für die Verschlüsselung verwendet, der private Schlüssel für die Entschlüsselung. Gegenwärtig wird bei der Verschlüsselung mit öffentlichem Schlüssel nur die RSA-Verschlüsselung unterstützt. Paare aus öffentlichem und privatem Schlüssel können mit den JRE-Standarddienstprogrammen `keytool` und `Ikeyman` generiert und verwaltet werden. Weitere Informationen zur Verwaltung von Zertifikaten mit zugehörigen öffentlichen und privaten Schlüsseln enthält der Abschnitt „Schlüssel, Zertifikate und Schlüssel Speicher verwalten“ auf Seite 103.

Die Datenverschlüsselung in IBM Security Directory Integrator wird durch die folgenden Systemeigenschaften konfiguriert (diese können in der Datei `global.properties` oder `solution.properties` festgelegt werden):

- *com.ibm.di.server.encryption.keystore*: Diese Eigenschaft gibt die Schlüssel Speicherdatei an, die den Schlüssel/das Schlüsselpaar für die Verschlüsselung enthält.
- *com.ibm.di.server.encryption.keystoretype*: Diese Eigenschaft gibt den Typ der Schlüssel Speicherdatei an.
- *com.ibm.di.server.encryption.key.alias*: Diese Eigenschaft gibt den Aliasnamen des Schlüssels/Schlüsselpaares im Schlüssel Speicher an.
- *com.ibm.di.server.encryption.transformation*: Diese Eigenschaft gibt den Namen der Verschlüsselungsumsetzung an (siehe Anmerkungen unten).

Das Kennwort des Schlüsselspeichers und das Kennwort des eigentlichen Schlüssels/Schlüsselpaares (das nicht mit dem Schlüsselspeicherkennwort identisch ist) sind in der Stashdatei des Servers angegeben (siehe Abschnitt „Stashdatei“ auf Seite 147). (Der Zugriff auf einen Schlüsselspeicher wird durch ein Kennwort geschützt, das bei der Erstellung des Schlüsselspeichers durch die Person definiert wird, die den Schlüsselspeicher erstellt, und das nur nach Angabe des aktuellen Kennworts geändert werden kann. Zusätzlich kann jeder private Schlüssel in einem Schlüsselspeicher durch ein eigenes Kennwort geschützt werden.)

Der Name der Umsetzung kann entweder RSA oder eine Umsetzung mit geheimem Schlüssel sein (z. B. AES/CBC/PKCS5Padding). Ausführlichere Erläuterungen der Bestandteile eines Umsetzungsnamens finden Sie unter http://www.ibm.com/developerworks/java/jdk/security/60/secguides/JceDocs/api_users_guide.html#trans. Allgemeine Informationen zur Java-Sicherheit (also der von IBM Security Directory Integrator verwendeten Sicherheit) werden unter der Adresse <http://www-128.ibm.com/developerworks/java/jdk/security/60/secguides/jsse2Docs/JSE2RefGuide.html> bereitgestellt.

Anmerkung:

1. Die Eigenschaften "com.ibm.di.server.encryption.*" betreffen nicht nur die Verschlüsselung von Konfigurationen, sondern auch die Verschlüsselung von Eigenschaftendateien, von JavaScript-Dateien und der Benutzerregistry für die Server-API.
2. Falls Sie den Verschlüsselungsschlüssel und/oder die Verschlüsselungsumsetzung ändern, kann der Server zuvor verschlüsselte Dateien nicht entschlüsseln. Als Ausweichlösung für dieses Problem entschlüsseln Sie die alten Dateien mit dem alten Schlüssel (der alte Schlüssel muss zu diesem Zweck verfügbar sein) und verschlüsseln sie mit dem neuen Schlüssel. Die Verschlüsselung und Entschlüsselung von Dateien kann mit dem Tool "cryptoutils" vorgenommen werden.
3. Beim RSA-Standardalgorithmus gilt hinsichtlich der Daten, die von ihm verarbeitet werden können, eine Längeneinschränkung. IBM Security Directory Integrator verwendet ein angepasstes Schema, das Eingabedaten in gleich große Blöcke unterteilt, die klein genug sind, und verschlüsselt jeden dieser Blöcke separat.
4. Mit RSA verschlüsselte Daten ergeben bei unterschiedlichen Verschlüsselungsausführungen unterschiedliche Verschlüsselungstexte. Dieser Effekt ergibt sich durch das mit RSA verwendeten PKCS#1-Auffüllungsschema.
5. Eine (symmetrische) Verschlüsselung mit einem geheimen Schlüssel kann entweder eine Blockchiffrierung oder eine Datenstromchiffrierung sein. Die Datenstromchiffrierung verschlüsselt die Bit der Nachricht jeweils nacheinander. Bei der Blockchiffrierung werden eine Reihe von Bit als Eingabe verwendet und als gemeinsame Einheit (ein so genannter "Block") verschlüsselt. Blockchiffrierungen (z. B. AES) verwenden einen Rückmeldungsmodus (sodass Muster im unverschlüsselten Text beim verschlüsselten Text nicht erhalten bleiben) sowie ein Auffüllungsschema (damit die Verschlüsselung von Daten möglich ist, deren Länge kein Vielfaches der Blockgröße für den Chiffrierwert darstellt). Datenstromchiffrierungen (z. B. RCE) verwenden weder einen Rückmeldungsmodus noch ein Auffüllungsschema.
6. Falls die Umsetzung eine Blockchiffrierung einbezieht, muss ein Auffüllungsschema verwendet werden (z. B. "PKCS5Padding"). Andernfalls kann der Server Daten, deren Länge keine der Blockgröße für den Chiffrierwert darstellt, nicht verschlüsseln. (Datenstromchiffrierungen verwenden die Auffüllung nicht und sind daher von dieser Einschränkung nicht betroffen.)

7. Der Algorithmus für den Schlüssel bzw. das Schlüsselpaar muss mit dem Algorithmus in der angegebenen Umsetzung übereinstimmen. Wird beispielsweise RSA als Umsetzung verwendet, muss ein RSA-Schlüsselpaar angegeben werden. Bei Verwendung der Umsetzung DES/ECB/PKCS5Padding müssen Sie einen DES-Schlüssel angeben. Mit dem Dienstprogramm keytool können Sie einen neuen geheimen Schlüssel generieren (siehe „Schlüssel, Zertifikate und Schlüsselspeicher verwalten“ auf Seite 103).
8. JKS-Schlüsselspeicher unterstützen geheime Schlüssel nicht. Daher sollten Sie einen anderen Schlüsselspeichertyp verwenden (z. B. JCEKS), falls Sie die Verschlüsselung mit geheimem Schlüssel nutzen wollen.
9. Wird eine Blockchiffrierung in einem Rückmeldungsmodus verwendet, der einen Initialisierungsvektor (IV) erfordert, wird den verschlüsselten Daten der Initialisierungsvektor in Form von unverschlüsseltem Text als Präfix vorangestellt. Der Initialisierungsvektor muss zwar nicht geheimgehalten werden, aber unvorhersehbar sein. Aus diesem Grund wird für jeden verschlüsselten Datenteil ein wahlfreier Initialisierungsvektor generiert. Die Generierung von wahlfreien Daten kann bisweilen ressourcenintensiv sein. Falls das Leistungsverhalten ein wichtiger Aspekt für Sie ist, kann es sinnvoll sein, einen Rückmeldungsmodus ohne Initialisierungsvektor (ECB) zu verwenden.
10. Welche Umsetzungen für geheime Schlüssel bei der Verschlüsselung unterstützt werden, ist vom Leistungsspektrum des Java-Sicherheitsproviders abhängig. Standardmäßig verwendet IBM Security Directory Integrator den Provider "IBMJCE". Unterstützte Blockchiffrierungen sind DES, AES, DESede (Triple DES), Blowfish und RC2. Diese können in jedem der Rückmeldungsmodi ECB, CBC, CFB, OFB und PCBC verwendet werden. Das einzige verfügbare Auffüllungsschema ist "PKCS5Padding". Die Blockchiffrierung "MARS" sollte nicht zur Verschlüsselung verwendet werden, weil die Auffüllung von ihr nicht unterstützt wird (siehe <http://www-128.ibm.com/developerworks/java/jdk/security/50/secguides/JceDocs/api/com/ibm/crypto/provider/Mars.html>). Unterstützte Datenstromchiffrierungen sind RC4 und ARCFOUR (hierbei handelt es sich im Grund genommen um dieselbe Chiffrierung, für die zwei unterschiedliche Namen verwendet werden). Die Datenstromchiffrierung "SEAL" erfordert große Schlüssel (160 Bit) und kann daher nur nach dem Konfigurieren einer nicht eingeschränkten IBM SDK-Richtlinie für die JRE von IBM Security Directory Integrator verwendet werden (<http://www.ibm.com/developerworks/java/jdk/security/60/#sdkpol>).

Trennung von Zertifikaten für PKI-Verschlüsselung und SSL

Verschlüsselte IBM Security Directory Integrator-Konfigurationsdatei neu erstellen

Mit dem Befehlszeilentool "cryptoutils" können Sie eine verschlüsselte IBM Security Directory Integrator-Konfigurationsdatei völlig neu erstellen.

Im Folgenden ist beschrieben, wie Sie eine verschlüsselte IBM Security Directory Integrator-Konfigurationsdatei ganz neu erstellen.

Verwendung des Befehlszeilentools "cryptoutils"

1. Erstellen Sie mit dem Konfigurationseditor eine normale, unverschlüsselte IBM Security Directory Integrator-Konfigurationsdatei.
2. Verschlüsseln Sie diese Konfigurationsdatei mit dem Befehlszeilentool cryptoutils wie im Abschnitt „Verschlüsselungsdienstprogramm von IBM Security Directory Integrator“ auf Seite 153 beschrieben.

3. Zur Ausführung dieser verschlüsselten Konfigurationsdatei müssen Sie den IBM Security Directory Integrator-Server im gesicherten Modus starten (siehe Abschnitt "Sicherheitsmodi für den Server").
4. Zur Bearbeitung dieser verschlüsselten Konfigurationsdatei können Sie eine der beiden Optionen verwenden, die im Abschnitt „Verschlüsselte IBM Security Directory Integrator-Konfigurationsdatei bearbeiten“ beschrieben sind.

Verschlüsselte IBM Security Directory Integrator-Konfigurationsdatei bearbeiten

Mit den hier aufgeführten Schritten können Sie eine verschlüsselte Datei bearbeiten.

Sie können zunächst die verschlüsselte Konfigurationsdatei mit dem Befehlszeilentool `cryptoutils` wie im Abschnitt „Verschlüsselungsdienstprogramm von IBM Security Directory Integrator“ auf Seite 153 beschrieben entschlüsseln. Anschließend können Sie die entschlüsselte Konfiguration mit dem Konfigurationseditor bearbeiten und abschließend die modifizierte Konfigurationsdatei wieder mit dem Tool `cryptoutils` verschlüsseln.

Standardverschlüsselung der Datei 'global.properties' oder 'solution.properties'

Führen Sie die nachstehend beschriebenen Schritte aus, um die Datei "global.properties" oder "solution.properties" zu verschlüsseln.

In den Dateien `global.properties` und `solution.properties` sind eine Vielzahl von Eigenschaften gespeichert, von denen einige sensible Daten wie beispielsweise Kennwörter darstellen können. Zum Schutz dieser sensiblen Daten kann IBM Security Directory Integrator diese Daten verschlüsseln.

Alle Eigenschaften, deren Namen das Präfix `{protect}-` enthalten, werden vom Server unter Verwendung seines öffentlichen Schlüssels mit PKI verschlüsselt. Der Schlüssel des Servers wird durch die Eigenschaft `com.ibm.di.server.encrypted.key.alias` angegeben. Er befindet sich in dem Schlüssel Speicher, der in der Eigenschaft `api.keystore` angegeben ist. Wenn Sie beispielsweise eine Eigenschaft `com.ibm.di.server.encrypted.keystore` verschlüsseln wollen, können Sie die folgende Zeile in der Datei `global.properties` oder `solution.properties` hinzufügen:

```
{protect}-com.ibm.di.any.property=entsprechender_wert
```

Bei seiner nächsten Ausführung stellt der Server fest, dass diese Eigenschaft verschlüsselt werden muss, und überschreibt sofort die Datei. Hierbei wird der Klartextwert "entsprechender_wert" in verschlüsselter Form geschrieben.

Anmerkung: Bei einigen Betriebssystemen (Linux/UNIX-Systeme mit entsprechender Konfiguration) besteht möglicherweise kein Schreibzugriff auf die Datei `global.properties`. In diesem Fall gibt der Server eine Warnung aus, dass die Datei nicht geschrieben/verschlüsselt wurde.

Der Schutz für die Eigenschaften in der Datei `global.properties` oder `solution.properties` ist auch über die Eigenschaftsspeicher "Globale Eigenschaften" und "Lösungseigenschaften" zugänglich, auf die Sie über die Option **Server Speicher durchsuchen** im Konfigurationseditor zugreifen können.

Verschlüsselung von Eigenschaften in externen Eigenschaftendateien

Sie können Eigenschaften in externen Eigenschaftendateien mithilfe eines benannten Zertifikats aus dem Schlüsselspeicher des Servers verschlüsseln.

Eigenschaften, die in externen Eigenschaftendateien gespeichert sind, können genauso wie Eigenschaften in den Dateien `global.properties` oder `solution.properties` durch Verschlüsselung geschützt werden.

Statt das Standardzertifikat des Servers zu verwenden, ist es möglich, Eigenschaften in externen Eigenschaftendateien mit einem speziell benannten Zertifikat aus dem Schlüsselspeicher des Servers zu verschlüsseln.

Weitere Informationen zur Verschlüsselung von Eigenschaften, die in diesen Dateien gespeichert sind, finden Sie unter „Standardverschlüsselung der Datei 'global.properties' oder 'solution.properties'“ auf Seite 152. Eigenschaften in einer externen Eigenschaftendatei besitzen die folgende Syntax:

```
[{protect}-]schlüsselwort <doppelpunkt | gleichheitszeichen> [{encr}][{java}]wert
```

- Das optionale Präfix `{protect}-` signalisiert, dass der Wert entweder verschlüsselt ist oder verschlüsselt werden soll. Wenn der Wert mit der Zeichenfolge `"{encr}"` beginnt, ist er bereits verschlüsselt.
- Das optionale Werteprefix `{java}` gibt an, dass der Wert ein serialisiertes Java-Objekt ist. Der Wert muss b64-codiert sein. Beispiel:

```
{protect}-api.truststore.pass={encr}J8AKimpEutu3Bb10Vg55F/5d5v02kXwCNUwNcQ3vINUc6K0719z9dEK3H430t2iTT1dZTI6FSSVin9KsCy  
BLmgv+n84w7He1K13ro2dFmZbTYKMXuxGoqN9nL2V0vZoptNqzoWvs6IN/p3Vkl1IBt1ao/9mEPEKuIwRnKtkQ89Bg=
```

Verschlüsselungsdienstprogramm von IBM Security Directory Integrator

Sie können die Dateien mit dem Dienstprogramm `cryptoutils` bearbeiten. Beachten Sie dabei die aufgeführten Parameter.

Das Verzeichnis `"tdi-installationsverzeichnis/serverapi"` enthält ein Dienstprogramm (`cryptoutils`), mit dem Sie Dateien entschlüsseln und wieder verschlüsseln können (beispielsweise die Identitätsregistrydatei), damit Sie die Datei manuell bearbeiten können.

Das Tool erkennt die folgenden Befehlszeilenparameter:

input {Erforderlich} Gibt die Datei an, die verschlüsselt oder entschlüsselt werden soll.

output

{Erforderlich} Gibt die neue Datei an, die nach erfolgter Verschlüsselung oder Entschlüsselung mit den Ergebnisdaten erstellt wird. Falls die Datei vorhanden ist, wird sie überschrieben.

mode {Erforderlich} Gibt den Modus an, in dem das Tool ausgeführt wird. Es kann einer der folgenden Modi verwendet werden:

- `encrypt`: Benutzerregistry verschlüsseln
- `decrypt`: Benutzerregistry entschlüsseln
- `encrypt_config`: IBM Security Directory Integrator-Konfigurationsdatei oder JavaScript-Datei verschlüsseln
- `decrypt_config`: IBM Security Directory Integrator-Konfigurationsdatei oder JavaScript-Datei entschlüsseln

- *encrypt_props*: Werte aller geschützten Eigenschaften in einer IBM Security Directory Integrator-Eigenschaftendatei verschlüsseln
- *decrypt_props*: Werte aller geschützten Eigenschaften in einer IBM Security Directory Integrator-Eigenschaftendatei entschlüsseln

Anmerkung: Benutzerregistrydateien werden anders verschlüsselt als Konfigurations- und JavaScript-Dateien.

keystore

{Erforderlich} Gibt die Schlüsselspeicherdatei an, die den Schlüssel für die Verschlüsselung/Entschlüsselung enthält.

storepass

{Erforderlich} Gibt das Kennwort für die Schlüsselspeicherdatei an.

alias {Erforderlich} Gibt den Aliasnamen des Verschlüsselungs-/Entschlüsselungsschlüssels im Schlüsselspeicher an.

keypass

{Optional} Gibt das Kennwort des Verschlüsselungs-/Entschlüsselungsschlüssels an. Standardmäßig wird für den Zugriff auf den Schlüssel das Schlüsselspeicherkenwort verwendet.

transformation

{Optional} Gibt den Namen der Verschlüsselungsumsetzung an, die für die Verschlüsselung/Entschlüsselung verwendet werden. Dies kann RSA oder eine beliebige Umsetzung mit geheimem Schlüssel sein (z. B. AES/CBC/PKCS5Padding). Die Standardeinstellung ist RSA.

storetype

{Optional} Gibt den Typ der Schlüsselspeicherdatei an (z. B. JKS). Bei diesem Parameter muss die Groß-/Kleinschreibung nicht beachtet werden, die Werte JCEKS und jceks sind also äquivalent. Falls dieser Parameter fehlt, wird der Standardschlüsselspeichertyp der JRE verwendet (konfiguriert durch die Sicherheitseigenschaft "keystore.type" in der Datei "java.security" der JRE).

cryptoproviderclass

{Optional} Gibt den Java-Sicherheitsprovider an, der für die Verschlüsselung/Entschlüsselung (jedoch nicht für den Zugriff auf den Schlüsselspeicher) verwendet wird. Standardmäßig werden die Provider aus der Sicherheitsproviderliste der JRE (konfiguriert in der JRE-Datei "java.security") verwendet.

Beispiele:

Benutzerregistry verschlüsseln

Ein im gesicherten Modus ausgeführter IBM Security Directory Integrator-Server erfordert die Verschlüsselung der Benutzerregistry mit dem Schlüssel des Servers.

Sie können eine Benutzerregistrydatei aus unverschlüsseltem Text folgendermaßen verschlüsseln:

```
cryptoutils -input registry.txt -output registry.enc -mode encrypt
-keystore ../testserver.jks -storepass server -alias server
```

IBM Security Directory Integrator-Konfiguration entschlüsseln

```
cryptoutils -input myconfig.enc.xml -output myconfig.xml -mode decrypt_config -keystore ../testserver.jks
-storepass server -alias server
```

Dieser Befehl entschlüsselt die Konfigurationsdatei "myconfig.enc.xml" (möglicherweise durch einen IBM Security Directory Integrator-Server er-

stellt, der im gesicherten Modus ausgeführt wird). Die entschlüsselte Konfiguration "myconfig.xml" kann anschließend ohne großen Aufwand im Konfigurationseditor modifiziert werden. Nach dem Modifizieren der Konfiguration kann diese erneut verschlüsselt werden, damit sie von einem IBM Security Directory Integrator-Server im gesicherten Modus gelesen und verwendet werden kann.

IBM Security Directory Integrator-Konfiguration mit symmetrischer Verschlüsselung (andere als Standardverschlüsselung "RSA") verschlüsseln

```
cryptoutils -input myconfig.xml -output myconfig.enc.xml -mode encrypt_config -keystore ../server.jck  
-storepass server -alias server -transformation AES/CBC/PKCS5Padding -storetype jceks
```

Der obige Befehl setzt voraus, dass der Schlüsselspeicher "server.jck" vorhanden ist. Es wird davon ausgegangen, dass der Schlüsselspeicher unter dem Aliasnamen "server" einen geheimen AES-Schlüssel enthält.

Datei "global.properties" entschlüsseln

Der IBM Security Directory Integrator-Server verschlüsselt automatisch die Werte von geschützten Eigenschaften, wenn er die Datei `global.properties/solution.properties` liest.

Sie können alle in der Datei `global.properties` verschlüsselten Werte folgendermaßen entschlüsseln:

```
cryptoutils -input ../etc/global.properties -output ../etc/global.properties -mode decrypt_props  
-keystore ../testserver.jks -storepass server -alias server
```

Anmerkung: Wenn das Tool `cryptoutils` zum Ver- und Entschlüsseln der „Benutzerregistry der Server-API“ auf Seite 140, von Konfigurationsdateien (Details zur Behandlung von verschlüsselten Konfigurationen durch den Server siehe „Sicherheitsmodi für den Server“ auf Seite 148) oder von TDI-Server-Hooks (siehe „Verschlüsselung von IBM Security Directory Integrator-Server-Hooks“ auf Seite 160) verwendet wird, wird eine Datei jeweils als Einheit verschlüsselt und entschlüsselt.

Im Modus für die Ver-/Entschlüsselung von Eigenschaftendateien werden hingegen nur die Werte der geschützten Eigenschaften und nicht die ganze Datei verschlüsselt. Daher sind nach dem Verschlüsseln einer Datei ".properties" mit dem Modus `encrypt_props` die Eigenschaftsschlüssel und die Kommentare in der Datei weiterhin für die Benutzer lesbar. Zusätzliche Angaben über geschützte Eigenschaften enthalten die Abschnitte „Standardverschlüsselung der Datei 'global.properties' oder 'solution.properties'“ auf Seite 152 und „Verschlüsselung von Eigenschaften in externen Eigenschaftendateien“ auf Seite 153.

Sicherheit für IBM Security Directory Integrator-Systemspeicher

Mit Derby können Sie das Repository von Benutzern und Kennwörtern definieren. Verwenden Sie die hier aufgeführte Liste, um für die einzelnen Eigenschaften den entsprechenden Wert festzulegen. Außerdem können Sie mit den hier aufgeführten Anweisungen den von Derby bereitgestellten Benutzerautorisierungsmechanismus verwenden.

Der IBM Security Directory Integrator-Systemspeicher ist die Datenbank oder die permanente Schicht, in der alle Informationen gespeichert werden, die ein IBM Security Directory Integrator-Server benötigt. Früher wurde diese Schicht nicht durch eine eigene Sicherheit geschützt. Jeder Benutzer konnte auf den Systemspeicher zugreifen. Ab IBM Security Directory Integrator Version 7.0 gibt es für den Systemspeicher eine konfigurierbare Sicherheit.

In IBM Security Directory Integrator 7.0 wird der Systemspeicher standardmäßig im Netzmodus verwendet. Auf diese Weise können eine Reihe von IBM Security Directory Integrator-Instanzen und andere Anwendungen gleichzeitig auf den Systemspeicher zugreifen. Weil der Systemspeicher über das Netz verfügbar ist, muss er durch eine gewisse Sicherheit umgeben sein, damit die durch den IBM Security Directory Integrator-Server verwalteten Daten geschützt sind.

Derby (zuvor Cloudscape genannt) bietet verschiedene Möglichkeiten, um das Repository von Benutzern und Kennwörtern zu definieren. Um anzugeben, welchen dieser Services Sie mit Ihrem Derby-System verwenden wollen, setzen Sie die Eigenschaft `derby.authentication.provider` auf den entsprechenden Wert. Die verschiedenen Einstellungen sind in den nachfolgenden Abschnitten erläutert.

Externer Verzeichnisservice

Ein Verzeichnisservice speichert Namen und Attribute dieser Namen. Derby verwendet JNDI (Java Naming and Directory Interface), um mit externen Verzeichnisservices zu interagieren, die eine Authentifizierung der Namen und Kennwörter von Benutzern bereitstellen können.

Sie können Derby so konfigurieren, dass die Authentifizierung von Benutzern anhand eines vorhandenen LDAP-Verzeichnisservice in Ihrem Unternehmen zulässig ist. LDAP (Lightweight Directory Access Protocol) bietet ein offenes Verzeichniszugriffsprotokoll, das über TCP/IP ausgeführt wird. Ein LDAP-Verzeichnisservice kann den Namen und das Kennwort eines Benutzers rasch authentifizieren.

Nach dem Konfigurieren einer Reihe von durch Derby definierten Eigenschaften können Sie mit der Verwendung eines externen Verzeichnisservices als Repository für Benutzernamen und Kennwörter beginnen.

Benutzerdefinierte Klasse

Beim Ansatz der benutzerdefinierten Klasse können Sie Derby an einen anderen externen Authentifizierungsservice als LDAP binden.

Legen Sie für `derby.authentication.provider` den vollständigen Namen einer Klasse fest, die die allgemein zugängliche Schnittstelle `org.apache.derby.authentication.UserAuthenticator` implementiert. Indem Sie eine eigene Klasse schreiben, die einige Minimalanforderungen erfüllt, können Sie Derby an einen externen Authentifizierungsservice binden.

Die Klasse, die den externen Authentifizierungsservice bereitstellt, muss die allgemein zugängliche Schnittstelle `org.apache.derby.authentication.UserAuthenticator` implementieren und gegebenenfalls Ausnahmebedingungen des Typs `java.sql.SQLException` auslösen.

Integrierte Derby-Benutzer

Derby stellt ein einfaches Repository zum Speichern der Benutzernamen und Kennwörter zur Verfügung. Zur Verwendung dieses integrierten Repositories muss die Eigenschaft `derby.authentication.provider=BUILTIN` festgelegt werden.

Der IBM Security Directory Integrator-Systemspeicher verwendet das integrierte Repository zum Speichern des Benutzernamens und Kennworts. Da IBM Security Directory Integrator für den Zugriff auf den Systemspeicher nur über einen einzigen Benutzer verfügt, ist dies der praktikabelste Provider, der verwendet werden kann.

Benutzerauthentifizierung

Die Details der Benutzerauthentifizierung betreffen die Authentifizierung von Benutzern. Der Benutzerauthentifizierungsmechanismus nimmt die Authentifizierung nur dann vor, wenn der Benutzername im benannten Repository (das ein beliebiges der oben aufgeführten Repositories sein kann) vorhanden und das Kennwort für den angegebenen Benutzer korrekt ist. Falls Sie jedoch eine weiter reichende Steuerung der Zugriffsberechtigungen benötigen, können Sie den von Derby bereitgestellten Benutzerautorisierungsmechanismus verwenden.

Der Master-Switch für die Anforderung, dass Benutzer anhand von bereitgestellten Parametern authentifiziert werden müssen, ist die Eigenschaft `derby.connection.requireAuthentication`; ihr Standardwert ist `TRUE`.

Die Zugriffsmodi können unter Verwendung der Eigenschaft `derby.database.defaultConnectionMode=fullAccess` festgelegt werden. Diese Eigenschaft definiert den Standardzugriffsmodus für alle Benutzer im Derby-Repository. Sie definiert außerdem die Zugriffsebene für den Systemspeicherbenutzer. Die unterschiedlichen, von Derby unterstützten Zugriffsebenen sind `fullAccess` (uneingeschränkter Zugriff), `readOnly` (Lesezugriff) und `noAccess` (kein Zugriff). Falls Sie jedoch für bestimmte Benutzer spezielle Zugriffsmodi verwenden wollen, können Sie den Zugriff unter Verwendung der folgenden Eigenschaften zuordnen:

- `derby.database.fullAccessUsers=<benutzernamen>`: Allen angegebenen Benutzern wird ein uneingeschränkter Zugriff erteilt.
- `derby.database.readOnlyAccessUsers=<benutzernamen>`: Allen angegebenen Benutzern wird der Lesezugriff erteilt.
- `derby.database.noAccessUsers=<benutzernamen>`: Allen angegebenen Benutzern wird der Zugriff auf die Datenbank verweigert.

Für die Variable `benutzernamen` sollte eine durch Kommas getrennte Liste von Benutzern angegeben werden. Beispiel:

```
derby.database.fullAccessUsers=sa, mary
```

In der aktuellen Version von IBM Security Directory Integrator gibt es nur einen einzigen Benutzer, der auf den Systemspeicher zugreift. Dieser Benutzer ist zur Ausführung aller Operationen für den Systemspeicher erforderlich, weshalb der Zugriffsmodus mit `fullAccess` definiert ist.

Verschiedene Funktionen für Konfigurationsdateien

Mit den nachstehenden Informationen erhalten Sie ein umfassendes Verständnis der verschiedenen Funktionen für Konfigurationsdateien.

Konfigurationsparametertyp "password"

Die Konfigurationsparameter einer IBM Security Directory Integrator-Komponente in einer Konfiguration können den Typ "string", "number", "boolean" usw. besitzen. Einer der verfügbaren Typen ist "password". Falls ein Konfigurationsparameter den Typ "password" hat, zeigt der Konfigurationseditor seinen Wert im Komponentenkonfigurationsfenster als Folge von Zeichen "*" an, sowohl beim Eingeben eines neuen Kennworts als auch beim Öffnen einer vorhandenen Konfiguration zur Bearbeitung oder Ausführung.

Kennwortschutz für Komponenten

Mit den hier aufgeführten Anweisungen können Sie die Komponentenkennwörter in einem Standardeigenschaftsspeicher definieren.

IBM Security Directory Integrator speichert Konfigurationsdaten in einer XML-Datei, die für alle Konfigurationswerte unverschlüsselten Text enthält. Dies betrifft auch sensible Informationen wie Kennwörter. IBM Security Directory Integrator unterstützt die Verschlüsselung der gesamten Konfigurationsdatei, schützt oder verschlüsselt jedoch sensible Informationen nicht, wenn die Konfigurationsdatei als unverschlüsselter Text gespeichert wird.

Zum besseren Schutz der Kennwörter, die für die verschiedenen Komponenten benötigt werden, bietet IBM Security Directory Integrator ein spezielles Verfahren. Die Kennwörter werden in einer Konfiguration mit unverschlüsseltem Text verdeckt und für gespeicherte Kennwörter gibt es eine Standardsicherheit. Hierzu werden (gespeicherte und abgerufene) Komponentenkennwörter nicht in der Konfigurationsdatei, sondern in einem Standardeigenschaftsspeicher abgelegt. In IBM Security Directory Integrator kann ein benutzerdefinierter Eigenschaftsspeicher ein beliebiges System sein, für das es einen Connector gibt. Am häufigsten wird eine externe Eigenschaftendatei als Standardeigenschaftsspeicher verwendet. Alle Komponentenkennwörter werden standardmäßig in diesem Standardeigenschaftsspeicher und (anders als bei älteren Versionen des Produkts) nicht in der Konfigurationsdatei abgelegt. Daher können Kennwörter von der Konfigurationsdatei isoliert werden, wenn diese Funktionsweise nicht explizit durch den Benutzer außer Kraft gesetzt wird (was bei der Anfangsentwicklung sinnvoll sein kann).

Kennwörter für konfigurierte Eigenschaften speichern

Mit den hier aufgeführten Anweisungen können Sie ein Kennwort mithilfe des Kennwortspeichers speichern.

Informationen zu diesem Vorgang

Der Kennwortschutzmechanismus ist direkt mit den Konfigurationsfenstern verbunden, die für einen Benutzer angezeigt werden. Die Konfigurationsfenster (also die Formulare) enthalten für jeden Parameter eine Beschreibung und seine Syntax. Einer der Typen für die Syntax ist der Typ *password*. Dieser Typ bewirkt, dass der Konfigurationseditor für die Bearbeitung ein Kennworttextfeld verwendet. Immer dann, wenn der Wert eines Parameters mit der Syntaxkomponente "password" geändert wird, wird der Wert des Kennworts in einem externen Repository gespeichert, das als *Kennwortspeicher* bezeichnet wird. Dieses externe Repository für Kennwörter wird im Konfigurationseditor auf der Seite *Eigenschaften (Kennwortspeicher)* konfiguriert und ist in der Konfigurationsdatei für die aktuelle IBM Security Directory Integrator-Lösung angegeben. Falls kein solcher Eigenschaftsspeicher konfiguriert ist, wird das Kennwort in der Konfigurationsdatei als unverschlüsselter Text gespeichert.

Wenn ein Standardkennwort konfiguriert ist, wird beim erstmaligen Speichern eines Parameters mit dem Typ "protected" oder "password" ein eindeutiger Eigenschaftsname generiert. Dieser Schlüssel wird als Schlüssel im Kennwortspeicher verwendet. Derselbe Eigenschaftsname wird als Standardeigenschaftsreferenz in die Konfigurationsdatei geschrieben. Wenn der Wert zu einem späteren Zeitpunkt abgerufen wird, findet eine Standardeigenschaftsauflösung statt, um den tatsächlichen Wert aus dem Kennwortspeicher abzurufen.

Falls ein Kennwortspeicher angegeben ist, wird für das Kennwort ein eindeutiger Schlüssel generiert und das Kennwort wird unter diesem Schlüssel im Kennwort-

speicher in verschlüsselter Form abgelegt. In der Konfigurationsdatei wird das Kennwort nur durch diesen Schlüssel referenziert.

Falls kein Kennwortspeicher angegeben ist, erscheint das Kennwort in der Konfigurationsdatei als unverschlüsselter Text.

Beispiel:

1. Erstellen Sie im Konfigurationseditor ein neues Projekt.
2. Klicken Sie in der Navigationsansicht mit der rechten Maustaste auf den Ordner "Eigenschaften" und wählen Sie für "Neuer Eigenschaftsspeicher" den Namen "MyProps" aus.
3. Geben Sie auf der Registerkarte "Connector" des neu erstellten Eigenschaftsspeichers im Feld "Pfad/URL für Objektgruppe" den Wert "MyProps.properties" ein.
4. Geben Sie an, dass der neue Eigenschaftsspeicher als Kennworteigenschaftsspeicher verwendet werden soll (klicken Sie hierzu in der Navigationsansicht mit der rechten Maustaste auf den neuen Eigenschaftsspeicher und wählen Sie die Option **Kennworteigenschaftsspeicher** aus).
5. Fügen Sie eine neue Fertigungslinie mit einem FTP-Client-Connector hinzu.
6. Geben Sie im Feld "Kennwort für Anmeldung" des FTP-Client-Connectors ein Kennwort ein.
7. Speichern Sie die Lösung und schließen Sie den Konfigurationseditor.

Nach der obigen Prozedur enthält die Konfigurationsdatei der erstellten Lösung Zeilen, die dem folgenden Text ähneln:

```
<parameter name="ftpPass">@SUBSTITUTE{property.MyProps:ftpPass-38ae53e8779cfd65}</parameter>
.....
<PasswordStore>MyProps</PasswordStore>
```

In der Datei "MyProperties.properties" gibt es eine Zeile, die in etwa dem folgenden Text entspricht:

```
{protect}-ftpPass-38ae53e8779cfd65={encr}GVJC01A7VUiW=
```

Dies bedeutet, dass die FTP-Kennwortkonfiguration in der Lösungsdatei eine verschlüsselte Eigenschaft aus dem aktuellen Kennwortspeicher ("MyProps") referenziert. Der verwendete Eigenschaftsschlüssel ist "ftpPass-38ae53e8779cfd65".

Ausgabe von Attributen als unverschlüsselter Text während Traceerstellung verhindern

Mit den hier aufgeführten Methoden können Sie sensible Daten während der Traceerstellung schützen.

Ersteller von IBM Security Directory Integrator-Lösungen benötigen ein Verfahren, mit dem verhindert werden kann, dass sensible Daten (z. B. Kennwörter) als unverschlüsselter Text ausgegeben werden, wenn eine Traceerstellung für die Lösung erforderlich ist. Daher wurden in IBM Security Directory Integrator einige Methoden, die mit der Attributklasse verbunden sind, dahingehend erweitert, dass angegeben wird, ob ein Attribut geschützt ist oder nicht. Falls das Attribut geschützt ist und die Traceerstellung aktiviert ist, wird anstelle des tatsächlichen Wertes eine feste Anzahl von Sternzeichen (*) ausgegeben.

Wenn im Taskaufrufblock (TaskCallBlock - TCB) Verbindungsparameter gefunden werden, werden die Werte in keinem Fall direkt durch IBM Security Directory Integrator protokolliert. Es wird zwar die Tatsache protokolliert, dass Parameter ange-

geben sind, aber die eigentlichen Werte werden nicht protokolliert. Falls für die Lösung ein Debug ausgeführt werden muss, kann für diese Werte manuell ein Speicherauszug erstellt werden (z. B. durch Scripterstellung).

Verschlüsselung von IBM Security Directory Integrator-Server-Hooks

Sie sollten sensible Daten verschlüsseln, bevor Sie sie zum Verzeichnis für Server-Hooks hinzufügen. Nachstehend erhalten Sie detaillierte Informationen dazu.

Server-Hook-Scripts werden durch die Erstellung von Dateien im Unterverzeichnis "serverhooks" des Lösungsverzeichnisses erstellt und verfügbar gemacht. Scripts, die sensible Informationen enthalten, sollten mit der Server-API verschlüsselt werden, bevor sie in das Verzeichnis aufgenommen werden. Zur Verschlüsselung von Scripts kann das Tool `cryptoutils` verwendet werden (siehe „Verschlüsselungsdienstprogramm von IBM Security Directory Integrator“ auf Seite 153). Beachten Sie, dass der IBM Security Directory Integrator-Server ausschließlich Dateien mit der Dateinamenerweiterung ".jse" entschlüsselt. Die Erweiterung ".jse" signalisiert dem IBM Security Directory Integrator-Server, dass die Scriptdatei verschlüsselt ist. Aus diesem Grund müssen Sie nach dem Verschlüsseln einer Server-Hook-Scriptdatei daran denken, die Dateinamenerweiterung in ".jse" zu ändern.

Ferner Konfigurationseditor und SSL

Beachten Sie bei der Arbeit mit dem fernen Konfigurationseditor und SSL die aufgeführten Punkte.

Der Konfigurationseditor, mit dem ferne Konfigurationsdateien (also Konfigurationsdateien auf einem fernen System) bearbeitet werden, wird als "ferner Konfigurationseditor" bezeichnet. Der ferne Konfigurationseditor von IBM Security Directory Integrator kann Fertigungslinien in Konfigurationen starten, die zur Bearbeitung geöffnet sind. Der ferne Konfigurationseditor ist ein Client der Server-API des fernen IBM Security Directory Integrator-Servers. Infolgedessen wird der ferne Konfigurationseditor als Client der Server-API authentifiziert und autorisiert. Damit dies bei Verwendung von SSL funktioniert, müssen die folgenden Bedingungen erfüllt sein:

1. Der Server, zu dem der ferne Konfigurationseditor eine Verbindung herstellt, muss so konfiguriert sein, dass die SSL-Clientauthentifizierung erforderlich ist. Dies ist eine Konfiguration der Server-API. Details können Sie unter „SSL-basierte Authentifizierung“ auf Seite 127 nachlesen.
2. Die IBM Security Directory Integrator-Instanz des fernen Konfigurationseditors muss so konfiguriert sein, dass die SSL-Clientauthentifizierung bereitgestellt wird. Dies wird in einer SSL-Clientauthentifizierung konfiguriert (siehe „SSL-Clientauthentifizierung“ auf Seite 114).

Diese SSL-Clientauthentifizierung wird benötigt, weil der ferne Konfigurationseditor Listener-Objekte verwendet, damit er benachrichtigt werden kann, wenn eine Fertigungslinie beendet wurde. Damit diese Funktionsweise bei SSL erreicht wird, muss der Client der Serveridentität vertrauen und der Server muss der Clientidentität vertrauen.

Fernen Konfigurationseditor verwenden

Beachten Sie die Einschränkungen bei der Verwendung des fernen Konfigurationseditors.

Die Verwendung eines fernen Konfigurationseditors unterscheidet sich etwas von der Verwendung eines lokalen Konfigurationseditors. Bei der Verwaltung einer Konfiguration auf einem fernen System mit dem fernen Konfigurationseditor müssen Sie einige Einschränkungen bedenken, die für den Konfigurationseditor im Fernmodus gelten. Zu beachtende Einschränkungen:

- Beim lokalen Bearbeiten von Konfigurationsdateien ist es ausreichend, wenn der entsprechende Dateisystemzugriff (Lese- und Schreibzugriff) auf die Konfigurationsdatei besteht. Beim Bearbeiten einer fernen Konfiguration benötigen Sie hingegen die Administratorberechtigung für die ferne Konfigurationsinstanz.
- Werden (unter Verwendung der Schaltflächen **Verbindung herstellen** in Zuordnungsfenstern) Verbindungen zu einer Datenquelle hergestellt, werden diese Verbindungen lokal ausgewertet.
Beispielsweise führt die Angabe "ldap://localhost:389" dazu, dass der Konfigurationseditor versucht, eine Verbindung zum lokalen LDAP-Server und nicht zum LDAP-Server auf dem fernen Computer herzustellen.
- Wenn bei der Generierung von Connectors für Web-Services Funktionskomponenten entstehen, die die WSDL-Datei, die JAR-Dateien usw. generieren (mit dem Generator für komplexe Typen), werden diese lokal generiert. Auf dem fernen System, mit dem der Konfigurationseditor verbunden ist, werden diese Komponenten nicht generiert und müssen zur Implementierung manuell in das ferne System hochgeladen werden.
- Der ferne Konfigurationseditor lässt nur das Bearbeiten und Anzeigen derjenigen Konfigurationen zu, die in dem Ordner vorhanden sind, der durch die Eigenschaft `api.config.folder` angegeben ist.
- Beim Arbeiten mit Operationen für den **Systemspeicher** (beispielsweise beim Löschen des Schlüssels für den Iteratorstatus usw.), die im Konfigurationseditor verfügbar sind, werden diese Operationen für den lokalen Systemspeicher und nicht für den Systemspeicher des fernen IBM Security Directory Integrator-Computers ausgeführt. Nur beim Ausführen der Fertigungslinie stellt die Fertigungslinie die Verbindung zum fernen Systemspeicher her, da die Fertigungslinie zum Zeitpunkt ihrer Ausführung in der fernen JVM ausgeführt wird.
- Wenn Sie den Editor für die **Parametersubstitution** verwenden (verfügbar über die Tastenkombination Strg+E), werden im Editor nur die lokalen Eigenschaften und nicht die auf dem fernen System festgelegten Eigenschaften angezeigt. Analog wird beim Erstellen und Speichern eines neuen Eigenschaftsspeichers (Dateityp) der Eigenschaftsspeicher (Datei) lokal gespeichert.
- Wenn Sie ferne Konfigurationsdateien mit dem Konfigurationseditor bearbeiten, unterliegen Sie der Authentifizierung und Autorisierung durch die Server-API, da der Konfigurationseditor als Clientanwendung agiert. Daher müssen Sie auf dem fernen Server die Zugriffsberechtigung `admin` (Administrator) besitzen, um den Konfigurationseditor auf diese Weise verwenden zu können.
- Bei der Verwendung des fernen Servers muss der ferne Server selbst einen ausreichenden Zugriff auf das lokale Dateisystem besitzen, in dem die Konfigurationsdateien gespeichert sind. Falls die Konfigurationsdateien in einem schreibgeschützten Dateisystem oder an einer Dateispeicherposition gespeichert sind, auf die die Benutzer-ID, unter der der ferne Server ausgeführt wird, keinen Schreibzugriff besitzt, können Sie ferne Konfigurationen nicht bearbeiten.

Übersicht über Konfigurationsdateien und Eigenschaften für Sicherheit

Nachstehend finden Sie eine Übersicht über Konfigurationsdateien und Eigenschaften für Sicherheit.

Tabelle 19. Tabelle der zuvor erläuterten Konfigurationsdateien und ihres Inhalts

Konfigurationsdatei	Position	Beschreibung
global.properties	<i>tdi-ausgangsverzeichnis/etc</i>	Diese Datei ist die primäre Konfigurationsdatei für den Server.
solution.properties	Lösungsordner	Diese Datei (<i>solution.properties</i>) ist zunächst eine Kopie der Datei <i>global.properties</i> , die von der aktuellen Lösung verwendet wird. Nachdem Sie Änderungen vorgenommen haben, setzen die Werte in dieser Datei die entsprechenden Werte der Datei <i>global.properties</i> außer Kraft.
registry.txt	<i>tdi-ausgangsverzeichnis/serverapi</i>	Diese Datei ist die Benutzerregistry für die Server-API. Sie wird durch die Eigenschaft "api.user.registry" in der Datei "global.properties" definiert.
build.properties	<i>tdi-ausgangsverzeichnis/etc</i>	Diese Datei enthält die Erstellungsinformationen, das Erstellungsdatum, die Version usw. von IBM Security Directory Integrator. Es handelt sich um eine Textdatei, die standardmäßig die native Codierung der Plattform verwendet.
tdisrvctl-log-4j.properties	<i>tdi-ausgangsverzeichnis/etc</i>	Diese Datei steuert die Protokollierungsstrategie für das Befehlszeilendienstprogramm "tdisrvctl".
Log4J.properties	<i>tdi-ausgangsverzeichnis/etc</i>	Diese Datei steuert die Protokollierungsstrategie für den Server (<i>ibmdisrv</i>), wenn dieser über die Befehlszeile gestartet wird.
jlog.properties	<i>tdi-ausgangsverzeichnis/etc</i>	Diese Datei steuert die Strategie für die Traceerstellung und die Erfassung von Fehlerdaten beim ersten Auftreten (First-Failure Data Capture - FFDC).
ibmdi.ico	<i>tdi-ausgangsverzeichnis/etc</i>	Diese Datei enthält eine Liste der Symbole für IBM Security Directory Integrator.
idisrv.sth	<i>tdi-ausgangsverzeichnis</i>	Diese Datei ist die Stashdatei für den IBM Security Directory Integrator-Server. Es handelt sich um eine Binärdatei, die das verschlüsselte Kennwort der Beispielschlüsselspeicherdatei für den Server (<i>testserver.jks</i>) enthält.
derby.properties	<i>tdi-ausgangsverzeichnis/etc</i>	Diese Datei enthält die Standardkonfiguration für den mit IBM Security Directory Integrator ausgelieferten Derby-Systemspeicher.
reconnect.rules	<i>tdi-ausgangsverzeichnis/etc</i>	Diese Datei enthält Text, der Regeln für die Verbindungswiederherstellung definiert und somit angibt, wie IBM Security Directory Integrator Ausnahmebedingungen bei der Verbindungswiederherstellung handhaben soll.
global.properties.v611	<i>tdi-ausgangsverzeichnis/etc</i>	Diese Datei dient als Beispielplatzhalter und ist während der Migration von Nutzen.
TDI0701.SYS2	<i>tdi-ausgangsverzeichnis/etc</i>	Dies ist die Datei mit der Programmkennung (Lizenz), die vom ITLM-Agenten verwendet wird, um IBM Security Directory Integrator zu erkennen.

Tabelle 19. Tabelle der zuvor erläuterten Konfigurationsdateien und ihres Inhalts (Forts.)

Konfigurationsdatei	Position	Beschreibung
pkcs11.cfg	<i>tdi-ausgangsverzeichnis/etc</i>	Diese Datei wird für die Initialisierung des IBM Providers für die PKCS11-Implementierung verwendet. Details finden Sie im Abschnitt über die Konfigurationsdatei für PKCS11.
testadmin.der	<i>tdi-ausgangsverzeichnis/serverapi</i>	Diese Datei ist das exportierte Zertifikat aus testadmin.jks.
testadmin.jks	<i>tdi-ausgangsverzeichnis/serverapi</i>	Diese Datei enthält einen Beispielschlüsselspeicher und Beispiel-Truststore für einen fernen Client der Server-API.
cryptoutils.bat(sh)	<i>tdi-ausgangsverzeichnis/serverapi</i>	Diese Datei ist ein Befehlszeilendienstprogramm (Shell-Script), das zum Verschlüsseln und Entschlüsseln von IBM Security Directory Integrator-Konfigurationen und der Benutzerregistrydatei verwendet wird.
testserver.jks	<i>tdi-ausgangsverzeichnis</i>	Diese Datei wird als Beispiel für den Schlüsselspeicher und den Truststore des Servers verwendet.
testserver.der	<i>tdi-ausgangsverzeichnis</i>	Diese Datei ist ein exportiertes Serverzertifikatsbeispiel, das in einen Truststore importiert werden kann.
am_config.properties	<i>tdi-ausgangsverzeichnis/ActionManager</i>	Diese Datei konfiguriert Action Manager.
am_logging.properties	<i>tdi-ausgangsverzeichnis/ActionManager</i>	Diese Datei konfiguriert die Protokollierung von Action Manager.
ibmdiservice.props	<i>tdi-ausgangsverzeichnis/win32_service</i>	Diese Datei konfiguriert den Windows-Dienst.
mqeconfig.props	<i>tdi-ausgangsverzeichnis/jars/plugins/</i>	Diese Datei ermöglicht die Konfiguration des IBM WebSphere MQ Everyplace-Service. In IBM Security Directory Integrator können Sie auf IBM WebSphere MQ Everyplace unter Verwendung der Authentifizierung für den IBM WebSphere MQ Everyplace-Minizertifikatsserver zum Ausstellen von Zertifikaten zugreifen. Die Zertifikate werden anschließend für die Authentifizierung verwendet. Bei der Authentifizierung müssen zusätzliche, in IBM Security Directory Integrator verfügbare Eigenschaften zur Datei mqeconfig.props hinzugefügt werden.

Anmerkung: Die Datei registry.txt kann mit dem Tool "cryptoutil" verschlüsselt und entschlüsselt werden (siehe „Verschlüsselungsdienstprogramm von IBM Security Directory Integrator“ auf Seite 153). Das Tool "cryptoutil" sollte nicht auf die Datei global.properties oder solution.properties angewendet werden. Sie können einzelne Eigenschaftswerte, nicht jedoch die gesamte Eigenschaftendatei verschlüsseln.

Tabelle 20. Tabelle der zuvor genannten Eigenschaften, ihrer Kenndaten, ihrer Funktionsweise, ihrer möglichen Werte und ihrer Verwendung

Name	Gültige Werte	Beschreibung
com.ibm.di.server.securemode	true/false	Der Aktivierungs-/Inaktivierungsswitch für den gesicherten Modus.

Tabelle 20. Tabelle der zuvor genannten Eigenschaften, ihrer Kenndaten, ihrer Funktionsweise, ihrer möglichen Werte und ihrer Verwendung (Forts.)

Name	Gültige Werte	Beschreibung
api.keystore	Dateiname	Der für SSL-Zertifikate verwendete Serverschlüsselspeicher. Zuvor com.ibm.di.server.keystore.
api.key.alias	Schlüsselaliasname	Der Schlüsselaliasname aus dem Schlüsselspeicher für SSL-Zertifikate. Zuvor com.ibm.di.server.key.alias.
{protect}-api.keystore.password	Schlüsselspeicherkenntwort für SSL	Das Schlüsselspeicherkenntwort für SSL. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
{protect}-api.key.password	Schlüsselkenntwort für SSL	Das Schlüsselkenntwort für SSL. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.server.encryption.keystore	Dateiname	Die Datenverschlüsselung für den Schlüsselspeicher, in dem sich der vom Server verwendete Schlüssel befindet. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.server.encryption.key.alias	Schlüsselaliasname	Der Schlüsselaliasname des Schlüsselspeichers für die Verschlüsselung. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.server.encryption.keystoretype	Schlüsselspeichertyp, also "JKS", "JCEKS" usw.	Der Typ des Schlüsselspeichers, in dem sich der Verschlüsselungsschlüssel des Servers befindet. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.server.encryption.transformation	"RSA" oder Umsetzung mit geheimem Schlüssel	Die zur Verschlüsselung verwendete Umsetzung des Servers. Diese Eigenschaft kann entweder auf "RSA" (Verschlüsselung mit öffentlichem Schlüssel) oder auf eine Umsetzung mit geheimem Schlüssel gesetzt sein. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
api.on	true/false	Der Aktivierungs-/Inaktivierungsswitch für die Server-API.
api.user.registry	Dateiname	Die Benutzerregistrydatei für die Server-API.
api.user.registry.encryption.on	true/false	Der Switch für eine verschlüsselte oder unverschlüsselte Benutzerregistry.

Tabelle 20. Tabelle der zuvor genannten Eigenschaften, ihrer Kenndaten, ihrer Funktionsweise, ihrer möglichen Werte und ihrer Verwendung (Forts.)

Name	Gültige Werte	Beschreibung
api.remote.on	true/false	Der Aktivierungs-/Inaktivierungsswitch für die ferne Server-API. Die Standardeinstellung ist "true".
api.remote.ssl.on	true/false	Der Switch, mit dem angegeben wird, ob SSL für die ferne Server-API erforderlich ist oder nicht.
api.remote.ssl.client.auth.on	true/false	Der Switch, mit dem angegeben wird, ob die SSL-Clientauthentifizierung für die ferne Server-API erforderlich ist oder nicht.
api.truststore	Dateiname	Der Server-Truststore.
api.truststore.pass	*	Das Kennwort für den Truststore.
api.remote.nonssl.hosts		Nicht-SSL-Adressen für das Akzeptieren von IP-Verbindungen ohne SSL.
api.custom.method.invoke.on	true/false	Server-API-Methoden für den angepassten Methodenaufruf. Der Wert "true" bedeutet, dass die Verwendung zulässig ist, bei "false" ist sie nicht zulässig.
api.custom.method.invoke.allowed.classes		Die Server-API-Klassen, die durch die Server-API-Methoden für den Aufruf custommethod direkt aufgerufen werden können.
api.custom.authentication	Scriptdateiname oder "[ldap]/[jaas]" für die integrierte LDAP- oder JAAS-Authentifizierung.	Die angepasste Authentifizierungsmethode.
api.custom.authentication.ldap.*		Die Gruppe der Eigenschaften für die LDAP-Authentifizierungskonfiguration.
javax.net.ssl.*		Die JSSE-Standardeigenschaftengruppe für den Schlüsselspeicher, den Truststore und deren Kennwörter.
com.ibm.di.server.pkcs11	false	Gibt an, ob mit PKCS11 kompatible Verschlüsselungseinheiten für SSL erforderlich sind oder nicht. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
{protect}-com.ibm.di.server.pkcs11.pass	administrator	Das Zugriffskennwort für die mit PKCS11 kompatible Verschlüsselungseinheit. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.

Tabelle 20. Tabelle der zuvor genannten Eigenschaften, ihrer Kenndaten, ihrer Funktionsweise, ihrer möglichen Werte und ihrer Verwendung (Forts.)

Name	Gültige Werte	Beschreibung
com.ibm.di.server.pkcs11.accl	false	Wenn diese Eigenschaft auf "true" gesetzt ist, müssen Hardware-verschlüsselungseinheiten für die Verschlüsselung verwendet werden.

Anmerkung: Alle in der obigen Tabelle aufgeführten Eigenschaften können in der Konfigurationsdatei `global.properties` festgelegt und durch Verwendung des Präfixes `{protect}`- mit einer Verschlüsselung geschützt werden (Details können Sie unter „Standardverschlüsselung der Datei 'global.properties' oder 'solution.properties'“ auf Seite 152 nachlesen).

Sicherheit für die Webverwaltungskonsole

Unter dem hier aufgeführten Link erhalten Sie Informationen zur Sicherheit für die Webverwaltungskonsole.

Informationen hierzu finden Sie unter „Sicherheit bei AMC und Action Manager“ auf Seite 283.

Sonstige Sicherheitsaspekte

Hier werden die verschiedenen Sicherheitsaspekte aufgeführt.

HTTP-Basisauthentifizierung

Sie können eine Komponente mit HTTP-Basisauthentifizierung authentifizieren. Prüfen Sie dazu die aufgeführten Parameter.

Bei einigen IBM Security Directory Integrator-Komponenten haben Sie die Möglichkeit, die HTTP-Basisauthentifizierung als Authentifizierungsmechanismus zu verwenden. Wie der Name andeutet, handelt es sich hierbei um eine einfache Authentifizierung. Die HTTP-Basisauthentifizierung sollte bei allen strengen Sicherheitsdefinitionen nicht als sicher eingestuft werden, da die Berechtigungsnachweise durch ihre Base64-Codierung ohne großen Aufwand entschlüsselt werden können. Zum Schutz der Daten sollten Sie komplexere Schemas verwenden (beispielsweise eine Kombination aus aktiviertem SSL und HTTP-Basisauthentifizierung). Falls die Komponente die HTTP-Basisauthentifizierung unterstützt, werden die folgenden Parameter angezeigt:

Authentifizierungsmethode

Gibt den Typ der HTTP-Authentifizierung an. Falls der Typ der HTTP-Authentifizierung auf `Anonym` gesetzt ist, findet keine Authentifizierung statt. Ist HTTP-Basisauthentifizierung angegeben, wird die HTTP-Basisauthentifizierung mit dem Benutzernamen und dem Kennwort verwendet, die in den Parametern `Benutzername` und `Kennwort` angegeben sind.

Spezielle Aspekte für SSL bei Lotus Domino

Nachstehend werden die SSL-Spezifikationen bei Lotus Domino aufgeführt.

Die Domino-APIs für SSL verwenden nicht JSSE, sondern sind stattdessen Domino-spezifisch. Dies bedeutet, dass der Truststore und der Schlüsselspeicher von IBM

Security Directory Integrator (siehe „SSL-Konfiguration von IBM Security Directory Integrator-Komponenten als Client“ auf Seite 113) bei der SSL-Konfiguration für den Domino-Änderungserkennungsconnector keine Rolle spielen. Für die SSL-Konfiguration des Domino-Änderungserkennungsconnectors wird eine Datei namens "TrustedCerts.class" verwendet. Diese Datei wird bei jedem Start des Prozesses DI-IOP (im Domino-Server) generiert und muss im Klassenpfad von IBM Security Directory Integrator enthalten sein (also in den Shell-Scripts *ibmdisrv* oder *ibmditk*, die den IBM Security Directory Integrator-Server bzw. den Konfigurationseditor von IBM Security Directory Integrator starten). Sie müssen die Datei "TrustedCerts.class" in einen lokalen Pfad kopieren, der in der CLASSPATH-Angabe enthalten ist, oder "Lotus\Domino\Data\Domino\Java" Ihrer Domino-Installation muss im Klassenpfad enthalten sein. Für diesen Connector ist es ohne Belang, ob der IBM Security Directory Integrator-Truststore oder -Schlüsselspeicher in der Datei `global.properties` (bzw. `solution.properties`) festgelegt ist oder nicht.

Anmerkung: Die vorstehenden Informationen beziehen sich auf die Konfiguration von SSL für den Notes-Connector und den Domino-Änderungserkennungsconnector, da diese SSL über IOP verwenden.

Zertifikate für Web-Service-Suite von IBM Security Directory Integrator

Mithilfe der nachstehenden Anweisungen und des bereitgestellten Beispiels können Sie dem Zertifikat für die Web-Service-Suite von IBM Security Directory Integrator einen Namen geben.

Der Teil "cn=" des definierten Namens (dn) eines Zertifikats, das mit den Server-Connectors für Web-Services in IBM Security Directory Integrator zu verwenden ist, muss mit dem DNS-Namen oder der IP-Adresse des Host-Computers übereinstimmen, auf dem IBM Security Directory Integrator ausgeführt wird. Andernfalls wird eine Ausnahmebedingung ausgelöst, weil der Client keine SSL-Verbindung zum Server-Connector von IBM Security Directory Integrator für Web-Services aufbauen kann. Beispiel für den Teil "cn=" des definierten Namens eines Zertifikats: `cn=www.myserver.com`. (Diese Einschränkung für den definierten Namen im Zertifikat des Servers ist auf das HTTP-Protokoll zurückzuführen; siehe RFC2818 "HTTP over TLS".

Anmerkung: Falls IBM Security Directory Integrator sowohl ein Client- als auch ein Serverzertifikat verwenden muss und nur das in der Datei `global.properties` oder `solution.properties` konfigurierte Standardzertifikat verwendet wird, muss es identisch sein. Alternativ kann eine angepasste Implementierung der Java-Klasse "SSLSocket" oder "SSLServerSocket" geschrieben werden, die ein vom Standard abweichendes Zertifikat verwendet.

Beispiel für Serverzertifikaterstellung

Die folgende Befehlszeile erstellt ein selbst signiertes Serverzertifikat im Schlüsselspeicher namens "MyServerKeyStore.jks".

```
keytool -alias MyServerCertAlias -keyalg RSA -genkey -dname cn=<server_ip_address>
-validity 365 -keystore MyServerKeyStore.jks -storepass mystorepass -keypass mykeypass
```

Das erstellte Zertifikat hat den Aliasnamen "MyServerCertAlias". Zur Erstellung des Schlüsselpaares wird der RSA-Algorithmus verwendet. Der definierte Name des Zertifikats ist die IP-Adresse des Servers. Das Zertifikat ist 365 Tage (also 1 Jahr) gültig. Das Kennwort des Schlüsselspeichers ist "mystorepass". Das Kennwort des erstellten privaten Schlüssels lautet "mykeypass". Das erstellte Zertifikat kann

anschließend für die Verwendung konfiguriert werden, indem die folgenden Eigenschaften in der Datei `global.properties` oder `solution.properties` definiert werden:

```
api.key.alias=MyServerCertAlias  
api.keystore=MyServerKeyStore.jks
```

IBM WebSphere MQ Everyplace-Authentifizierung mit Minizertifikaten

Nachstehend finden Sie Anweisungen zur IBM WebSphere MQ Everyplace-Authentifizierung mit Minizertifikaten.

IBM WebSphere MQ Everyplace-Komponenten für IBM Security Directory Integrator können implementiert werden, um den über IBM WebSphere MQ Everyplace-Minizertifikate authentifizierten Zugriff zu nutzen. Zur Verwendung dieser IBM WebSphere MQ Everyplace-Komponenten muss IBM WebSphere MQ Everyplace 2.0.1.7 und IBM WebSphere MQ Everyplace Server Support ES06 heruntergeladen und installiert werden. Die Verwendung eines über Zertifikate authentifizierten Zugriffs verhindert, dass der Warteschlangenmanager oder eine Anwendung eines anonymen IBM WebSphere MQ Everyplace-Clients eine Kennwortänderungsanforderung an den IBM WebSphere MQ Everyplace-Kennwortspeicherconnector übergeben kann.

Weitere Informationen zur Konfiguration der IBM WebSphere MQ Everyplace-Authentifizierung mit Minizertifikaten finden Sie in dem entsprechenden Abschnitt in der Veröffentlichung *Plug-ins für den Kennwortabgleich*.

Kapitel 7. Regelsteuerkomponente für Verbindungswiederherstellung

Nachstehend erhalten Sie Informationen zur Regelsteuerkomponente für die Verbindungswiederherstellung.

Einführung

Der IBM Security Directory Integrator-Server unterstützt Regeln für die Verbindungswiederherstellung, die während der Aktivitätszeit eines Connectors auf bestimmte Fehlersituationen angewendet werden. Der Server ergreift die in Regeln niedergelegten Maßnahmen gemäß den Bedingungen, die bei der Kommunikation mit Zielsystemen auftreten.

Die Fertigungslinie fragt die Regelsteuerkomponente für die Verbindungswiederherstellung immer dann ab, wenn ein Connector eine Ausnahmebedingung ausgibt, und die Steuerkomponente empfiehlt für die aktuelle Situation eine Aktionsfolge. Der Fertigungsliniencode agiert dann in der vorgeschlagenen Weise.

Mögliche Aktionen:

- Verbindung wiederherstellen
- Ausnahmebedingung nicht bearbeiten und durch weitere Fehlermechanismen wie beispielsweise Fehlerhooks verarbeiten lassen

Die Aktion *reconnect* (Verbindung wiederherstellen) führt nur dann zu einem Verbindungswiederherstellungsversuch, wenn die Verbindungswiederherstellung durch die Optionen aktiviert ist, die auf der Registerkarte "Verbindungsfehler" für den Connector im Konfigurationseditor verfügbar sind. Falls die Verbindungswiederherstellung in dieser Konfiguration nicht aktiviert ist, wird im Fehlerfall ungeachtet der Entscheidung der Regelsteuerkomponente für die Verbindungswiederherstellung nicht versucht, die Verbindung wiederherzustellen.

Im Rahmen der Verbindungswiederherstellung wird der Connector automatisch erneut gestartet und (bei entsprechender Konfiguration) an seine vorherige Position gebracht. Hierzu wird zunächst die Beendigung und anschließend die Initialisierung für den Connector ausgeführt. Bei Iteratorconnectors werden optional Einträge übersprungen, bis die Position vor der Verbindungswiederherstellung erreicht ist. Bei jedem Verbindungswiederherstellungsversuch wird der entsprechende Hook für die Verbindungswiederherstellung aufgerufen. Das Script im Hook kann die Konfiguration später ändern, sodass eine nachfolgende Verbindungswiederherstellung erfolgreich verläuft. Wenn der Benutzer eine Funktionsübernahme angegeben hat, wird der Versuch einer automatische Funktionsübernahme oder eines Failbacks versucht, wenn Versuche einer Verbindungswiederherstellung fehlschlagen.

Die Aktion *error* (Fehler) impliziert, dass keine automatische Verbindungswiederherstellung versucht wird und dass die entsprechenden Fehlerhooks aufgerufen werden. Die Hooks können später eine angepasste Fehlerbehebung oder eine Fehlerprotokollierung durchführen.

Regeln für die Verbindungswiederherstellung

Nachstehend erhalten Sie Informationen zu Regeltypen, Abschnitten von Regeln, Abschnitten von Fehlersituationen sowie zu verschachtelten Ausnahmebedingungen.

Die Regelsteuerkomponente für die Verbindungswiederherstellung trifft anhand von konfigurierten Regeln Entscheidungen. Jede Regel beschreibt, welche Aktion ausgeführt werden soll, wenn eine bestimmte Fehlersituation eintritt. Die Steuerkomponente verwendet zwei Typen von Regeln:

- **Integrierte Regeln:** Diese Regeln sind in den Dateien `tdi.xml` für jede Connectordatei gespeichert und im Paket der JAR-Datei des Connectors enthalten. Infolgedessen sind diese Regeln für die jeweilige Connectorklasse immer spezifisch und ergeben für alle Connectornamen eine Übereinstimmung. Diese Regelliste ist die Standardliste der Regelsteuerkomponente für die Verbindungswiederherstellung, wenn eine Fehlersituation für einen bestimmten Connector verarbeitet wird. Falls Sie eigene Connectors in Java programmiert haben, finden Sie im Abschnitt zur Definition der Regeln zum Wiederherstellen einer Connectorverbindung des Anhangs zur Implementierung eigener Komponenten in Java zum Abschnitt *Referenzinformationen* im IBM Knowledge Center for IBM Security Directory Integrator Informationen dazu, wie Sie eigene integrierte Regeln erstellen können.
- Aus Gründen der Kompatibilität mit früheren Releases von IBM Security Directory Integrator fügt die Regelsteuerkomponente für die Verbindungswiederherstellung bei ihrer Einrichtung implizit eine Gruppe von Regeln zu den integrierten Regeln hinzu, die bei allen Ausnahmebedingungen des Typs "IOException" und "CommunicationException" den Versuch einer Verbindungswiederherstellung vorschreiben (`java.io.IOException` und `javax.naming.CommunicationException`).
- **Benutzerdefinierte Regeln:** Diese Regeln werden aus einer externen Textdatei namens `etc/reconnect.rules` geladen. Diese Regelliste überschreibt die integrierten Regeln. Informationen hierzu finden Sie unter „Konfiguration von benutzerdefinierten Regeln“ auf Seite 172.

Jede Regeln gilt für bestimmte Connectors und bestimmte Fehlersituationen.

Eine Regel setzt sich aus den folgenden Bestandteilen zusammen:

- **Connectorklasse:** Die Java-Klasse der Connectors, für die die Regel gilt.
- **Connectorname:** Der Name der Connectorkomponente, wie er in der Konfigurationsdatei der gegenwärtig ausgeführten Lösung angegeben ist.
- **Ausnahmebedingungsklasse:** Die Basisklasse der Ausnahmebedingungen, für die die Regel gilt.
- **Regulärer Ausdruck:** Ein regulärer Ausdruck, der mit den Nachrichten der Ausnahmebedingungen übereinstimmt, für die die Regel gilt.
- **Aktion:** Die durch die Regel vorgeschriebene Aktion. Zulässige Aktionen sind *error* (Fehler) oder *reconnect* (Verbindung wiederherstellen).

Eine Fehlersituation wird durch die folgenden Elemente beschrieben:

- **Connectorklasse:** Die Klasse des Connectors, der die Ausnahmebedingung auslöst.
- **Connectorname:** Der Name des Connectors, der die Ausnahmebedingung auslöst.

- **Ausnahmebedingung:** Die durch den Connector ausgelöste Ausnahmebedingung. Es handelt sich hierbei um eine Unterklasse von `java.lang.Throwable`.

Eine Regel wird auf eine Fehlersituation angewendet, falls gleichzeitig alle folgenden Bedingungen erfüllt sind:

- Die Regel gilt für den Connector in der Fehlersituation (Unterklassen der in der Regel beschriebenen Connectorklasse ergeben ebenfalls eine Übereinstimmung).
- Die Regel gilt für den Namen des Connectors, der die Fehlersituation verursacht hat.
- Die Ausnahmebedingung ist eine Instanz der Ausnahmebedingungsklasse, für die die Regel gilt.
- Die Regel enthält keinen regulären Ausdruck für den Abgleich der Ausnahmebedingungsnachricht oder der reguläre Ausdruck stimmt mit der Nachricht der Ausnahmebedingung überein.

Wenn eine bestimmte Fehlersituation auftritt, ermittelt die Regelsteuerkomponente für die Verbindungswiederherstellung diejenige Regel, die am genauesten mit der Fehlersituation übereinstimmt. Zunächst durchsucht die Steuerkomponente die benutzerdefinierten Regeln. Wird keine übereinstimmende Regel gefunden, werden die integrierten Regeln durchsucht. Wenn weiterhin keine übereinstimmende Regel gefunden wird, schreibt die Steuerkomponente die Standardaktion "error" (Fehler) vor. Falls in den benutzerdefinierten Regeln eine übereinstimmende Regel gefunden wird, werden die integrierten Regeln nicht durchsucht. Dies gilt auch dann, wenn sich unter den integrierten Regeln eine spezifischere Regel befindet.

Anmerkung: Falls zwei oder mehr Regeln mit einer Fehlersituation übereinstimmen, wird die am meisten spezifische Regel ausgewählt. Gibt es mehrere solcher Regeln, von denen keine spezifischer als die anderen ist, wird die erste Regel in der Liste ausgewählt. Aus diesem Grund ist die Reihenfolge der Regeln in der Regelliste von Bedeutung. Beispiel: Die folgenden Regeln sind vorhanden (die dargestellte Syntax ist eine Pseudosyntax, die nur zur Verdeutlichung verwendet wird):

```
...exceptionClass = "java.io.IOException", exceptionMessageRegExp = ".*", action = "error"...
...exceptionClass = "java.io.IOException", exceptionMessageRegExp = "\\w*", action = "reconnect"...
```

Falls eine Ausnahmebedingung des Typs "java.io.Exception" mit der Nachricht "problem" ausgelöst wird, wird die erste Regel ausgewählt, obwohl beide Regeln mit dem Fehler übereinstimmen, weil keine Regel spezifischer als die andere Regel ist (die Ausgabe des Abgleichs für den regulären Ausdruck wird nicht in die Gewichtung einbezogen).

Verschachtelte Ausnahmebedingungen

Einige Ausnahmebedingungen sind in anderen Ausnahmebedingungen verschachtelt. Wenn die Regelsteuerkomponente für die Verbindungswiederherstellung eine Liste von Regeln durchsucht (z. B. die integrierten Regeln), sucht die Steuerkomponente zuerst nach einer Regel, die mit der übergeordneten Ausnahmebedingung übereinstimmt. Falls keine übereinstimmende Regel gefunden wird, durchsucht die Steuerkomponente dieselbe Liste noch einmal, ermittelt jedoch dieses Mal, ob eine Übereinstimmung für die verschachtelte Ausnahmebedingung gefunden werden kann (falls die übergeordnete Ausnahmebedingung keine verschachtelte Ausnahmebedingung enthält, wird diese Suche übersprungen). Bitte beachten Sie, dass die Regelsteuerkomponente für die Verbindungswiederherstellung nur versucht, die Ausnahmebedingung der ersten Verschachtelungsebene abzugleichen. Falls weitere Verschachtelungsebenen von Ausnahmebedingungen vorhanden sind, werden sie ignoriert.

Anmerkung: Die automatische Funktionsübernahme ist für Connectors im Servermodus nicht möglich.

Konfiguration von benutzerdefinierten Regeln

Beachten Sie beim Definieren einer Regel deren Format sowie die übrigen hier aufgeführten wichtigen Informationen. Außerdem finden Sie hier einige Beispiele.

Die Liste der benutzerdefinierten Regeln wird in einer Textdatei namens `reconnect.rules` definiert. Diese Datei befindet sich im Unterordner "etc" des Lösungsordners von IBM Security Directory Integrator (bzw. des Installationsordners von IBM Security Directory Integrator, falls kein Lösungsordner definiert wurde). Jede Regel ist in einer einzigen Zeile angegeben. Das Format einer Regel lautet folgendermaßen:

```
<connectorklasse>:<connectorname>:<ausnahmebedingungsklasse>:<aktion>:<regulärer_ausdruck>
```

Hierbei gilt Folgendes:

- `<connectorklasse>` steht für den vollständig qualifizierten Namen der Java-Klasse des Connectors.
- `<connectorname>` steht für den Namen des Connectors, wie er in die Fertigungslinie eingefügt ist.
- `<ausnahmebedingungsklasse>` ist der vollständig qualifizierte Name der Java-Klasse der Ausnahmebedingung.
- `<aktion>` steht entweder für "error" (Fehler) oder "reconnect" (Verbindung wiederherstellen).
- `<regulärer_ausdruck>` steht für einen regulären Java-Ausdruck, wie er in der Javadoc der Klasse `java.util.regex.Pattern` unter <http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html> beschrieben ist.

Anmerkung:

1. Außer der Aktion kann jeder Bestandteil leer sein. Falls ein Teil leer ist, bedeutet dies eine Übereinstimmung mit allen Vorkommen.
2. Jeder Bestandteil ist verbindlich. Dies bedeutet, dass die umgebenden Doppelpunkte auch dann vorhanden sein müssen, wenn der Teil leer ist. (Infolgedessen muss jede Zeile mindestens 4 Doppelpunkte enthalten. Jeder Doppelpunkt trennt zwei angrenzende Teile der Regel. Der Wert 4 ist ein Mindestwert, weil der reguläre Ausdruck ebenfalls Doppelpunkte enthalten kann. Diese Doppelpunkte kollidieren nicht mit der Syntexanalyse der Regel, da der reguläre Ausdruck in einer Regel an letzter Stelle angegeben wird.)
3. Redundante Leerzeichen sind nicht zulässig.
4. Der reguläre Ausdruck beginnt unmittelbar nach dem vierten Doppelpunkt und erstreckt sich bis zum Ende der Zeile.
5. Die Datei mit den benutzerdefinierten Regeln ist keine Java-Eigenschaftendatei. Hauptgrund hierfür ist, dass der Schlüssel für eine Regel alle Regelteile (mit Ausnahme der Aktion "reconnect") enthalten muss, um eindeutig zu sein. Der einzige Nutzen, der sich aus der Verwendung des Java-Eigenschaftenmechanismus ergeben würde, wäre somit die Abtrennung der Aktion von den anderen Regelteilen. Dies wäre jedoch mit dem Nachteil verbunden, dass für Leerzeichen, Doppelpunkte und Gleichheitszeichen (die Voraussetzungen für einen gültigen Eigenschaftsschlüssel sind) Escapezeichen verwendet werden müssten. Auch wenn das Java-Eigenschaftenframework verwendet würde, wäre weiterhin eine angepasste Syntexanalyse des Eigenschaftsschlüssels erforderlich, um die Regelteile aus ihm zu extrahieren.

6. Der reguläre Ausdruck (nicht die Aktion "reconnect") steht in jeder Zeile an letzter Stelle. Dieses Muster wurde gewählt, damit im regulären Ausdruck für Doppelpunkte (die als Begrenzer der Regelteile gelten) keine Escapezeichen verwendet werden müssen.
7. Der reguläre Ausdruck muss mit dem gesamten Nachrichtentext übereinstimmen. Nehmen wir beispielsweise an, dass der Nachrichtentext, den Sie abgleichen wollen, an einer beliebigen Stelle die Wörter "Irgendein Fehler" enthält. Ein passender regulärer Ausdruck könnte dann folgendermaßen lauten:

```
.*Irgendein Fehler.*
```

Das Zeichen "." stimmt mit jedem Zeichen (außer einer Zeilenschaltung) überein, der Änderungswert * gibt 0 oder mehr an. Nehmen wir nun an, die Nachricht endet mit einer neuen Zeile. Wenn dies der Fall ist, würde der obige reguläre Ausdruck keine Übereinstimmung ergeben. Stattdessen könnte ein regulärer Ausdruck wie beispielsweise der Folgende verwendet werden:

```
.*Irgendein Fehler.*\r?\n?
```

Die Zeichenfolgen "\r" und "\n" geben Rücklauf- und Zeilenvorschubzeichen an, der Änderungswert ? gibt 0 oder 1 Vorkommen an.

8. Sie müssen die Verbindungswiederherstellung weiterhin in der Konfiguration des Connectors konfigurieren. Entsprechende Informationen enthält der Abschnitt „Allgemeine Konfiguration der Verbindungswiederherstellung“ auf Seite 174.

Beispiele

Für das Beispiel werden zwei Regeln verwendet:

```
com.ibm.di.connector.ReconnectTestConnector::myconnname:java.io.IOException:error:.*\Wfatal\W.*  
::java.io.IOException:reconnect:
```

Verbindung mit JDBC-Connector wiederherstellen

Der JDBC-Connector von IBM Security Directory Integrator ist im Iteratormodus für die Iteration einer Tabelle aus DB2 konfiguriert und für die Funktion zur Verbindungswiederherstellung aktiviert. Zum Zeitpunkt der Lösungsausführung ist die DB2-Instanz jedoch noch nicht gestartet worden. Damit die Verbindungswiederherstellung funktioniert, müssen die folgenden Ausnahmedetails in der Datei `reconnect.rules` angegeben werden:

```
com.ibm.di.connector.JDBCConnector::com.ibm.db2.jdbc.DB2Exception:reconnect:
```

Verbindung mit RAC-Connector wiederherstellen

Dieser Connector ist veraltet und wird in einer zukünftigen Version von IBM Security Directory Integrator entfernt.

Der RAC-Connector von IBM Security Directory Integrator ist im Iteratormodus konfiguriert und für die Funktion zur Verbindungswiederherstellung aktiviert. Falls der Server des Agentencontrollers inaktiv ist, müssen die folgenden Ausnahmedetails in der Datei `reconnect.rules` angegeben sein, damit der RAC-Connector versucht, die Verbindung wiederherzustellen:

```
com.ibm.di.connector.RACConnector::org.eclipse.tptp.platform.execution.exceptions.AgentControllerUnavailableException:reconnect:
```

Hinweise zu Ausnahmedingungen

Beachten Sie bei der Arbeit mit Ausnahmedingungen die nachstehenden Hinweise.

Jede Umgebung und jede Lösung, die für eine bestimmte Umgebung mit IBM Security Directory Integrator erstellt wird, ist normalerweise eindeutig. Benutzerdefinierte Regeln sind kundenspezifisch. Die Funktionalität wird zur Verfügung gestellt, damit Lösungen basierend auf den für die Umgebung oder Lösung spezifischen Ausnahmebedingungen automatisch versuchen können, die Verbindung wiederherzustellen. Informationen zu speziellen Ausnahmebedingungen, die durch die IBM Security Directory Integrator-APIs für jeden Connector zurückgegeben werden, enthält die Java-API-Dokumentation von IBM Security Directory Integrator.

Zusätzlich sind einige IBM Security Directory Integrator-Komponenten auf zugrunde liegende Bibliotheken angewiesen und die APIs dieser Bibliotheken lösen für bestimmte Situationen Ausnahmebedingungen aus. In der folgenden Liste sind einige zentrale IBM Security Directory Integrator-Komponenten angegeben, in denen Sie nach zusätzlichen Informationen zu Ausnahmebedingungen und zu den möglichen Ursachen der Ausnahmebedingungen suchen können. Diese Informationen sind bei der Entscheidung von Nutzen, ob Sie für bestimmte mögliche Ausnahmebedingungen angepasste Regeln für die Verbindungswiederherstellung erstellen wollen:

- LDAP-Connector: Der LDAP-Connector ist von den mit der JRE ausgelieferten JNDI-Bibliotheken abhängig. Weitere Informationen zur JNDI-Schnittstelle, ihren APIs und den möglicherweise von ihr ausgelösten Ausnahmebedingungen finden Sie unter der Adresse <http://java.sun.com/j2se/1.5.0/docs/api/javax/naming/package-summary.html>.
- JDBC-Connector: Der JDBC-Connector ist vom konfigurierten JDBC-Treiber abhängig. Weitere Informationen zu den Ausnahmebedingungen, die ausgelöst werden können, enthalten die Java-API-Dokumentation oder das Referenzmaterial für den konfigurierten JDBC-Treiber. Der Unterabschnitt "Understanding JDBC Drivers" des Abschnitts über den JDBC-Connector im Handbuch *Referenzinformationen* enthält Links zur Dokumentation für eine Reihe von häufig verwendeten JDBC-Treibern.

Allgemeine Konfiguration der Verbindungswiederherstellung

Nachstehend sind die Konfigurationsoptionen für die Regel zur Verbindungswiederherstellung aufgeführt.

Die Angabe einer Regel für die Verbindungswiederherstellung ist erforderlich, damit versucht wird, eine Verbindung wiederherzustellen. Die Regel allein ist jedoch nicht ausreichend. Eine weitere Voraussetzung besteht darin, dass die Verbindungswiederherstellung in der allgemeinen Konfiguration der Verbindungswiederherstellung aktiviert ist. Dies kann auf der Registerkarte **Verbindungsfehler** im Konfigurationseditor vorgenommen werden. Falls die Verbindungswiederherstellung in dieser Konfiguration nicht aktiviert ist, wird im Fehlerfall ungeachtet der Entscheidung der Regelsteuerkomponente für die Verbindungswiederherstellung nicht versucht, die Verbindung wiederherzustellen. Die folgende Liste enthält die Konfigurationsoptionen:

Anzahl der Wiederholungen

Hier ist angegeben, wie häufig versucht wird, die Verbindung zu wiederherzustellen, wenn ein Problem auftritt, bevor der Versuch abgebrochen wird. Falls später ein neues Problem auftritt, wird dieselbe Anzahl Versuche unternommen.

Verzögerung zwischen den Wiederholungen

Hier ist angegeben, wie viele Sekunden zwischen den einzelnen Versuchen für die Verbindungswiederherstellung und vor dem ersten Versuch verstreichen sollen.

Herstellen der Verbindung bei Anfangsverbindungsfehler erneut versuchen

Wenn dieser Parameter aktiviert ist und nach der Initialisierung des Connectors keine Verbindung aufgebaut werden kann, wird ein Verbindungswiederherstellungsversuch ausgeführt. Es handelt sich hierbei nicht um eine echte Verbindungswiederherstellung, da zuvor keine Verbindung aufgebaut wurde. Der Mechanismus ist jedoch generell identisch.

Verbindung bei Unterbrechung automatisch wiederherstellen

Wenn dieser Parameter aktiviert ist und die Verbindung nach der Initialisierung des Connectors unterbrochen wird, wird versucht, die Verbindung wiederherzustellen.

Automatisch vorwärts springen

Dieser Parameter legt fest, dass nach der Verbindungswiederherstellung automatisch so oft vorwärts gesprungen werden soll, dass die Anzahl der Anzahl der erfolgreichen Lesevorgänge entspricht.

Automatische Funktionsübernahme

Wenn dieser Parameter gesetzt ist, wird nach dem Fehlschlagen einer automatischen Verbindungswiederholung eine automatische Funktionsübernahme versucht.

Connector für Übernahme

Der Name des Connectors in der Ressourcenbibliothek, der für die automatische Funktionsübernahme verwendet wird.

Failback nach

Wenn dieses Feld auf einen positiven Wert gesetzt wurde, wird nach Ablauf der angegebenen Sekunden ein automatischer Failback versucht. Wenn der Failback fehlschlägt, wird erst nach einem erneuten Ablauf der angegebenen Sekunden ein neuer Versuch gestartet.

Anmerkung: Sowohl beim Parameter **Herstellen der Verbindung bei Anfangsverbindungsfehler erneut versuchen** als auch beim Parameter **Verbindung bei Unterbrechung automatisch wiederherstellen** ermittelt die Steuerkomponente für die Verbindungswiederherstellung, ob die Ausnahmebedingung zu einem Verbindungswiederherstellungsversuch führt oder ob es sich um einen allgemeineren Fehler handelt.

Kapitel 8. Systemwarteschlange

Mit der Systemwarteschlange können Sie Nachrichten speichern und zwischen IBM Security Directory Integrator-Servern und Fertigungslinien weiterleiten.

Die Systemwarteschlange ist ein IBM Security Directory Integrator-Subsystem für die JMS-Nachrichtenübertragung, das dem IBM Security Directory Integrator-Systemspeicher ähnelt. Die Systemwarteschlange unterstützt die Entwicklung von IBM Security Directory Integrator-Lösungen, bei denen eine asynchrone Übertragung benötigt wird, damit die Arbeit auf mehrere Fertigungslinien verteilt werden kann. Die Systemwarteschlange kann entweder IBM WebSphere MQ oder IBM WebSphere MQ Everyplace als Basis für das JMS-Nachrichtenübertragungssystem sowie jedes beliebige andere JMS-System verwenden (unter der Voraussetzung, dass der JMS-Scripttreiber dieses JMS-System korrekt adressieren kann).

Anmerkung: Der Systemwarteschlangenconnector (siehe *Referenzinformationen*) überträgt Daten nicht direkt an die Systemwarteschlange, sondern verwendet stattdessen die Server-API als Vermittlungsstelle.

In IBM Security Directory Integrator wird die Systemwarteschlange durch den Installationsprozess standardmäßig aktiviert.

Konfiguration der Systemwarteschlange

Nachstehend erhalten Sie Informationen zur Konfiguration der Systemwarteschlange und den erforderlichen Eigenschaften.

Die Systemwarteschlange wird mit den folgenden treiberspezifischen Java-Eigenschaften konfiguriert, die in der IBM Security Directory Integrator-Datei `global.properties` oder `solution.properties` angegeben werden:

systemqueue.on

Dieser Parameter gibt an, ob die Systemwarteschlange beim Start des IBM Security Directory Integrator-Servers gestartet und initialisiert werden soll. Gültige Werte sind `true` und `false`. Der Standardwert ist `true`.

systemqueue.jmsdriver.name

Dieser Parameter gibt den vollständig qualifizierten Namen der Java-Klasse an, die als JMS-Treiber für die Systemwarteschlange verwendet werden soll. Als Wert kann der Name einer vom Benutzer bereitgestellten Klasse oder eine der folgenden JMS-Standardtreiberimplementierungen von IBM Security Directory Integrator verwendet werden:

- `com.ibm.di.systemqueue.driver.ActiveMQ` (siehe „Apache ActiveMQ-Parameter“ auf Seite 178, standardmäßiger JMS-Provider)
- `com.ibm.di.systemqueue.driver.IBMMQe` (siehe „Parameter für IBM WebSphere MQ Everyplace“ auf Seite 180)
- `com.ibm.di.systemqueue.driver.IBMMQ` (siehe „Parameter für IBM WebSphere MQ“ auf Seite 180)
- `com.ibm.di.systemqueue.driver.IBMMB` (siehe „Parameter für Microbroker“ auf Seite 181)
- `com.ibm.di.systemqueue.driver.JMSScriptDriver` (anderes JMS-System, siehe hierzu „Parameter für JMS-Scripttreiber“ auf Seite 181)

Der Standardwert ist "com.ibm.di.systemqueue.driver.ActiveMQ".

Abhängig vom Parameter "systemqueue.jmsdriver.name" trifft einer der folgenden Abschnitte zu:

Apache ActiveMQ-Parameter

Nachstehend erhalten Sie Informationen zu Apache ActiveMQ-Parametern.

Um ActiveMQ für die Systemwarteschlange als JMS-Provider zu verwenden, müssen Sie die Eigenschaft `systemqueue.jmsdriver.name` in `global.properties/solution.properties` auf `com.ibm.di.systemqueue.driver.ActiveMQ` setzen. Der ActiveMQ-Treiber hat die folgenden Parameter:

- **jms**
oker - Die ActiveMQ-Serveradresse (Protokoll, IP-Adresse und TCP-Portnummer). Beispiel:
`tcp://localhost:6161` oder `ssl://localhost:616171` zur Verwendung der SSL-Verbindung.

Der Standardwert lautet folgendermaßen:

```
vm://localhost?brokerConfig=xbean:etc/activemq.xml
```

Dieser Wert führt ActiveMQ im eingebetteten Modus aus. Die Datei `etc/activemq.xml` enthält die ActiveMQ-Standardkonfiguration.

Anmerkung:

1. Der Pfad zur ActiveMQ-XML-Konfigurationsdatei (nach xbean:) darf keine Leerzeichen enthalten. Weitere Informationen finden Sie unter <https://issues.apache.org/activemq/browse/AMQ-1385>.
Wenn der Pfad Leerzeichen enthält, ist für jedes Leerzeichen eine dreimalige URL-Codierung erforderlich; dadurch wird eine Transformation in `%2520` erreicht.
2. Die Systemwarteschlange initialisiert ActiveMQ beim Start, wenn der Parameter **systemqueue.on=true** in der Datei `solution.properties` auf "true" gesetzt ist.

Konfiguration

Nachstehend erhalten Sie Informationen zur ActiveMQ-Konfiguration und den erforderlichen Parametern.

Die ActiveMQ-Konfiguration basiert auf der Datei `activemq.xml`, die sich an der Position `tdi-installationsordner/etc` befindet. Die ActiveMQ-Konfigurationsparameter sind folgende.

Broker

Dieser ActiveMQ-Nachrichtenbroker besteht aus Transportconnectors, Netzconnectors und Eigenschaften, die für die Konfiguration des Brokers verwendet werden. Die Attribute lauten wie folgt:

- `brokerName="localhost"` - Der Name des Brokers.
- `dataDirectory="./ActivemqDataStore"` - Das Verzeichnis, das für die Speicherung der Daten von ActiveMQ verwendet wird.
- `useShutdownHook="true"` - Legt fest, ob zum Schließen des Brokers, falls die JVM beendet wurde, ein Beendigungshandler verwendet wird.
- `useJmx="true"` - Legt fest, ob die Services des Brokers für die JMX zugänglich gemacht werden sollen.

managementContext

Dieser Parameter konfiguriert, wie die ActiveMQ in der JMX zugänglich gemacht wird. Die Attribute lauten wie folgt:

- createConnector="true" - Legt fest, ob die ActiveMQ ihren eigenen JMX-Connector erstellt.
- o connectorPort="1099" - Der Port des Connectors. Der Wert ist standardmäßig 1099.

persistenceAdapter/kahaDB

Dieser Parameter konfiguriert die Nachrichtenpersistenz für den Broker. Die Attribute lauten wie folgt:

- journalMaxFileLength="32mb" - Legt die maximale Größe der Nachrichtendatenprotokolle fest.
- checksumJournalFiles="true" - Erstellt eine Kontrollsumme für eine Journaldatei, um die Überprüfung der beschädigten Journals zu aktivieren.
- checkForCorruptJournalFiles="true" - Überprüft, falls aktiviert, beim Start beschädigte Journaldateien und versucht, diese wiederherzustellen.

transportConnectors

Dieser Parameter besteht aus Transportconnectors, bei denen die ActiveMQ empfangsbereit ist. Die Attribute lauten wie folgt:

- name="openwire" - Der Name des Transportconnectors.
- uri="tcp://localhost:61616" - Die Adresse des Transportconnectors.

Anmerkung: Weitere Informationen zu den in der XML-Konfigurationsdatei verwendeten XML-Objekten finden Sie in der Veröffentlichung "ActiveMQ's XBean XML Reference 5.0" unter <http://activemq.apache.org/xbean-xml-reference-50.html>.

Protokollierung

Die ActiveMQ basiert auf "log4j", um Informationen im Broker-Client und dem Broker zu protokollieren. Durch die folgenden Zeilen in der Datei "log4j.properties" wird die ActiveMQ-Protokollierung durch Setzen der Standardprotokollierungskategorien von ActiveMQ-Elementen konfiguriert.

- log4j.logger.org.apache.activemq=INFO
- log4j.logger.org.apache.activemq.spring=WARN
- log4j.logger.org.apache.activemq.web.handler=WARN
- log4j.logger.org.springframework=WARN
- log4j.logger.org.apache.xbean=WARN
- log4j.logger.org.apache.camel=ERROR

SSL mit ActiveMQ verwenden

Sie können die ActiveMQ für die Verwendung der SSL-Verbindung durch Angabe des Elements <sslContext> und der korrekten Transportconnector-URI in der XML-Konfigurationsdatei der ActiveMQ konfigurieren.

Die ActiveMQ basiert auf Zertifikaten für die Verwendung der SSL-Verbindung. Standardmäßig wird die ActiveMQ zur Wiederverwendung der IBM Security Directory Integrator-Server-API-Zertifikate im Ordner tdi-installationsordner/serverapi als keyStore und trustStore konfiguriert. Für die Wiederverwendung müssen die Namen der Client- und Serverschlüsselspeicherdateien im Element <sslContext> der Konfigurationsdatei von ActiveMQ angegeben werden. Beispiel:

```
<sslContext>
  <sslContext
    keyStore="file:./serverapi/testadmin.jks" keyStorePassword="administrator"
    trustStore="file:./serverapi/testadmin.jks" trustStorePassword="administrator"/>
</sslContext>
```

Dabei ist `testadmin.jks` der Name des IBM Security Directory Integrator-Zertifikats und `password` das Kennwort des IBM Security Directory Integrator-Zertifikats.

Anmerkung: Das Element `<sslContext>` sowie alle Parameter werden nur dann berücksichtigt, wenn die Eigenschaften vom Typ `javax.net.ssl` in der Datei `solution.properties` von IBM Security Directory Integrator nicht definiert sind. Standardmäßig verwendet ActiveMQ die `javax`-Eigenschaften aus der IBM Security Directory Integrator-API und nicht die Eigenschaften, die im Tag `<sslContext>` definiert sind.

Parameter für IBM WebSphere MQ Everyplace

Nachstehend erhalten Sie Informationen zu IBM WebSphere MQ Everyplace und den erforderlichen Parametern.

Damit IBM WebSphere MQ Everyplace als JMS-Provider für die Systemwarteschlange verwendet werden kann, muss ein Warteschlangenmanager für IBM WebSphere MQ Everyplace erstellt werden. Dies kann mit dem unter „Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace“ auf Seite 185 beschriebenen Dienstprogramm erfolgen, das im Produktpaket von IBM Security Directory Integrator enthalten ist.

`systemqueue.jmsdriver.param.mqe.file.ini`

Dieser für IBM WebSphere MQ Everyplace spezifische Parameter gibt den relativen Dateisystemdateinamen der IBM WebSphere MQ Everyplace-Initialisierungsdatei an. Diese Eigenschaft ist erforderlich und nur dann wirksam, wenn in der Eigenschaft `"systemqueue.jmsdriver.name"` der IBM WebSphere MQ Everyplace-JMS-Treiber angegeben ist. Der Standardwert ist `MQePWStore/pwstore_server.ini`. Dies ist die Standardposition für die IBM WebSphere MQ Everyplace-Initialisierungsdatei, die von dem unter „Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace“ auf Seite 185 beschriebenen Dienstprogramm erstellt wird.

Die Systemwarteschlange ist standardmäßig aktiviert. Falls Sie IBM WebSphere MQ Everyplace als Systemwarteschlange verwenden wollen, können Sie die folgende verkürzte Aktivierungsprozedur verwenden:

1. Setzen Sie die Eigenschaft `systemqueue.on` in der Datei `global.properties` oder `solution.properties` auf `true`.
2. Konfigurieren Sie IBM WebSphere MQ Everyplace durch den folgenden Aufruf:

```
cd lösungsverzeichnis
(Geben Sie bei Verwendung des Installationsverzeichnisses
cd tdi-installationsverzeichnis an.)
tdi-installationsverzeichnis/jars/plugins/mqeconfig.sh
tdi-installationsverzeichnis/jars/plugins/mqeconfig.props create server
(1 Zeile)
```

Parameter für IBM WebSphere MQ

Nachstehend erhalten Sie Informationen zu den Parametern für IBM WebSphere MQ.

Bei den folgenden Parametern handelt es sich um IBM WebSphere MQ-spezifische Parameter. Weitere Informationen zu diesen Parametern enthalten die Angaben über die Eigenschaften für die Initialisierung des MQ-JMS-Treibers im Abschnitt „Beispiel für Konfiguration der Systemwarteschlange“ auf Seite 184.

systemqueue.jmsdriver.param.jms

oker (IP-Adresse und TCP-Portnummer)

systemqueue.jmsdriver.param.jms.serverChannel

(Für die MQ-Serverinstanz definierter Serverkanal)

systemqueue.jmsdriver.param.jms.qManager

(Name des für die MQ-Serverinstanz definierten Warteschlangenmanagers)

systemqueue.jmsdriver.param.jms.sslCipher

(Name der Cipher Suite, die der beim Konfigurieren des MQ-Serverkanals ausgewählten Verschlüsselung entspricht, z. B. SSL_RSA_WITH_RC4128_MD5)

systemqueue.jmsdriver.param.jms.sslUseFlag

(Bei Anforderung einer SSL-Verbindung "true", andernfalls "false")

Parameter für Microbroker

Nachstehend erhalten Sie Informationen zu den Parametern für Microbroker.

Damit Microbroker (MB) als JMS-Provider für die Systemwarteschlange verwendet wird, muss die Eigenschaft `systemqueue.jmsdriver.name` in der Datei `global.properties` oder `solution.properties` auf den Wert `com.ibm.di.systemqueue.driver.IBMMB` gesetzt werden.

Der Microbroker-Treiber besitzt die folgenden Parameter (diese sind hier ohne das Präfix "systemqueue.jmsdriver.param." aufgeführt):

jms

oker Die Adresse des MB-Servers (IP-Adresse und TCP-Portnummer), z. B. "9.126.6.120:1883".

jms.clientID

Die Client-ID. Diese Eigenschaft ist erforderlich.

Anmerkung: Damit Microbroker als JMS-Provider für die Systemwarteschlange verwendet werden kann, werden einige JAR-Dateien von Microbroker benötigt. Eine Beispielliste der erforderlichen JAR-Dateien ist im Abschnitt "External System Configuration" für den Microbroker des JMS-Connectors im Handbuch *Referenzinformationen* verfügbar.

Parameter für JMS-Scripttreiber

Nachstehend erhalten Sie Informationen zu den Parametern für JMS-Scripttreiber.

Beim JMS-Treiber haben Sie die Möglichkeit, durch eine Scripterstellung in JavaScript Konnektivität für einen beliebigen JMS-Provider bereitzustellen, ohne hierzu Java-Code schreiben und erstellen zu müssen. Der JMS-Treiber fungiert als Brücke zwischen der Systemwarteschlange und einem benutzerdefinierten JavaScript-Teil im lokalen Dateisystem, der für die Erstellung eines Objekts `javax.jms.QueueConnectionFactory` oder eines Objekts `javax.jms.TopicConnectionFactory` zuständig ist. Diese Objekte werden mit einem providerspezifischen Verfahren abgerufen.

systemqueue.jmsdriver.param.js.jsfile

Dies ist ein spezieller Parameter für den JMS-Scripttreiber (er wird also be-

rücksichtigt, wenn die Eigenschaft **systemqueue.jmsdriver.name** auf `com.ibm.di.systemqueue.driver.JMSScriptDriver` gesetzt ist). Er gibt den Namen der Datei an, die den vom Benutzer bereitgestellten JavaScript-Code für die Verwendung des gewünschten JMS-Systems enthält. Weitere Informationen zu diesem Parameter enthalten die Angaben über die Einstellungen für den JMS-Treiber im Abschnitt „Beispiel für Konfiguration der Systemwarteschlange“ auf Seite 184. Bitte beachten Sie, dass die Namen der Java-Eigenschaften ohne das Präfix `systemqueue.jmsdriver.param.` angegeben sind.

systemqueue.jmsdriver.param.js.jsscript

Diese Eigenschaft gibt den Scripthauptteil an, der den JavaScript-Code für die Bildung einer Schnittstelle mit dem entsprechenden JMS-Provider enthält. Falls dieser Parameter nicht angegeben wird, wird der Parameter **systemqueue.jmsdriver.param.js.jsfile** zum Laden des auszuführenden JavaScripts verwendet.

systemqueue.jmsdriver.param.user.xxxx

Hierbei handelt es sich um benutzerdefinierte Eigenschaften, die von der Systemwarteschlange an die konfigurierte JMS-Treiberimplementierung übergeben werden. Falls beispielsweise die Eigenschaft

```
systemqueue.jmsdriver.param.user.my.prop1=myvalue1
```

festgelegt ist, erhält der konfigurierte JMS-Treiber eine Eigenschaft mit dem Namen `user.my.prop1` und dem Wert `myvalue1`.

systemqueue.auth.username

Dies ist der Benutzername, der durch die Systemwarteschlange für die Authentifizierung beim konfigurierten JMS-System verwendet wird. Wenn er nicht festgelegt ist, verwendet die Systemwarteschlange keine Authentifizierung beim konfigurierten JMS-System.

systemqueue.auth.password

Dies ist das Kennwort, das durch die Systemwarteschlange für die Authentifizierung beim konfigurierten JMS-System verwendet wird. Dieser Parameter wird nur dann verwendet, wenn der Parameter `systemqueue.auth.username` angegeben ist.

JavaScript-Objekt "env"

Nachstehend erhalten Sie Informationen zum JavaScript-Objekt "env".

Der JavaScript-Teil, der durch den JMS-Treiber ausgeführt wird, muss auf ein JavaScript-Objekt namens *env* zugreifen. Es handelt sich hierbei um ein Objekt des Typs "java.util.Hashtable", das providerspezifische Parameter für das Herstellen der Verbindung zum JMS-Provider enthält. Diese Parameter werden durch den JavaScript-Code verwendet, um auf die jeweilige JMS-Systemserverinstanz zuzugreifen.

Sie können in der Datei `global.properties` oder `solution.properties` mit dem Präfix `systemqueue.jmsdriver.param` angegeben werden. Falls für ein JMS-System beispielsweise ein URL-Parameter benötigt wird, kann die folgende Eigenschaft in der Datei `global.properties` oder `solution.properties` festgelegt werden:

```
systemqueue.jmsdriver.param.myjmssystem.url=myjmsserver.mydomain.com:12345
```

Diese Definition würde bewirken, dass sie von der Systemwarteschlange als Eintrag in der Hashtabelle "env" an den JavaScript-Code übergeben wird. Der Schlüssel des Eintrags wäre "myjmssystem.url" (die Systemwarteschlange entfernt das Präfix) und der Wert wäre "myjmsserver.mydomain.com:12345".

JavaScript-Objekt "ret"

Nachstehend erhalten Sie Informationen zum JavaScript-Objekt "ret" und den erforderlichen Parametern.

Der JavaScript-Teil, der durch den JMS-Treiber ausgeführt wird, greift auf ein JavaScript-Objekt namens *ret* zu. Dies ist ein Objekt des Typs "com.ibm.di.systemqueue.driver.JMSScriptDriver.Ret". Es handelt sich um eine Instanz der untergeordneten Klasse *Ret* der JMS-Scripttreiberklasse. Dieses Objekt *ret* wird verwendet, um die providerspezifischen Objekte, die der JavaScript-Code aus dem JMS-System erhält, an den JMS-Scripttreiber und letztendlich an die Systemwarteschlange zurückzugeben. Das Objekt *ret* kann außerdem verwendet werden, um Fehlerinformationen an den JMS-Treiber und die Systemwarteschlange zurückzugeben.

Dieses Objekt "ret" besitzt die folgenden Elemente, die über JavaScript definiert werden können:

- `queueConnectionFactory`: Ein Objekt des Typs "javax.jms.QueueConnectionFactory". Dies ist die Position, an der das vom jeweiligen JMS-System erhaltene Objekt "javax.jms.QueueConnectionFactory" gespeichert werden soll.
- `topicConnectionFactory`: Ein Objekt des Typs "javax.jms.TopicConnectionFactory". Dies ist die Position, an der das vom jeweiligen JMS-System erhaltene Objekt "javax.jms.TopicConnectionFactory" gespeichert werden soll.
- `errorCode`: Ein Objekt des Typs "java.lang.Object". Dies ist die Position, an der ein Objekt mit Fehlerinformationen gespeichert werden soll. Ein Beispiel für ein solches Objekt wäre ein Objekt "java.lang.Exception".
- `errordescr`: Ein Objekt des Typs "java.lang.String". Dies ist die Position, an der Fehlerbeschreibungstexte gespeichert werden sollen.

JavaScript-Beispiel für Fiorano MQ

Mithilfe des bereitgestellten Beispiels können Sie Informationen zu Fiorano MQ erhalten.

Für die Verwendung des Drittanbietersystems "Fiorano MQ" werden im Ordner *tdi-installationsverzeichnis/examples* eine Beispielkonfiguration und JavaScript-Code bereitgestellt, die nachstehend dargestellt sind:

```
var ctx = new Packages.java.util.Hashtable();
ctx.put("jms.username", "anonymous");
ctx.put("jms.password", "anonymous");
ctx.put("jms.
oker", "http://192.168.113.220:1856");
ctx.put("jms.qManager", "fiorano.jms.runtime.naming.FioranoInitialContextFactory");

var ic = new javax.naming.InitialContext(ctx);

var queueFactory = ic.lookup("primaryQCF");
var topicFactory = ic.lookup("primaryTCF");

ret.queueConnectionFactory = queueFactory;
main.logmsg("driverFiorano.js : QueueConnectionFactory : " + queueFactory);

ret.topicConnectionFactory = topicFactory;
main.logmsg("driverFiorano.js : TopicConnectionFactory : " + topicFactory);
```

Anmerkung: Dieser JavaScript-Teil veranschaulicht, wie die Parameter im JavaScript-Code fest codiert werden können. Alternativ kann das JavaScript-Objekt *env* verwendet werden, um etwaige vom Benutzer bereitgestellte Parameter aus der Datei `global.properties` oder `solution.properties` abzurufen. Die Verwendung des Objekts *env* für den Parameterabruf erleichtert Änderungen an der Konfiguration, da nur Eigenschaften in der Datei `global.properties` oder `solution.properties` geändert werden müssen und keine Bearbeitung des Ja-

vaScript-Codes erforderlich ist. Dies bedeutet, dass auch Benutzer ohne JavaScript-Kenntnisse in der Lage wären, die Konfiguration zu ändern.

Beispiel für Konfiguration der Systemwarteschlange

Nachstehend finden Sie ein Beispiel für die Konfiguration der Systemwarteschlange.

```
##-----  
## System Queue settings  
##-----  
## If set to "true" the System Queue is  
## initialized on startup and can be used;  
## otherwise the System Queue is not  
## initialized and cannot be used.  
systemqueue.on=true  
  
## Specifies the fully qualified name of  
## the class that will be used as a JMS Driver.  
# systemqueue.jmsdriver.name=com.ibm.di.systemqueue.driver.JMSScriptDriver  
# systemqueue.jmsdriver.name=com.ibm.di.systemqueue.driver.IBMMQ  
systemqueue.jmsdriver.name=com.ibm.di.systemqueue.driver.ActiveMQ  
  
### MQe JMS driver initialization properties  
## Specifies the location of the MQe initialization file.  
## This file is used to initialize MQe on TDI server startup.  
# systemqueue.jmsdriver.param.mqe.file.ini=MQePWStore/pwstore_server.ini  
  
### MQ JMS driver initialization properties  
systemqueue.jmsdriver.param.jms  
oker=192.168.113.54:1414  
systemqueue.jmsdriver.param.jms.serverChannel=S_s04win  
systemqueue.jmsdriver.param.jms.qManager=QM_s04win  
systemqueue.jmsdriver.param.jms.sslCipher=SSL_RSA_WITH_RC4128_MD5  
systemqueue.jmsdriver.param.jms.sslUseFlag=true  
  
### JMS Javascript driver initialization properties  
## Specifies the location of the script file  
# systemqueue.jmsdriver.param.js.jsfile=driver.js  
  
### ActiveMQ driver initialization properties  
## Specifies the location of the ActiveMQ initialization file.  
## This file is used to initialize ActiveMQ on TDI server startup.  
systemqueue.jmsdriver.param.jms  
oker=vm://localhost?brokerConfig=xbean:etc/activemq.xml  
  
## This is the place to put any JMS provider specific properties needed by a JMS Driver,  
## which connects to a 3rd party JMS system.  
## All JMS Driver properties should begin with the 'systemqueue.jmsdriver.param.' prefix.  
## All properties having this prefix are passes to the JMS Driver on initialization after  
## removing the 'systemqueue.jmsdriver.param.' prefix from the property name.  
# systemqueue.jmsdriver.param.user.param1=value1  
# systemqueue.jmsdriver.param.user.param2=value2  
# ...  
  
## Credentials used for authenticating to the target JMS system  
# {protect}-systemqueue.auth.username=<username>  
# {protect}-systemqueue.auth.password=<password>
```

Sicherheit und Authentifizierung

Nachstehend erhalten Sie Informationen zur Sicherheit und zur Authentifizierung sowie zu deren jeweiligen Methoden.

Verschlüsselung

Von den JMS-Standardtreibern wird SSL nur durch den Treiber für MQ unterstützt. Der IBM WebSphere MQ Everyplace-JMS-Treiber kann nur zusammen mit einem lokalen Warteschlangenmanager verwendet werden. Diese Bedingung wird durch die IBM WebSphere MQ Everyplace-Architektur vorgegeben. Der JMS-Scripttreiber ist ein generischer Treiber, der alle durch das entsprechende, vom Benutzer bereitgestellte JavaScript unterstützten Elemente unterstützt.

Authentifizierung

Einige JMS-Systeme wie beispielsweise IBM WebSphere MQ können die Authentifizierung über Benutzername und Kennwort verwenden oder sogar erfordern. Die Systemwarteschlange bietet zwei Standardeigenschaften in der Datei `global.properties` oder `solution.properties`, mit denen ein Benutzername und ein Kennwort für die Systemwarteschlange konfiguriert und bereitgestellt werden können. Es handelt sich um die Eigenschaften `systemqueue.auth.username` und `systemqueue.auth.password`. Diese beiden Eigenschaften werden durch die Standardverschlüsselung des IBM Security Directory Integrator-Servers für Eigenschaften geschützt, die mit `{protect}`-gekennzeichnet sind. Hierdurch werden die Werte der Eigenschaften verschlüsselt, nachdem diese Eigenschaften festgelegt wurden und der IBM Security Directory Integrator-Server gestartet wurde. Weitere Informationen zu diesen beiden Eigenschaften enthält der Abschnitt „Konfiguration der Systemwarteschlange“ auf Seite 177.

Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace

Nachstehend erhalten Sie Informationen zum Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace.

Für die Konfiguration und Verwendung von IBM WebSphere MQ Everyplace als Standardsystemwarteschlange müssen Sie den MQ-Warteschlangenmanager im neuen Lösungsverzeichnis mit dem Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace einrichten. Der eingerichtete IBM WebSphere MQ Everyplace-Warteschlangenmanager besitzt zwei vordefinierte Warteschlangen:

- **_default**: Diese Warteschlange dient allgemeinen Zwecken.
- **passwords**: Dient für Kennwörter zu allgemeinen Zwecken. Diese Warteschlange wird durch die JMS-Komponenten für den Kennwortspeicher verwendet, um Kennwortänderungen zu speichern. Die Systemwarteschlange ist somit sofort einsatzbereit.

Das Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace in IBM Security Directory Integrator (ein Befehlszeilendienstprogramm) erstellt bei der erstmaligen Einrichtung des IBM WebSphere MQ Everyplace-Warteschlangenmanagers eine IBM WebSphere MQ Everyplace-Standardwarteschlange. Diese IBM WebSphere MQ Everyplace-Standardwarteschlange heißt "**_default**". Sie wird ausschließlich zum Zweck eines größeren Verwendungskomforts erstellt, damit ein Benutzer das Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace verwenden kann, um IBM WebSphere MQ Everyplace (mit dem entsprechenden Befehl des Konfigurationsdienstprogramms für IBM WebSphere MQ Everyplace) einzurichten und anschließend die Systemwarteschlange und den Systemwarteschlangenconnector sofort verwenden zu können.

Mit dem Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace von IBM Security Directory Integrator können Sie außerdem IBM WebSphere MQ Everyplace-Benutzerwarteschlangen erstellen und löschen, die durch die Systemwarteschlange und den Systemwarteschlangenconnector verwendet werden sollen.

IBM WebSphere MQ Everyplace-Warteschlange mit dem Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace erstellen

Nach Eingabe des folgenden Befehls wird eine IBM WebSphere MQ Everyplace-Warteschlange namens `warteschlangenname` unter Verwendung der Konfigurationsdatei `mqeconfig.props` erstellt:

```
mqeconfig mqeconfig.props create queue warteschlangenname
```


IBM WebSphere MQ Everyplace-Warteschlange mit dem Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace löschen

Nach Eingabe des folgenden Befehls wird eine IBM WebSphere MQ Everyplace-Warteschlange namens `warteschlangenname` unter Verwendung der Konfigurationsdatei `mqeconfig.props` gelöscht:

```
mqeconfig mqeconfig.props delete queue warteschlangenname
```

Falls für Ihre Lösung eine spezielle Konfiguration erforderlich ist, können Sie mit dem IBM WebSphere MQ Everyplace-Explorer eine Feinabstimmung Ihrer IBM WebSphere MQ Everyplace-Konfiguration vornehmen. Der IBM WebSphere MQ Everyplace-Explorer ist im Produktpaket von IBM Security Directory Integrator nicht enthalten, kann aber als Teil des Pakets "IBM WebSphere MQ Everyplace Server Support ES06" unter der folgenden Adresse heruntergeladen werden: http://www-1.ibm.com/support/docview.wss?rs=0&dc=D400&q1=MQe&q2=MQ+Everyplace&uid=swg24007943&loc=en_US&cs=utf-8&cc=us&lang=en.

Sicherheit für Warteschlange durch Authentifizierung von IBM WebSphere MQ Everyplace-Nachrichten

Sie können durch die Authentifizierung von IBM WebSphere MQ Everyplace-Nachrichten Sicherheit für die Warteschlange gewährleisten.

In IBM Security Directory Integrator kann der Zugriff auf IBM WebSphere MQ Everyplace durch Authentifizierung geschützt werden. Hierzu werden mit dem IBM WebSphere MQ Everyplace-Server für Minizertifikate Zertifikate ausgestellt, die für die Authentifizierung verwendet werden. Zu diesem Zweck müssen mehrere zusätzliche Eigenschaften, die in IBM Security Directory Integrator verfügbar sind, zur Eigenschaftendatei `mqeconfig.props` hinzugefügt werden. Diese Datei enthält die Konfigurationseigenschaften des Konfigurationsdienstprogramms für IBM WebSphere MQ Everyplace.

Die vom IBM WebSphere MQ Everyplace-Server für Minizertifikate ausgestellten Zertifikate haben einen konfigurierbaren Gültigkeitszeitraum. Standardmäßig beträgt der Gültigkeitszeitraum 12 Monate. In der IBM WebSphere MQ Everyplace-Dokumentation ist angegeben, dass ausgestellte Zertifikate vor Ablauf des Zeitraums verlängert werden sollten. Zu diesem Zweck enthält das Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace eine Option für die Verlängerung von Zertifikaten. Nach Eingabe des folgenden Befehls wird das Zertifikat verlängert:

```
mqeconfig mqeconfig.props renewcert {client | server}
```

1. Wenn die letzte Befehlsoption "client" lautet, müssen die folgenden Werte in der Datei `mqeconfig.props` festgelegt sein:

- **clientRootFolder:** Das Verzeichnis, in dem sich die IBM WebSphere MQ Everyplace-Konfigurationsinstanz befindet.
- **certServerReqPin:** Dieser Wert wird als persönliche Identifikationsnummer für die einmalige Authentifizierung der angegebenen authentifizierbaren Entität verwendet, wenn die Zertifikatverlängerung vom IBM WebSphere MQ Everyplace-Server für Minizertifikate angefordert wird.
- **certServerIPAndPort:** Dieser Wert wird als Zieladresse für Anforderungen an den IBM WebSphere MQ Everyplace-Server für Minizertifikate verwendet. Der Wert hat das Format "FastNetwork:<host>:<port>". Hierbei muss für "host" der Computername, die TCP/IP-Adresse oder der Hostname angegeben werden, unter dem/der der IBM WebSphere MQ Everyplace-Server für Minizertifikate ausgeführt wird.

- **certRenewalEntityName:** Der Name der über IBM WebSphere MQ Everyplace authentifizierbaren Entität, für die eine Zertifikatverlängerung angefordert wird. Typische Namen von Entitäten sind nachfolgend aufgeführt. Unter der Voraussetzung, dass die Entität tatsächlich in der Registry des Warteschlangenmanagers vorhanden ist, auf die durch den Wert von "clientRootFolder" verwiesen wird, kann jedoch jeder im IBM WebSphere MQ Everyplace-Minizertifikat konfigurierte Entitätsname verwendet werden:
 - PWStoreClient: Clientseitiger IBM WebSphere MQ Everyplace-Warteschlangenmanager
 - PWStoreServer+passwords: Clientseitiger Proxy für ferne Warteschlange
2. Wenn die letzte Befehlsoption "server" lautet, müssen die folgenden Werte in der Datei mqeconfig.props festgelegt sein:
- **serverRootFolder:** Das Verzeichnis, in dem sich die IBM WebSphere MQ Everyplace-Konfigurationsinstanz befindet.
 - **certServerReqPin:** Dieser Wert wird als persönliche Identifikationsnummer für die einmalige Authentifizierung der angegebenen authentifizierbaren Entität verwendet, wenn die Zertifikatverlängerung vom IBM WebSphere MQ Everyplace-Server für Minizertifikate angefordert wird.
 - **certServerIPAndPort:** Dieser Wert wird als Zieladresse für Anforderungen an den IBM WebSphere MQ Everyplace-Server für Minizertifikate verwendet. Der Wert hat das Format "FastNetwork:<host>:<port>". Hierbei muss für "host" der Computername, die TCP/IP-Adresse oder der Hostname angegeben werden, unter dem/der der IBM WebSphere MQ Everyplace-Server für Minizertifikate ausgeführt wird.
 - **certRenewalEntityName:** Der Name der über IBM WebSphere MQ Everyplace authentifizierbaren Entität, für die eine Zertifikatverlängerung angefordert wird. Typische Namen von Entitäten sind nachfolgend aufgeführt. Unter der Voraussetzung, dass die Entität tatsächlich in der Registry des Warteschlangenmanagers vorhanden ist, auf die durch den Wert von "serverRootFolder" verwiesen wird, kann jedoch jeder im IBM WebSphere MQ Everyplace-Minizertifikat konfigurierte Entitätsname verwendet werden:
 - PWStoreServer: Serverseitiger IBM WebSphere MQ Everyplace-Warteschlangenmanager
 - PWStoreServer+passwords: Eigentliche Warteschlange auf der Serverseite

Unterstützung für DNS-Namen in der Konfiguration der IBM WebSphere MQ Everyplace-Warteschlange

Zur Unterstützung dieser Funktionsweise ist keine zusätzliche Codierung erforderlich.

Es sollte beachtet werden, dass die DNS-Unterstützung eigentlich eine IBM WebSphere MQ Everyplace-Funktion ist, da die Implementierungen von IBM Security Directory Integrator-Komponenten einfach die Konfigurationseigenschaften aus der Datei mqeconfig.props über die IBM WebSphere MQ Everyplace-APIs übergeben. Die folgenden Eigenschaften in der Datei mqeconfig.props können Werte für DNS-Namen oder IP-Adressen akzeptieren:

- serverIP
- certServerIPAndPort

Konfiguration von Hochverfügbarkeit für den IBM WebSphere MQ Everyplace-Transport von Kennwortänderungen

Nachstehend erhalten Sie Informationen zum Konfigurieren von Hochverfügbarkeit für den IBM WebSphere MQ Everyplace-Transport von Kennwortänderungen.

Zur Unterstützung von Hochverfügbarkeitsimplementierungen haben Sie die Möglichkeit, mehrere Instanzen der IBM WebSphere MQ Everyplace-Komponenten von IBM Security Directory Integrator zu implementieren und zu konfigurieren. Bei einigen Implementierungen kann es erforderlich sein, mehrere IBM WebSphere MQ Everyplace-Kennwortspeicherkomponenten für IBM Security Directory Integrator zu konfigurieren. Wenn beispielsweise für mehrere Windows-Domänencontroller Kennwortänderungs-Plug-ins konfiguriert wurden, ist es wahrscheinlich, dass es separate Instanzen der clientseitigen IBM WebSphere MQ Everyplace-Warteschlangenmanager namens "PWStoreClient" gibt. Außerdem gibt es für jeden clientseitigen Warteschlangenmanager eine ferne Warteschlangenproxyverbindung zur Warteschlange des Warteschlangenmanagers auf der IBM WebSphere MQ Everyplace-Serverseite, die durch den IBM WebSphere MQ Everyplace-Kennwortconnector für IBM Security Directory Integrator genutzt wird. Der Name des Proxys für die ferne Warteschlange lautet "PWStoreServer+passwords". Wenn Sie diesen Typ eines Implementierungsszenarios verwenden, werden die Authentifizierungszertifikate, die diesen beiden IBM WebSphere MQ Everyplace-Entitäten (also "PWStoreClient" und "PWStoreServer+passwords") zugeordnet sind, mehrmals angefordert und ausgestellt. Dies erfolgt bei jeder Ausführung des Dienstprogramms "mqeconfig". Vor der Ausführung der zweiten und aller nachfolgenden Instanzen des Dienstprogramms "mqeconfig" muss die Zertifikatausstellung für jede der oben genannten IBM WebSphere MQ Everyplace-Entitäten erneut aktiviert werden.

Bei manchen Implementierungen kann es sinnvoller sein, den IBM WebSphere MQ Everyplace-Kennwortconnector für IBM Security Directory Integrator so zu konfigurieren, dass er eine bestimmte Voraussetzung für die Hochverfügbarkeit unterstützt. Möglicherweise erwarten Sie, dass eine Implementierung, die diesen Voraussetzungstyp unterstützt, mehrere Instanzen des IBM WebSphere MQ Everyplace-Kennwortconnectors für IBM Security Directory Integrator mit jeweils einer eigenen zugehörigen Konfiguration des IBM WebSphere MQ Everyplace-Warteschlangenmanagers einsetzt. In diesem Fall würden Sie mehrere identische serverseitige IBM WebSphere MQ Everyplace-Konfigurationen implementieren. Dies erlaubt es einer Netzlastausgleichsfunktion, Anforderungen vom Client des IBM WebSphere MQ Everyplace-Kennwortspeichers für an eine verfügbare Serverinstanz weiterzuleiten. Jeder IBM WebSphere MQ Everyplace-Warteschlangenmanager auf der Serverseite wird mit dem Dienstprogramm "mqeconfig" konfiguriert. Wenn dieses Dienstprogramm ausgeführt wird, fordert es automatisch Authentifizierungszertifikate vom IBM WebSphere MQ Everyplace-Server für Minizertifikate für die Entitäten mit den Namen "PWStoreServer" und "PWStoreServer+passwords" an. Diese Entitäten stellen die Namen für den Warteschlangenmanager bzw. die Warteschlange dar. Vor der Ausführung der zweiten und aller nachfolgenden Instanzen des Dienstprogramms "mqeconfig" muss die Zertifikatausstellung für beide oben genannten IBM WebSphere MQ Everyplace-Entitäten erneut aktiviert werden.

Funktionen im Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace für die ferne Konfiguration

Mit den hier aufgeführten Anweisungen können Sie Funktionen im Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace für die ferne Konfiguration bereitstellen.

Ferne IBM WebSphere MQ Everyplace-Warteschlange mit dem Konfigurationsdienstprogramm erstellen

Nach Eingabe des folgenden Befehls wird eine ferne IBM WebSphere MQ Everyplace-Warteschlange namens "warteschlangenname" unter Verwendung der Konfigurationsdatei mqeconfig.props erstellt:

```
mqeconfig mqeconfig.props create remotequeue warteschlangenname  
targetQMname [ip_oder_hostname_für_mq übertragungSPORT]
```

In der obigen Befehlszeile sind die Parameter ip_oder_hostname_für_mq und übertragungSPORT optional. Falls sie fehlen, wird nur die Definition einer fernen Warteschlange erstellt. Wenn Sie diese beiden Parameter angeben, wird vor der Erstellung der Definition einer fernen Warteschlange auch eine Verbindungsdefinition erstellt.

Anmerkung: Eine ferne Warteschlange kann ohne eine Verbindungsdefinition nicht verwendet werden. Zudem können mehrere ferne Warteschlangen für die gemeinsame Nutzung einer einzigen Verbindung definiert sein. Der Parameter "targetQMname" gibt den Namen des fernen IBM WebSphere MQ Everyplace-Warteschlangenmanagers an.

Ferne IBM WebSphere MQ Everyplace-Warteschlange mit dem Konfigurationsdienstprogramm für IBM WebSphere MQ Everyplace löschen

Nach Eingabe des folgenden Befehls wird eine ferne IBM WebSphere MQ Everyplace-Warteschlange namens "warteschlangenname" unter Verwendung der Konfigurationsdatei mqeconfig.props gelöscht:

```
mqeconfig mqeconfig.props delete remotequeue warteschlangenname targetQMname
```

In der obigen Befehlszeile gibt der Parameter "targetQMname" den Namen des fernen IBM WebSphere MQ Everyplace-Warteschlangenmanagers an.

Kapitel 9. Verschlüsselung und FIPS-Modus

Nachstehend erhalten Sie Informationen und Links zur Verschlüsselung.

IBM Security Directory Integrator kann die folgenden Objekte verschlüsseln, um die Vertraulichkeit von Daten zu ermöglichen:

- Konfigurationsdateien
- Eigenschaftswerte in Eigenschaftendateien
- Dateien der Serverbenutzerregistry
- JavaScript-Dateien

Als *Verschlüsselung* wird der Prozess bezeichnet, mit dem ein vom Benutzer lesbarer Text (so genannter *unverschlüsselter Text*) ausgewählt und sein Inhalt verborgen wird. Dieser Prozess dient dazu, Daten sicherer zu machen, die im unverschlüsselten Textformat vorliegen. Unverschlüsselter Text ist in Kleinbuchstaben geschrieben. Verschlüsselter Text wird auch als *chiffrierter Text* bezeichnet. Verschlüsselter oder chiffrierter Text ist in Großbuchstaben geschrieben.

Anmerkung: Wenn dem Namen einer Eigenschaft in einer Konfigurationsdatei das Präfix `{protect}`- vorangestellt ist, bedeutet dies, dass der Eigenschaftswert verschlüsselt ist oder werden soll. Das Präfix `{protect}`- ist optional. Bereits verschlüsselte Werte beginnen mit dem Präfix `{encr}`.

Weitere Informationen können Sie den Abschnitten „Mit verschlüsselten IBM Security Directory Integrator-Konfigurationsdateien arbeiten“ auf Seite 149 und „Verschlüsselung von Eigenschaften in externen Eigenschaftendateien“ auf Seite 153 entnehmen.

Beispiel:

```
[{protect}-]schlüsselwort <doppelpunkt | gleichheitszeichen> [{encr}][{java}]wert
```

Der Wert für `{java}` muss mit b64 codiert sein. Beispiel:

```
{protect}-api.truststore.pass={encr}J8AKimpEutu3Bb10Vg55F/5d5v02kXWcNUWncQ3vINUc6K0719z9dEk3H430t2iTT1dZTI6FSSV  
in9KsCyBLmgv+n84w7He1K13ro2dFmZbTYKMXuxGoqN9nL2V0vZoptNqzoWvs6IN/p3VkiIBt1ao/9mEPEKuIwRnKtKQ89Bqg=
```

IBM Security Directory Integrator zur Ausführung im FIPS-Modus konfigurieren

Sie können IBM Security Directory Integrator zur Ausführung im FIPS-Modus konfigurieren.

Federal Information Processing Standard (FIPS) Publication 140-2 (kurz "FIPS PUB 140-2") ist ein von den Regierungsbehörden in den USA aufgestellter IT-Sicherheitsstandard, der für die Anerkennung von Verschlüsselungsmodulen verwendet wird.

Wenn der IBM Security Directory Integrator-Server für die Ausführung im FIPS-Modus konfiguriert ist, verwendet der Server die gemäß FIPS 140-2 zertifizierten Verschlüsselungsmodule. IBM Security Directory Integrator generiert keine Verschlüsselungsschlüssel. Die Schlüssel werden mit externen Dienstprogrammen wie *keytool* und *Ikeyman* erstellt. Informationen zum Einsatz der Verschlüsselung in IBM Security Directory Integrator finden Sie in Kapitel 6, „Sicherheit“, auf Seite 103. Für die Erstellung, die Bearbeitung, den Export und die Gesamtverwaltung von Schlüsselspeichern und Truststores kann das GUI-Dienstprogramm *Ikeyman* oder das Be-

fehlszeilendienstprogramm `keytool` verwendet werden. Die ausführbare Datei "keytool.exe" befindet sich je nach Plattform im Verzeichnis *stammverzeichnis/jvm/jre/bin* bzw. *stammverzeichnis/jvm/bin*.

Unterstützung des symmetrischen Chiffrierwerts

Mit der Unterstützung des symmetrischen Chiffrierwerts können Sie eine Konformität mit FIPS 140-2 erreichen.

Eine Nachricht wird verschlüsselt, um sie in ein bedeutungsloses Textformat zu ändern, das als "verschlüsselter Text" bezeichnet wird und für einen Dritten, der die Nachricht abfängt, ohne Aussage ist. Es gibt viele verschiedene Verschlüsselungsalgorithmen, die auch "Chiffrierwerte" genannt werden. Einer der am weitesten verbreiteten Chiffrierwerte ist der *symmetrische* Chiffrierwert. Der symmetrische Chiffrierwert besitzt einen Schlüssel, den sowohl der Absender als auch der Empfänger speichern kann. Der Absender verwendet diesen Schlüssel, um die Nachricht zu verschlüsseln. Der Empfänger verwendet denselben Schlüssel, um die Nachricht zu entschlüsseln.

Es wird eine optionale Konfiguration für die Verwendung eines symmetrischen Chiffrierwerts (nämlich Advanced Encryption Standard - AES) bereitgestellt. Der mit AES verschlüsselte symmetrische Chiffrierwert ermöglicht es Kunden, die FIPS-konforme Lösungen benötigen, bei der Verschlüsselung einen unterstützten Chiffrierwert zu verwenden.

Die folgende Eigenschaft definiert den Chiffrierwert:

```
com.ibm.di.securityTransformation=DES/ECB/NoPadding
```

Diese Eigenschaft definiert einen Chiffrierwert für die kennwortbasierte Verschlüsselung oder Entschlüsselung von IBM Security Directory Integrator-Konfigurationen.

FIPS-Verschlüsselung

Durch die Verwendung von FIPS können Sie IBM Security Directory Integrator und den IBM Security Directory Integrator-Server auf geschützte Weise ausführen. Außerdem können Sie zusätzliche Eigenschaften konfigurieren, wenn IBM Security Directory Integrator in einem speziellen Modus (z. B. dem FIPS-Modus) betrieben werden soll.

Connectors, Funktionskomponenten, Parser:

Nachstehend erhalten Sie Informationen zu Connectors, Funktionskomponenten und Parsern.

FIPS 140-2 kommt lediglich im Zusammenhang mit Verschlüsselungsfunktionalität wie SSL, digitale Unterzeichnung, Verschlüsselung, verschlüsseltes Hashverfahren und Generierung von Zufallszahlen zum Einsatz.

Konfiguration mit SSL

FIPS 140-2 erfordert, dass TLS als Protokoll für die SSL-Kommunikation verwendet wird. SSLv3 und dessen Vorläufer sind nicht zulässig. Wenn der FIPS-Modus aktiviert ist, können IBM Security Directory Integrator-Komponenten, die SSL verwenden, nicht mit externen Systemen kommunizieren, die TLS nicht unterstützen.

JDBC und Systemspeicher

Der DB2-JDBC-Treiber des Typs 4 (`com.ibm.db2.jcc.DB2Driver`), der mit IBM Security Directory Integrator ausgeliefert wird, unterstützt SSL nur auf FIPS-konformer Weise.

Die Apache Derby-Treiber (Netztreiber und integrierter Treiber) unterstützen SSL in Version 10.5.3 nicht (dies ist die Version, die im Produktpaket von IBM Security Directory Integrator vor Version 7.2 enthalten ist).

Die Datenbanksteuerkomponente von Apache Derby Version 10.8 kann jedoch eine Datenbankverschlüsselung ausführen. IBM Security Directory Integrator verwendet Derby standardmäßig als Systemspeicher. Falls Sie die Apache Derby Version 10.8-Funktionalität für die Datenbankverschlüsselung im FIPS-Modus verwenden, müssen Sie darauf achten, den von IBM zertifizierten Verschlüsselungsprovider `IBMJCEFIPS` als den für die Verschlüsselung verwendeten Provider anzugeben, und außerdem einen von FIPS genehmigten Chiffrierwert für die Verschlüsselung auswählen. Das folgende Beispiel zeigt, wie der Systemspeicher für die Verwendung von Derby mit einer FIPS-konformen Datenbankverschlüsselung konfiguriert wird:

```
com.ibm.di.store.database=jdbc:derby://localhost:1527/C:\TDI\TDISysStoreEnc;create=true;
  dataEncryption=true;encryptionKey=c566bab9ee8b62a5ddb4d9229224c678;encryptionAlgorithm=AES/CBC/NoPadding;
  encryptionProvider=com.ibm.crypto.fips.provider.IBMJCEFIPS
com.ibm.di.store.jdbc.driver=org.apache.derby.jdbc.ClientDriver
com.ibm.di.store.jdbc.urlprefix=jdbc:derby:
com.ibm.di.store.jdbc.user=APP
```

JMS und Systemwarteschlange

IBM WebSphere MQ Everyplace-Minizertifikate beinhalten eine Verschlüsselung, die nicht FIPS-konform ist. Diese Sicherheitsfunktion von IBM WebSphere MQ Everyplace sollte daher im FIPS-Modus nicht verwendet werden.

Der JMS-Provider von IBM WebSphere MQ 5.3 kann SSL nicht in einem FIPS-konformen Modus ausführen. Im FIPS-Modus sollte SSL nicht zusammen mit diesem Provider verwendet werden.

Zur Verwendung der FIPS-konformen SSL-Kommunikation zwischen IBM Security Directory Integrator und IBM WebSphere MQ müssen Sie Folgendes ausführen:

1. Stellen Sie sicher, dass die IBM WebSphere MQ-Installation auf dem Stand von Version 7.1 oder höher ist.
2. Stellen Sie sicher, dass der entsprechende Warteschlangenmanager auf der MQ-Seite eine FIPS-konforme SSL-Kommunikation erfordert.
3. Stellen Sie sicher, dass der entsprechende SSL-Kanal des Warteschlangenmanagers eine FIPS-konforme SSL-Chiffrierwertspezifikation verwendet.
4. Aktivieren Sie den FIPS-Modus für IBM Security Directory Integrator. Wenn der FIPS-Modus für IBM Security Directory Integrator aktiviert ist, wird der FIPS-Modus automatisch für alle JMS-SSL-Verbindungen zu IBM WebSphere MQ aktiviert.
5. Kopieren Sie die JAR-Dateien des JMS-Clients aus der IBM WebSphere MQ-Installation nach IBM Security Directory Integrator. Eine Liste der erforderlichen Clientbibliotheken für MQ 7.1 sowie Angaben über deren Implementierung in IBM WebSphere MQ enthält die Dokumentation zum JMS-Connector im Handbuch *Referenzinformationen*.
6. Konfigurieren Sie auf der IBM Security Directory Integrator-Seite eine FIPS-konforme SSL-Cipher Suite, die mit der für den SSL-Kanal des MQ-Warteschlangenmanagers konfigurierten SSL-Chiffrierwertspezifikation kompatibel ist. Hier-

zu können Sie den Parameter `jms.sslCipher` des JMS-Connectors und die Systemeigenschaft `systemqueue.jmsdriver.param.jms.sslCipher` des MQ-Treibers für die Systemwarteschlange verwenden. Weitere Informationen finden Sie im Abschnitt zur Zuordnung von *SSL-Chiffrierwertspezifikationen und -CipherSuites* und deren FIPS-Kompatibilität in der WebSphere MQ-Dokumentation.

IBM Security Directory Integrator-Server und FIPS:

Beachten Sie die hier aufgeführten Auswirkungen, wenn Sie mit dem IBM Security Directory Integrator-Server und mit FIPS arbeiten.

Bei einer Ausführung in diesem Modus wird die Verwendung von Verschlüsselungsmodulen gemäß FIPS 140-2 durch den IBM Security Directory Integrator-Server erzwungen.

Anmerkung: Falls der Server unter Aktivierung von FIPS und SSL ausgeführt wird, verwenden Sie keine Clients mit SSL für die Kommunikation über sichere Sockets. In diesem Fall verwendet der Server TLS und es ist kein erfolgreicher Verbindungsaufbau möglich. Anstelle der Verwendung von SSL müssen Sie sicherstellen, dass Sie wie der Server TLS für die Kommunikation über sichere Sockets verwenden.

Die Ausführung des IBM Security Directory Integrator-Servers im FIPS-Modus hat die folgenden Auswirkungen:

- Nur FIPS-konforme Verschlüsselungsalgorithmen sind für die Verschlüsselung und Entschlüsselung von Konfigurationen, Eigenschaften usw. zulässig.
- Externe Tools, die eine Ver-/Entschlüsselung verwenden (Ikeyman, createstash, cryptoutils, keytool usw.), sollten in einer FIPS-konformen Weise eingesetzt werden.
- Komponenten können nicht mit externen Systemen kommunizieren, die für die Socketkommunikation nicht TLS verwenden.
- Einige Komponenten sollten nicht verwendet werden, wenn der Server im FIPS-Modus ausgeführt wird, weil sie die FIPS-Konformität aufheben. Eine Liste mit Details über die Komponentenkonformität enthält Tabelle 21 auf Seite 196.

FIPS-Modus aktivieren:

Bei der Verwendung von FIPS werden viele Konfigurationsoptionen von IBM Security Directory Integrator geändert. Daher müssen Sie verschiedene Regeln bedenken, damit die FIPS-Konformität erhalten bleibt.

Einige Regeln sind in diesem Dokument erwähnt. Weitere Regeln finden Sie unter den Adressen <https://w3.webahead.ibm.com/w3ki/download/attachments/370821/FIPS+140+Guidelines.pdf?version=1> und <http://www.ibm.com/developerworks/java/jdk/security/60/FIPShowto.html>.

FIPS-Modus in IBM Security Directory Integrator aktivieren

1. Setzen Sie die Eigenschaft `com.ibm.di.server.fipsmode.on` in der Datei `global.properties` oder `solution.properties` auf **true**.
2. Stellen Sie sicher, dass als Wert für die Eigenschaft `com.ibm.di.securityTransformation` ein FIPS-konformer Algorithmus verwendet wird, beispielsweise `AES/ECB/NoPadding`. Dieser Algorithmus wird verwendet, wenn Sie versuchen, eine verschlüsselte Konfiguration zu öffnen.

3. Die Hardwareverschlüsselung kann im FIPS-Modus nicht zusammen mit SSL verwendet werden. Das zugrunde liegende SSL-Modul IBMJSSE2 unterstützt die Hardwareverschlüsselung im FIPS-Modus nicht. Entsprechende Angaben können Sie unter der folgenden Adresse nachlesen: <http://www-128.ibm.com/developerworks/java/jdk/security/60/secguides/jsse2Docs/JSSE2RefGuide.html#runfips>. Die hardwarebasierten SSL-Schlüssel für die Server-APIs können im FIPS-Modus nicht verwendet werden. Die Eigenschaft `com.ibm.di.server.pkcs11` darf in der Datei `global.properties` und `solution.properties` entweder nicht angegeben oder muss auf "false" gesetzt sein.
4. Stellen Sie sicher, dass die Serververschlüsselung eine mit FIPS 140-2 konforme Umsetzung verwendet.

Der Server verwendet standardmäßig die Verschlüsselung mit öffentlichem Schlüssel und dem RSA-Algorithmus. Die Option für die RSA-Verschlüsselung ist jedoch mit FIPS 140-2 nicht konform. Aus diesem Grund müssen Sie manuell eine andere Verschlüsselungsumsetzung konfigurieren, die bei FIPS zulässig ist. Die folgenden Schritte sind ein Beispiel dafür, wie Sie IBM Security Directory Integrator für die Verwendung des AES-Chiffrierwerts zur Verschlüsselung konfigurieren:

- Generieren Sie einen geheimen AES-Schlüssel und legen Sie ihn in einem Schlüssel Speicher ab. Hierzu können Sie das Dienstprogramm `keytool` verwenden, das sich im Ordner `bin` einer IBM Security Directory Integrator-Installation befindet. Beispiel:

```
keytool -genseckey -alias server -keyalg AES -keysize 128 -keystore server.jck -storepass mypass -storetype jceks
-keypass mykeypass -providerClass com.ibm.crypto.fips.provider.IBMJCEFIPS
```

Dieser Befehl erstellt eine neue Schlüssel Speicherdatei `server.jck` des Typs JCEKS (JKS-Schlüssel Speicher können keine geheimen Schlüssel verwenden) mit einem AES-Schlüssel in der Größe 128 unter dem Aliasnamen `server`. Das Kennwort für den erstellten Schlüssel Speicher lautet `mypass`. Achten Sie insbesondere auf den Parameter `keygenproviderclass` – es muss unbedingt der nach FIPS zertifizierte Provider angegeben sein, falls Sie eine Konformität mit FIPS 140-2 erreichen wollen. Bitte beachten Sie, dass es sich hierbei lediglich um ein Beispiel handelt. Sie können die Werte für Dateinamen, Kennwörter und Aliasnamen frei wählen.

- Ändern Sie die IBM Security Directory Integrator-Einstellungen so, dass die Verschlüsselung mit geheimem Schlüssel zusammen mit dem neu generierten Schlüssel verwendet wird. Legen Sie beispielsweise in der Datei `global.properties` oder `solution.properties` die folgenden Eigenschaften fest:

```
com.ibm.di.server.encryption.keystore=server.jck
com.ibm.di.server.encryption.keystoretype=jceks
com.ibm.di.server.encryption.key.alias=server
com.ibm.di.server.encryption.transformation=AES/CBC/PKCS5Padding
```

- Migrieren Sie alle vorhandenen Dateien, die mit dem alten Schlüssel verschlüsselt wurden:

Alle verschlüsselten Dateien, die vor der Einführung des neuen Schlüssels vorhanden waren, müssen migriert werden. Zur Migration erfolgt eine Entschlüsselung mit dem alten Schlüssel und (optional) eine erneute Verschlüsselung mit dem neuen Schlüssel (siehe „Verschlüsselungsartefakte (Schlüssel, Zertifikate, Schlüssel Speicher, verschlüsselte Dateien) verwalten“ auf Seite 208). Die Datei `global.properties` können Sie beispielsweise folgendermaßen migrieren:

```
cryptoutils -input ../etc/global.properties -output ../etc/global.properties
-mode decrypt_props -keystore ../testserver.jks -storepass server -alias server
-transformation RSA -storetype jks -keypass server

cryptoutils -input ../etc/global.properties -output ../etc/global.properties
-mode encrypt_props -keystore ../server.jck -storepass mypass -alias server
-transformation AES/CBC/PKCS5Padding -storetype jceks -keypass mykeypass
```

- Generieren Sie die Stashdatei des IBM Security Directory Integrator-Servers (siehe „Stashdatei“ auf Seite 147) erneut, damit die neuen Kennwörter für den Schlüssel Speicher und den Verschlüsselungsschlüssel berücksichtigt werden. Hierzu können Sie das Dienstprogramm `createstash` verwenden, das sich im Ordner `bin` einer IBM Security Directory Integrator-Installation befindetet. Beispiel:
`createstash mypass mykeypass com.ibm.crypto.fips.provider.IBMJCEFIPS`
- Verwenden Sie in Ihren Lösungen ausschließlich FIPS-konforme IBM Security Directory Integrator-Komponenten. Die folgende Tabelle enthält eine entsprechende Liste:

Tabelle 21. FIPS-konforme Komponenten

Directory Integrator-Komponente	FIPS-Modus zulässig?	Anmerkungen
Connectors		
Änderungserkennungsconnector für Active Directory	Ja	Verwendet JSSE-Standardfactorys für SSL.
Fertigungslinienconnector	Ja	Wird als Client der Server-API ausgeführt.
Server-Connector für Axis Easy-Web-Service	Ja	Verwendet JSSE-Standardfactorys für SSL.
Befehlszeilenconnector	Ja	Bietet keine Verschlüsselungsfunktionen.
Domino/Lotus Notes-Connectors	Nein	Die Verschlüsselungsfunktionen von Domino/Notes 7 sind nicht FIPS-konform. (Notes 8.0.1 enthält möglicherweise eine gewisse Art der FIPS-Unterstützung.)
ITIM-DSMLv2-Connector	Ja	Verwendet JSSE-Standardfactorys für SSL.
DSMLv2-SOAP-Connector	Ja	Verwendet JSSE-Standardfactorys für SSL.
DSMLv2-SOAP-Server-Connector	Ja	Verwendet JSSE-Standardfactorys für SSL.
Exchange-Änderungsprotokollconnector	Ja	Verwendet JSSE-Standardfactorys für SSL.
Dateiconnector	Ja	Bietet keine Verschlüsselungsfunktionen.
FTP-Client-Connector	Ja	Bietet keine Verschlüsselungsfunktionen.
GLA-Connector	Ja	Bietet keine Verschlüsselungsfunktionen. Dieser Connector ist veraltet und wird in einer zukünftigen Version von IBM Security Directory Integrator entfernt.
HTTP-Client-Connector	Ja	Verwendet JSSE-Standardfactorys für SSL.
Alter HTTP-Client-Connector	Ja	Verwendet JSSE-Standardfactorys für SSL.
HTTP-Server-Connector	Ja	Verwendet JSSE-Standardfactorys für SSL.
Alter HTTP-Server-Connector	Ja	Bietet keine Verschlüsselungsfunktionen.
IBM Security Directory Integrator-Änderungsprotokollconnector	Ja	Verwendet JSSE-Standardfactorys für SSL.
ITIM-Agentenconnector	Ja	Bietet keine Verschlüsselungsfunktionen.

Tabelle 21. FIPS-konforme Komponenten (Forts.)

Directory Integrator-Komponente	FIPS-Modus zulässig?	Anmerkungen
JDBC-Connector	Teilweise	Falls keine Verschlüsselungsfunktionen (SSL, Verschlüsselung) verwendet werden, ist der Connector FIPS-konform. Andernfalls ist die FIPS-Konformität von der FIPS-Konformität der Verschlüsselungsfunktionalität des verwendeten JDBC-Treibers abhängig. Eine Erläuterung der FIPS-Konformität von JDBC-Treibern finden Sie unter „Connectors, Funktionskomponenten, Parser“ auf Seite 192.
JMS-Connector	Teilweise	Falls keine Verschlüsselungsfunktionen (SSL, Verschlüsselung) verwendet werden, ist der Connector FIPS-konform. Andernfalls ist die FIPS-Konformität von der FIPS-Konformität der Verschlüsselungsfunktionalität des verwendeten JDBC-Treibers abhängig. Eine Erläuterung der FIPS-Konformität von JMS-Providern finden Sie unter „Connectors, Funktionskomponenten, Parser“ auf Seite 192.
JMX-Connector	Ja	Bietet keine Verschlüsselungsfunktionen.
JNDI-Connector	Ja	Verwendet JSSE-Standardfactorys für SSL.
LDAP-Connector	Ja	Verwendet JSSE-Standardfactorys für SSL.
LDAP-Server-Connector	Ja	Verwendet JSSE-Standardfactorys für SSL.
Mailbox-Connector	Ja	Verwendet JSSE-Standardfactorys für SSL.
Speicherwarteschlangen-connector	Teilweise	Ist von der FIPS-Konformität des für den Systemspeicher verwendeten JDBC-Treibers abhängig. (Die Speicherwarteschlange verwendet den Systemspeicher zur Datenspeicherung.) Eine Erläuterung der FIPS-Konformität von JDBC-Treibern finden Sie unter „Connectors, Funktionskomponenten, Parser“ auf Seite 192.
Speicherdatenstrom-connector	Ja	Bietet keine Verschlüsselungsfunktionen.
IBM WebSphere MQ Everyplace-Kennwortspeicherconnector	Teilweise	Im FIPS-Modus ist für den Nachrichtenschutz nur PKCS#7 zulässig. Die Option für die RSA-Verschlüsselung darf nicht verwendet werden. Die IBM WebSphere MQ Everyplace-Minizertifikate sind nicht FIPS-konform und dürfen daher im FIPS-Modus nicht verwendet werden.
Änderungserkennungs-connector für Sun Directory	Ja	Verwendet JSSE-Standardfactorys für SSL.

Tabelle 21. FIPS-konforme Komponenten (Forts.)

Directory Integrator-Komponente	FIPS-Modus zulässig?	Anmerkungen
Eigenschaftsconnector	Teilweise	<p>Falls die Verschlüsselung inaktiviert ist, ist der Connector FIPS-konform.</p> <p>Andernfalls ist die FIPS-Konformität von dem für die Verschlüsselung verwendeten Chiffrierwert abhängig.</p> <p>Ein Beispiel für einen nach FIPS 140-2 genehmigten Chiffrierwert ist AES. Andere genehmigte Chiffrierwerte sind unter der folgenden Adresse aufgeführt: http://csrc.nist.gov/publications/fips/fips140-2/fips1402annexa.pdf</p> <p>Die Verschlüsselungsoption für den Server ist immer FIPS-konform, sofern IBM Security Directory Integrator für den FIPS-Modus korrekt konfiguriert ist. (Weitere Informationen finden Sie unter „FIPS-Modus aktivieren“ auf Seite 194.)</p>
Serverbenachrichtigungsconnector	Ja	Wird als Client der Server-API ausgeführt.
Systemwarteschlangenconnector	Teilweise	<p>Falls von der Systemwarteschlange keine Verschlüsselungsfunktionen (SSL, Verschlüsselung) verwendet werden, ist der Connector FIPS-konform.</p> <p>Andernfalls ist die FIPS-Konformität von der FIPS-Konformität des JMS-Providers abhängig, der von der Systemwarteschlange verwendet wird.</p> <p>Eine Erläuterung der FIPS-Konformität von JMS-Providern finden Sie unter „Connectors, Funktionskomponenten, Parser“ auf Seite 192.</p>
Connectors für Windows-Benutzer und -Gruppen	Ja	Bietet keine Verschlüsselungsfunktionen.
Systemspeicherconnector	Teilweise	Ist von der FIPS-Konformität des vom Systemspeicher verwendeten JDBC-Treibers abhängig.
RAC-Connector	Ja	Bietet keine Verschlüsselungsfunktionen. Dieser Connector ist veraltet und wird in einer zukünftigen Version von IBM Security Directory Integrator entfernt.
RDBMS-Änderungsprotokollconnector	Teilweise	Siehe Anmerkung zum JDBC-Connector.
SNMP-Connector	Ja	Bietet keine Verschlüsselungsfunktionen.
SNMP-Server-Connector	Ja	Bietet keine Verschlüsselungsfunktionen.
TAM-Connector	Ja	Die IBM Security Directory Integrator-Laufzeit für Java ist FIPS-konform.
TCP-Connector	Ja	Verwendet JSSE-Standardfactorys für SSL.
TCP-Server-Connector	Ja	Verwendet JSSE-Standardfactorys für SSL.

Table 21. FIPS-konforme Komponenten (Forts.)

Directory Integrator-Komponente	FIPS-Modus zulässig?	Anmerkungen
Zeitgeberconnector	Ja	Bietet keine Verschlüsselungsfunktionen.
URL-Connector	Ja	Bietet keine Verschlüsselungsfunktionen.
Server-Connector für Web-Service-Empfänger	Ja	Verwendet JSSE-Standardfactorys für SSL.
z/OS-Änderungsprotokollconnector	Ja	Verwendet JSSE-Standardfactorys für SSL.
Funktionskomponenten		
Funktionskomponente für Castor-Java in XML	Ja	Bietet keine Verschlüsselungsfunktionen.
Funktionskomponente für Castor-XML in Java	Ja	Bietet keine Verschlüsselungsfunktionen.
EMFXMLToSDO	Ja	Bietet keine Verschlüsselungsfunktionen.
EMFSDOToXML	Ja	Bietet keine Verschlüsselungsfunktionen.
Funktionskomponente für Fertigungslinien	Ja	Wird als Client der Server-API ausgeführt.
Funktionskomponente für Java-Klassen	Teilweise	Ist von der FIPS-Konformität der Java-Klasse abhängig, deren Methode durch die Funktionskomponente aufgerufen wird. Falls die Java-Klasse keine Verschlüsselung (SSL, Verschlüsselung, Signierung, verschlüsselte Hashfunktionen usw.) verwendet, kann sie im FIPS-Modus problemlos eingesetzt werden.
Funktionskomponente für Parser	Teilweise	Ist von der FIPS-Konformität des Parsers abhängig, der für die Funktionskomponente konfiguriert ist.
Funktionskomponente für CBE-Generator	Ja	Bietet keine Verschlüsselungsfunktionen.
SendEMail-Funktionskomponente	Ja	Verwendet JSSE-Standardfactorys für SSL.
Funktionskomponente für Speicherwarteschlange	Teilweise	Ist von der FIPS-Konformität des vom Systemspeicher verwendeten JDBC-Treibers abhängig. (Die Speicherwarteschlange verwendet den Systemspeicher zur Datenspeicherung.) Eine Erläuterung der FIPS-Konformität von JDBC-Treibern finden Sie unter „Connectors, Funktionskomponenten, Parser“ auf Seite 192.
Funktionskomponente für Axis Java in SOAP	Ja	Bietet keine Verschlüsselungsfunktionen.
WrapSOAP-Funktionskomponente	Ja	Bietet keine Verschlüsselungsfunktionen.
Funktionskomponente für den Aufruf von SOAP-WS	Ja	Verwendet JSSE-Standardfactorys für SSL.

Tabelle 21. FIPS-konforme Komponenten (Forts.)

Directory Integrator-Komponente	FIPS-Modus zulässig?	Anmerkungen
Funktionskomponente für Axis SOAP in Java	Ja	Bietet keine Verschlüsselungsfunktionen.
Funktionskomponente für Axis-EasyInvoke-SOAP-WS	Ja	Verwendet JSSE-Standardfactorys für SSL.
Funktionskomponente für Generator für komplexe Typen	Ja	Bietet keine Verschlüsselungsfunktionen.
Funktionskomponente für ferne Befehlszeile	Teilweise	Die Verschlüsselungsfunktionen des RXA-Toolkits sind nicht FIPS-konform. Falls keine Verschlüsselungsfunktionen verwendet werden, kann die Komponente im FIPS-Modus eingesetzt werden.
Funktionskomponente für z/OS-TSO/E-Befehlszeile	Teilweise	Ist von der FIPS-Konformität der Verschlüsselung abhängig, die in den durch die Funktionskomponente aufgerufenen TSO-Befehl einbezogen ist.
Komponentensuite für SAP-ABAP-Anwendungsserver	Nein	Das SAP-Verschlüsselungsmodul wurde nicht nach FIPS 140-2 zertifiziert. Falls keine Verschlüsselungsfunktionen verwendet werden, können die Komponenten im FIPS-Modus eingesetzt werden.
Parser	Ja	Keine der IBM Security Directory Integrator-Parserkomponenten verwendet eine Verschlüsselung, sodass alle Parser im FIPS-Modus eingesetzt werden können.

Eigenschaft "com.ibm.di.server.fipsmode.on" festlegen

Um den FIPS-Modus in IBM Security Directory Integrator zu aktivieren, müssen Sie in der Datei `global.properties` oder `solution.properties` eine bestimmte Eigenschaft angeben. Die Eigenschaft heißt `com.ibm.di.server.fipsmode.on` und kann entweder auf **true** oder auf **false** gesetzt werden. Wenn diese Eigenschaft auf **true** gesetzt ist, wird der IBM Security Directory Integrator-Server im FIPS-Modus ausgeführt. In diesem Modus wird der IBM FIPS-Sicherheitsprovider in der Providerliste der IBM Security Directory Integrator-JVM vor den IBM JCE-Sicherheitsprovider gestellt. Wenn die Eigenschaft für die FIPS-Aktivierung in IBM Security Directory Integrator auf "true" gesetzt ist, wird auch im IBM JSSE2-Provider der FIPS-Modus aktiviert und als Standardfactorys für SSL-Sockets von JSSE werden die Factorys aus dem IBM JSSE2-Provider festgelegt. Standardmäßig ist der FIPS-Modus in IBM Security Directory Integrator nicht aktiviert, die Eigenschaft `com.ibm.di.server.fipsmode.on` also auf **false** gesetzt.

Verschlüsselungsalgorithmen im FIPS-Modus verwenden

Es können ausschließlich FIPS-konforme Verschlüsselungsalgorithmen verwendet werden. Dies bedeutet, dass Sie ausschließlich FIPS-konforme Algorithmen verwenden dürfen, wenn der FIPS-konforme Modus erhalten bleiben soll. Bei Verwendung anderer Algorithmen wird die FIPS-Konformität nicht eingehalten.

Eigenschaft "com.ibm.di.securityTransformation" festlegen

Beim Öffnen einer verschlüsselten Konfiguration verwendet IBM Security Directory Integrator die Eigenschaft `com.ibm.di.securityTransformation`, um den Algorithmus abzurufen, mit dem die Konfiguration entschlüsselt wird. Falls für diese Eigenschaft ein Algorithmus angegeben ist, der nicht FIPS-konform ist, und der FIPS-Modus für den IBM Security Directory Integrator-Server aktiviert ist, wird eine Ausnahmebedingung ausgelöst. Die Ausnahmebedingungsnachricht lautet in etwa: `CTGDIC012E Could not load file<dateipfad>. No such algorithm: <algorithmusname>`.

Um diese Ausnahmebedingung zu verhindern, legen Sie bei einer Ausführung im FIPS-Modus immer FIPS-konforme Algorithmen für diese Eigenschaft fest. Standardmäßig ist die Eigenschaft `com.ibm.di.securityTransformation` auf `DES/ECB/Nopadding` gesetzt. Dieser Algorithmus ist **nicht** FIPS-konform. Diese Eigenschaft definiert außerdem einen Chiffrierwert für die kennwortbasierte Verschlüsselung und Entschlüsselung von IBM Security Directory Integrator-Konfigurationen.

Eigenschaften bei Ausführung im FIPS-Modus automatisch festlegen

- IBM Security Directory Integrator legt eine relevante Systemeigenschaft fest, die in der Datei `global.properties` nicht standardmäßig vorhanden ist. Diese Eigenschaft heißt `com.ibm.di.cryptoProvider` und ist bei der Ausführung im FIPS-Modus auf den Sicherheitsprovider `IBMJCEFIPS` gesetzt. Wenn diese Eigenschaft in der Datei `global.properties` festgelegt ist, müssen Sie beachten, dass dieser spezielle Wert verwendet wird. Ist für die Eigenschaft ein nicht FIPS-konformer Provider angegeben, führt dies dazu, dass IBM Security Directory Integrator selbst bei einer Ausführung im FIPS-Modus **nicht** FIPS-konform ist.
- Im FIPS-Modus werden bestimmte JSSE-Socket-Factorys verwendet. Dies sind die `IBMJSSE2-Socket-Factorys`. Der IBM Security Directory Integrator-Server führt dies automatisch durch, indem er die Eigenschaften `ssl.SocketFactory.provider` und `ssl.ServerSocketFactory.provider` auf die JSSE-Implementierungsklassen im Provider `IBMJSSE2` setzt.

Befehlszeilentool für Erstellung der Stashdatei im FIPS-Modus verwenden

Um eine Stashdatei zu erstellen, die mit den Standards aus FIPS 140-2 konform ist, müssen Sie die Providerklasse `IBMJCEFIPS` bei Verwendung des Dateitools `createtash` als dritten Parameter angeben. Beispiel:

```
tdi-installationsverzeichnis\bin\createtash Password Password com.ibm.crypto.fips.provider.IBMJCEFIPS
```

Alternativen zur RSA-Verschlüsselung im FIPS-Modus verwenden

Konfigurieren Sie IBM Security Directory Integrator im FIPS-Modus für die Verwendung von Advanced Encryption Standard (AES) anstelle des RSA-Verschlüsselungsalgorithmus. Ein Chiffrierwert für den geheimen Schlüssel, der mit FIPS 140-2 konform ist, ist erforderlich. Das Akronym "RSA" steht für "Rivest, Shamir, Adleman", den Erfindern des Algorithmus. Der RSA-Algorithmus ist ein starker Verschlüsselungsalgorithmus, der zum Senden von Daten über das Internet eingesetzt wird. Der RSA-Chiffrierwert darf Schlüssel für den Transport (SSL, TLS) nur so verschlüsseln, dass diese innerhalb der Grenzen des genehmigten Modus von FIPS 140-2 Level 1 bleiben (siehe <http://www.ibm.com/developerworks/java/jdk/security/60/FIPShowto.html>).

Externe Tools im FIPS-Modus ausführen:

Sie können die Befehlszeilensyntax zur Angabe des geeigneten Verschlüsselungsproviders und beim Generieren eines geheimen Schlüssels verwenden.

createstash

Übergeben Sie den nach FIPS 140-2 zertifizierten Verschlüsselungsprovider IBM-JCEFIPS in der Befehlszeile als expliziten Providerparameter:

```
createstash mypass mykeypass com.ibm.crypto.fips.provider.IBMJCEFIPS
```

cryptoutils

Übergeben Sie den nach FIPS 140-2 zertifizierten Verschlüsselungsprovider IBM-JCEFIPS in der Befehlszeile mit der Option *cryptoprotiderclass* folgendermaßen als expliziten Provider:

```
cryptoutils -input registry.txt -output registry.enc -mode encrypt -keystore ../testserver.jks -storepass server  
-alias server -cryptoprotiderclass com.ibm.crypto.fips.provider.IBMJCEFIPS
```

FIPS-Eigenschaften für IBM Security Directory Integrator konfigurieren:

Mit den hier aufgeführten Anweisungen können Sie FIPS-Eigenschaften für IBM Security Directory Integrator konfigurieren.

Dienstprogramm "keytool/Ikeyman" im FIPS-Modus ausführen

Um die Dienstprogramme "keytool" und "Ikeyman" im FIPS-Modus verwenden zu können, müssen Sie die Datei "java.security" im Verzeichnis *tdi-installationsverzeichnis/jvm/jre/lib/security* bearbeiten. Legen Sie in den ersten beiden Zeilen der Datei "java.security" zuerst den Provider IBMJCEFIPS und dann den Sicherheitsprovider IBMJCE fest. Beispiel:

```
security.provider.1=com.ibm.crypto.fips.provider.IBMJCEFIPS  
security.provider.2=com.ibm.crypto.provider.IBMJCE
```

Bei Solaris und HP-UX sollte jedoch in der Liste der Sicherheitsprovider an erster Stelle immer der Provider SUN angegeben sein.

SSL- und PKI-Zertifikate konfigurieren

Nachstehend erhalten Sie detaillierte Informationen zu SSL- und PKI-Zertifikaten.

IBM Security Directory Integrator verwendet sowohl Secure Sockets Layer (SSL) als auch Public Key Infrastructure (PKI) als Verschlüsselungsverfahren. SSL und PKI sind für viele IBM Security Directory Integrator-Komponenten und IBM Security Directory Integrator-Serverkomponenten eine wichtige Grundlage. SSL ermöglicht die Verschlüsselung und Authentifizierung des Datenaustausches im Netz zwischen zwei fernen kommunizierenden Partnern. Analog ermöglicht PKI Benutzern nicht gesicherter Netze einen sicheren und nicht öffentlichen Austausch von Daten, indem ein Paar aus einem öffentlichen und einem privaten Verschlüsselungsschlüssel verwendet wird, das über einen anerkannten Aussteller erhalten und gemeinsam genutzt wird. Weitere Informationen finden Sie unter „SSL- und PKI-Zertifikate konfigurieren“.

SSL-Zertifikat

Ein SSL-Zertifikat befindet sich auf einem sicheren Server und wird zum Verschlüsseln der Daten eingesetzt, die den Server identifizieren. Das SSL-Zertifikat

hilft dabei, die Zugehörigkeit zu der Entität, die es anfordert, nachzuweisen und enthält Informationen zum Zertifikatseigner, zur Domäne, für die das Zertifikat ausgestellt wurde, zum Namen der ausstellenden Zertifizierungsstelle sowie zum Stammelement und zum Land, in dem es ausgestellt wurde.

PKI-Zertifikat

Ein PKI-Zertifikat ermöglicht es Benutzern eines nicht gesicherten Netzes, den Datenaustausch sicherer und privater zu machen. PKI verwendet ein Verschlüsselungsschlüsselpaar, das über einen anerkannten Aussteller, der als "Zertifizierungsstelle" bezeichnet wird, erhalten und gemeinsam genutzt wird. Mit PKI können Sie ein Zertifikat, das eine Einzelperson oder eine Organisation identifiziert, sowie Verzeichnisservices erhalten, in denen die Zertifikate gespeichert werden. Die Zertifizierungsstelle kann die Zertifikate bei Bedarf auch widerrufen. Ein digitales Zertifikat wird am häufigsten eingesetzt, um zu prüfen, ob ein Benutzer, der eine Nachricht sendet, die vorgegebene Identität besitzt, und um für den Empfänger die Verschlüsselung der Antwort bereitzustellen.

Mit den folgenden Schritten können Sie separate Konfigurationsoptionen für Zertifikate angeben, die für die PKI-Verschlüsselung und für SSL verwendet werden:

1. Fügen Sie die folgenden Eigenschaften hinzu:

```
com.ibm.di.server.encryption.keystore  
com.ibm.di.server.encryption.key.alias  
api.keystore.password  
api.key.password
```

2. Benennen Sie die folgenden Eigenschaften wie gezeigt um:

```
com.ibm.di.server.keystore ---- > api.keystore  
com.ibm.di.server.key.alias ----->api.key.alias
```

Anmerkung: Die Datei `idisrv.sth` enthält nun lediglich das Kennwort für die Verschlüsselungsdatei.

Verschlüsselung und Entschlüsselung mit dem Dienstprogramm "CryptoUtils" vornehmen

Nachstehend erhalten Sie Informationen zur Verschlüsselung und Entschlüsselung mit dem Dienstprogramm "CryptoUtils".

Bei Verwendung von IBM Security Directory Integrator können Sie sensible Eigenschaften in der Datei `global.properties` oder `solution.properties` mit PKI verschlüsseln. Eines der Verfahren, das für die Entschlüsselung von mit PKI verschlüsselten Eigenschaften genutzt werden kann, ist die Verwendung des Eigenschafteneditors im Konfigurationeditor. Das Befehlszeilendienstprogramm "CryptoUtils" stellt eine weitere Methode für die Entschlüsselung von mit PKI verschlüsselten Eigenschaftendateien dar. Zur Entschlüsselung müssen Sie Ihre PKI-Berechtigungsnachweise angeben, damit nicht berechtigte Benutzer nicht auf sensible Informationen zugreifen können. Sie können Eigenschaftendateien, die mit PKI verschlüsselte Eigenschaften enthalten, mit dem Dienstprogramm "CryptoUtils" entschlüsseln. Weitere Informationen finden Sie unter „Verschlüsselungsdienstprogramm von IBM Security Directory Integrator“ auf Seite 153.

Mit Zertifikaten arbeiten

Nachstehend erhalten Sie Informationen zum Arbeiten mit Zertifikaten.

Ein Benutzer, der eine verschlüsselte Nachricht senden möchte, beantragt bei der Zertifizierungsstelle ein digitales Zertifikat. Die Zertifizierungsstelle stellt ein ver-

schlüsseltes digitales Zertifikat aus, das den öffentlichen Schlüssel des Antragstellers und weitere Identifikationsdaten enthält. Die Zertifizierungsstelle gibt ihren eigenen öffentlichen Schlüssel in Druckmedien oder möglicherweise über das Internet bekannt. Der Empfänger einer verschlüsselten Nachricht verwendet den öffentlichen Schlüssel der Zertifizierungsstelle, um das an die Nachricht angehängte digitale Zertifikat zu entschlüsseln, prüft, ob das Zertifikat durch die Zertifizierungsstelle ausgestellt wurde, und ruft dann aus dem Zertifikat den öffentlichen Schlüssel und die Identifikationsdaten des Absenders ab. Mit diesen Informationen ist der Empfänger in der Lage, eine verschlüsselte Antwort zu senden.

Es gibt zwei Typen von digitalen Zertifikaten:

- Von einer Zertifizierungsstelle signierte Zertifikate
- Selbstsignierte Zertifikate

Von einer Zertifizierungsstelle signierte Zertifikate werden durch eine Zertifizierungsstelle wie beispielsweise VeriSign und thawte signiert. Ein selbst signiertes Zertifikat ist ein Identitätszertifikat, das von seinem Ersteller selbst signiert wird.

IBM Security Directory Integrator bietet separate Konfigurationsoptionen für Zertifikate, die für die PKI-Verschlüsselung bzw. für SSL-Verbindungen verwendet werden. Durch die unabhängige Konfiguration von PKI- und SSL-Zertifikaten können Sie Ihre verschlüsselten Eigenschaften separat zum Upgradeprozess für Ihre SSL-Zertifikate migrieren.

Unter PKI bindet eine Zertifizierungsstelle öffentliche Schlüssel an Benutzeridentitäten. Die Benutzeridentität muss für jede Zertifizierungsstelle eindeutig sein. Zertifikate für öffentliche Schlüssel erfassen alle Informationen zum Benutzer, zur Benutzeridentität, zum öffentlichen Schlüssel, zur Bindung, zu Gültigkeitsbedingungen sowie weitere Attribute, die in Zertifikaten mit öffentlichen Schlüsseln, die durch die Zertifizierungsstelle ausgestellt werden, permanent gespeichert sind.

Die für SSL verwendeten Zertifikate können unter Umständen ablaufen. Möglicherweise müssen SSL-Zertifikate aus Sicherheitsgründen auch häufig aktualisiert werden. Zertifikate, die für die PKI-Verschlüsselung verwendet werden, können länger bestehen bleiben, als es bei SSL-Zertifikaten angemessen ist. PKI-Zertifikate sollten für den Fall, dass Daten mit dem Zertifikat für öffentlichen Schlüssel verschlüsselt wurden, aufbewahrt werden. Daher können Sie bei IBM Security Directory Integrator PKI- und SSL-Zertifikate separat konfigurieren. Jeder Server für eine SSL-Verbindung und jeder Client mit PKI-Authentifizierung muss bei der lokalen Zertifizierungsstelle ein Zertifikat beantragen und das resultierende Zertifikat in seinem Truststore speichern.

Die folgenden Eigenschaften werden zur Datei `global.properties` hinzugefügt:

```
com.ibm.di.server.encrypted.keystore
com.ibm.di.server.encrypted.key.alias
```

Die folgenden Eigenschaftensvariablen werden auf dieselben Werte gesetzt, die bereits in der Datei `global.properties` angegeben sind:

```
api.keystore=truststore
api.key.alias=server
```

Unterschiede zwischen selbst signierten und von Zertifizierungsstellen signierten Zertifikaten

Mithilfe der hier angegebenen Informationen können Sie selbst signierte und von Zertifizierungsstellen signierte Zertifikate vergleichen.

Von einer Zertifizierungsstelle signierte Zertifikate

Selbstsignierte Zertifikate

Zertifizierungsstellen wie beispielsweise VeriSign erfordern eine Prozedur, durch die Antragsteller ihre Identitäten nachweisen und Zertifikate erhalten können, die sowohl die Identität der Zertifikatantragsteller als auch ihre eigene Identität als Unterzeichner eines Zertifikats authentifizieren.

Normalerweise gibt es eine lokale Zertifizierungsstelle, die Zertifikate stammen also nicht von anerkannten Zertifizierungsstellen wie VeriSign usw. Für die lokale Zertifizierungsstelle selbst sollte durch eine anerkannte Zertifizierungsstelle ein Stammzertifikat ausgestellt worden sein. Dies ist jedoch nicht immer der Fall. Wenn es sich bei dem Stammzertifikat der lokalen Zertifizierungsstelle um ein selbst signiertes Zertifikat handelt, müssen Sie es in den Truststore jedes Servers oder Clients importieren, der SSL verwendet.

In diesem Fall generiert jeder Server für eine SSL-Verbindung und jeder Client für die PKI-Authentifizierung ein eigenes selbst signiertes Zertifikat. Anschließend muss das Zertifikat in eine Datei exportiert und in verschiedene Truststores importiert werden. Falls der Client C eine Verbindung zum Server S herstellt, muss der Truststore von C das selbst signierte Zertifikat von S enthalten. Führt der Client C eine PKI-Authentifizierung (symmetrisches SSL) beim Server S durch, muss der Truststore von S das selbst signierte Zertifikat von C enthalten. Anmerkung: Selbst signierte Zertifikate können als Clientzertifikat oder als Serverzertifikat verwendet werden. Entsprechende Verwendungshinweise finden Sie im Abschnitt „Schlüssel, Zertifikate und Schlüsselspeicher verwalten“ auf Seite 103. Jeder Server für eine SSL-Verbindung und jeder Client mit PKI-Authentifizierung muss dann bei der lokalen Zertifizierungsstelle ein Zertifikat beantragen und das resultierende Zertifikat in seinem Truststore speichern.

Zertifikate mit PKI und SSL konfigurieren

Nachstehend erhalten Sie Informationen zum Konfigurieren von Zertifikaten mit PKI und SSL.

IBM Security Directory Integrator bietet separate Konfigurationsoptionen für Zertifikate, die für die PKI-Verschlüsselung bzw. für SSL-Verbindungen verwendet werden. Durch die unabhängige Konfiguration von PKI- und SSL-Zertifikaten können Sie Ihre verschlüsselten Eigenschaften separat zum Upgradeprozess für Ihre SSL-Zertifikate migrieren.

Unter PKI bindet eine Zertifizierungsstelle öffentliche Schlüssel an Benutzeridentitäten. Die Benutzeridentität muss für jede Zertifizierungsstelle eindeutig sein. Zertifikate für öffentliche Schlüssel erfassen alle Informationen zum Benutzer, zur Benutzeridentität, zum öffentlichen Schlüssel, zur Bindung, zu Gültigkeitsbedingungen sowie weitere Attribute, die in Zertifikaten mit öffentlichen Schlüsseln, die durch die Zertifizierungsstelle ausgestellt werden, permanent gespeichert sind.

Die für SSL verwendeten Zertifikate können unter Umständen ablaufen. Möglicherweise müssen SSL-Zertifikate aus Sicherheitsgründen auch häufig aktualisiert werden. Zertifikate, die für die PKI-Verschlüsselung verwendet werden, können länger bestehen bleiben, als es bei SSL-Zertifikaten angemessen ist. PKI-Zertifikate sollten für den Fall, dass Daten mit dem Zertifikat für öffentlichen Schlüssel verschlüsselt wurden, aufbewahrt werden. Daher können Sie bei IBM Security Directory Integrator PKI- und SSL-Zertifikate separat konfigurieren. Jeder Server für eine SSL-Ver-

bindung und jeder Client mit PKI-Authentifizierung muss bei der lokalen Zertifizierungsstelle ein Zertifikat beantragen und das resultierende Zertifikat in seinem Truststore speichern.

Die folgenden Eigenschaften werden zur Datei `global.properties` hinzugefügt:

```
com.ibm.di.server.encryption.keystore  
com.ibm.di.server.encryption.key.alias
```

Die folgenden Eigenschaftensvariablen werden auf dieselben Werte gesetzt, die bereits in der Datei `global.properties` angegeben sind:

```
api.keystore=truststore  
api.key.alias=server
```

Verschlüsselungsschlüssel auf Hardwareeinheiten verwenden

Mit den hier aufgeführten Anweisungen können Sie Verschlüsselungsschlüssel verwenden, die sich auf Hardwareeinheiten befinden.

Der für Signierung und Verschlüsselung durch Ron Rivest, Adi Shamir und Leonard Adleman entwickelte RSA-Algorithmus ist ein anerkannter Chiffrierwert für öffentliche Schlüssel. RSA Laboratories (Teil der EMC Corp.) hat den Standard PKCS#11 veröffentlicht, in dem eine plattformunabhängige API für Hardwareverschlüsselungstoken (z. B. Hardwaresicherheitsmodule und Chipkarten) definiert ist. Die PKCS#11-API definiert die am häufigsten eingesetzten Typen von Verschlüsselungsobjekten, zu denen Folgende gehören:

- RSA-Schlüssel
- X.509-Zertifikate
- DES/Triple DES-Schlüssel (DES = Data Encryption Standard)
- Alle zum Verwenden, Erstellen oder Generieren, Modifizieren und Löschen der vorstehenden Objekte erforderlichen Funktionen

Der Standard "PKCS#11" von Public-Key Cryptography Standards (PKCS) stellt eine einheitliche Anwendungsschnittstelle für Verschlüsselungsservices auf unterschiedlichen Plattformen bereit, die verschiedenste Hardwareverschlüsselungseinheiten verwenden. Mit Schlüsselspeichereinheiten für die Hardwareverschlüsselung können Schlüssel auf Hardwareeinheiten gespeichert werden. IBM Security Directory Integrator unterstützt private Schlüssel und Zertifikate auf Verschlüsselungseinheiten, die PKCS#11-konform sind. Die Unterstützung wird für alle Hardwareeinheiten bereitgestellt, die durch die mit Java Runtime Environment (JRE) ausgelieferten IBM Java-PKCS-Bibliotheken unterstützt werden. PKCS-Standards sind eine Gruppe von einheitlichen Protokollen, die einen sicheren Informationsaustausch über Netze unter Verwendung von PKI (Public Key Infrastructure) ermöglichen. IBM Security Directory Integrator kann SSL-Schlüssel auf den Hardwareeinheiten speichern. Um die Anforderungen für die Speicherung von Schlüsseln auf Hardwareeinheiten zu erfüllen, sind die folgenden neuen Eigenschaften in der Datei `global.properties` verfügbar:

```
##PKCS11 options  
##Set the value of following properties to use PKCS11 enabled devices to store TDI servers  
##private key /certificate.  
com.ibm.di.pkcs11cfg=etc\pkcs11.cfg  
com.ibm.di.server.pkcs11=false  
com.ibm.di.server.pkcs11.library=  
com.ibm.di.server.pkcs11.slot=  
{protect}-com.ibm.di.server.pkcs11.password=PASSWORD
```

Der Standardwert der Eigenschaft `com.ibm.di.server.pkcs11` ist "false". Der Wert, der der Eigenschaft `com.ibm.di.server.pkcs11.password` entspricht, ist verschlüsselt.

IBMPCKS11 verwenden

Mit IBMPCKS11 können Sie auf Einheiten zugreifen sowie SSL-Schlüssel und -Zertifikate speichern.

IBM Security Directory Integrator verwendet IBMPCKS11, um auf Verschlüsselungshardwareeinheiten zuzugreifen, auf denen SSL-Schlüssel und -Zertifikate gespeichert werden. Die Unterstützung wird für alle Hardwareeinheiten bereitgestellt, die durch die mit Java Runtime Environment ausgelieferten IBM Java-PKCS-Bibliotheken unterstützt werden.

Tabelle 22. Bei SSL unterstützte Eigenschaften

Eigenschaft	Standardwert	Beschreibung
<code>com.ibm.di.pkcs11.cfg</code>	<code>etc\pkcs11.cfg</code>	Die CFG-Datei wird verwendet, um auf den Pfad der Konfigurationsdatei zu verweisen, die zur Initialisierung des IBM PKCS11-Implementierungsproviders benötigt wird.
<code>com.ibm.di.server.pkcs11</code>	false	Für SSL werden PKCS#11-konforme Verschlüsselungseinheiten verwendet.
<code>com.ibm.di.server.pkcs11.library</code>		Geben Sie mit dieser Eigenschaft den Pfad zur PKCS11-Clientbibliothek an.
<code>com.ibm.di.server.pkcs11.slot</code>		Geben Sie die Steckplatznummer der Einheit an.
<code>{protect}-com.ibm.di.server.pkcs11.pass</code>		Der Zugriff auf die PKCS11-konforme Verschlüsselungseinheit erfolgt unter Verwendung dieses Kennworts.
<code>com.ibm.di.server.pkcs11.accl</code>	false	Verwenden Sie den Wert "true", um Hardwareverschlüsselungseinheiten für Verschlüsselungsoperationen festzulegen.

Auffüllung aktivieren oder inaktivieren

Mithilfe der hier bereitgestellten Informationen und Links können Sie die Auffüllung aktivieren oder inaktivieren.

Beim Auffüllen werden zusätzliche Bit zu einer Übertragung hinzugefügt, damit die Übertragung exakt die erforderliche Größe hat. Bei einigen Verschlüsselungs- und Entschlüsselungsalgorithmen muss die Eingabe ein exaktes Vielfaches der Blockgröße sein. Falls der zu verschlüsselnde unverschlüsselte Text kein exaktes Vielfaches ist, müssen Sie vor der Verschlüsselung durch das Hinzufügen einer Auffüllungszeichenfolge den Text *auffüllen*. Teilen Sie dem Empfänger zur Entschlüsselung mit, wie die Auffüllung entfernt wird.

Anmerkung: Alle in der Datei `global.properties` aufgeführten Eigenschaften können in der Konfigurationsdatei mit demselben Namen festgelegt werden. Es empfiehlt sich, stattdessen die Datei `solution.properties` zu bearbeiten, falls eine solche Datei vorhanden ist. Diese Eigenschaften können durch Verwendung des Präfixes `{protect}-` mit einer Verschlüsselung geschützt werden (Details können Sie unter „Standardverschlüsselung der Datei 'global.properties' oder 'solution.properties'“ auf Seite 152 nachlesen).

Beim Festlegen der Eigenschaft für die Auffüllung lautet der Standardwert `DES/ECB/NoPadding`. Die Eigenschaft für die Auffüllung definiert einen Algorithmus oder Chiffrierwert für die kennwortbasierte Verschlüsselung und Entschlüsselung von IBM Security Directory Integrator-Konfigurationen. Sie heißt `com.ibm.di.securityTransformation`.

Verschlüsselungsartefakte (Schlüssel, Zertifikate, Schlüsselspeicher, verschlüsselte Dateien) verwalten

Nachstehend erhalten Sie Informationen zur Verwaltung von Verschlüsselungsartefakten (Schlüssel, Zertifikate, Schlüsselspeicher, verschlüsselte Dateien).

Anmerkung: Die SSL-Standardzertifikate von IBM Security Directory Integrator wurden in Version 7.1.1 geändert. Aus diesem Grund kann IBM Security Directory Integrator 7.1.1 die folgenden Elemente nicht entschlüsseln, wenn sie mit den IBM Security Directory Integrator-Standardzertifikaten verschlüsselt wurden:

- Verschlüsselte Kennwörter in "global.properties" oder "solution.properties"
- Geschützte Eigenschaften in externen Eigenschaftsspeichern
- Verschlüsselte XML-Konfigurationsdateien von IBM Security Directory Integrator in früheren Versionen

Aus diesem Grund müssen Sie alle verschlüsselten Kennwörter mit Ihrer vorherigen Version von IBM Security Directory Integrator entschlüsseln. Weitere Informationen zum Verschlüsselungsdienstprogramm finden Sie in „Verschlüsselungsdienstprogramm von IBM Security Directory Integrator“ auf Seite 153. Sobald die Kennwörter als Text abgerufen wurden, können Sie sie in der aktuellen Version verwenden. Die Kennwörter werden mit den neuen Standardzertifikaten verschlüsselt, sobald der Server oder der Konfigurationseditor gestartet wurde.

Geänderter Verschlüsselungsschlüssel

Jede Änderung des Schlüssels, den der Server für die Verschlüsselung verwendet, führt dazu, dass vorhandene verschlüsselte Dateien migriert werden müssen. Um eine verschlüsselte Datei zu migrieren, sollten Sie sie mit dem alten Verschlüsselungsschlüssel entschlüsseln und dann mit dem neuen Schlüssel verschlüsseln. Für die Verschlüsselung und Entschlüsselung können Sie das Tool „Verschlüsselungsdienstprogramm von IBM Security Directory Integrator“ auf Seite 153 verwenden.

Dateien, die häufig verschlüsselt werden oder verschlüsselte Teile enthalten, sind in den folgenden Abschnitten beschrieben: „Mit verschlüsselten IBM Security Directory Integrator-Konfigurationsdateien arbeiten“ auf Seite 149, „Benutzerregistry der Server-API“ auf Seite 140 und „Standardverschlüsselung der Datei 'global.properties' oder 'solution.properties'“ auf Seite 152 (Eigenschaftendateien von IBM Security Directory Integrator können verschlüsselte Eigenschaften enthalten, die Dateien selbst sind jedoch nicht verschlüsselt).

Anmerkung: Standardmäßig sind alle sensiblen Eigenschaften (z. B. Kennwörter) in der Datei `global.properties` oder `solution.properties` verschlüsselt. Es gilt die Faustregel, dass bei einer Änderung des Verschlüsselungsschlüssels für den Server die Dateien `global.properties` und `solution.properties` immer migriert werden sollten.

Geändertes Kennwort für Verschlüsselungsschlüssel oder Schlüsselspeicher

Der Server liest das Kennwort für den Schlüsselspeicher, der den Verschlüsselungsschlüssel enthält, sowie das Kennwort für den eigentlichen Verschlüsselungsschlüssel aus seiner „Stashdatei“ auf Seite 147. Falls eines dieser Kennwörter geändert wurde, muss die Stashdatei daher aktualisiert werden. Zu diesem Zweck können Sie das Tool `createstash` verwenden.

Abgelaufenes Verschlüsselungszertifikat

Falls der Server die Verschlüsselung mit öffentlichem Schlüssel verwendet, kann es sein, dass das Zertifikat, das dem Verschlüsselungsschlüsselpaar zugeordnet ist, zu einem bestimmten Zeitpunkt abläuft. In diesem Fall kann das Zertifikat mit der Prozedur verlängert werden, die im Abschnitt Gültigkeit eines Zertifikats mit "key-tool" verlängern beschrieben ist. Bei dieser Prozedur bleiben die zugrunde liegenden Schlüssel unverändert erhalten, sodass eine Migration vorhandener verschlüsselter Dateien nicht erforderlich ist.

Kapitel 10. IBM Security Directory Integrator-Server-API konfigurieren

Mithilfe der hier bereitgestellten Informationen und Links können Sie die IBM Security Directory Integrator-Server-API konfigurieren.

Die Server-API von IBM Security Directory Integrator stellt eine Reihe von Programmaufrufen bereit, die für die Entwicklung von IBM Security Directory Integrator-Lösungen sowie die lokale und ferne Interaktion mit dem Server verwendet werden können. Sie umfasst außerdem eine Verwaltungsschicht, die Server-API-Aufrufe über die Schnittstelle von JMX (Java Management Extensions) zugänglich macht. In diesem Abschnitt erhalten Sie Informationen zu den Eigenschaften, mit denen Sie die Server-API konfigurieren können.

- Informationen zur Server-API finden Sie im Handbuch *Referenzinformationen* in "Anhang C. Server-API".
- Zusätzliche Angaben über das Konfigurieren der Server-API finden Sie unter „Server-API konfigurieren“ auf Seite 120.
- Informationen zur Serversicherheit von IBM Security Directory Integrator können Sie dem Abschnitt „Ferne Server-API“ auf Seite 119 entnehmen.

Server-ID

Nachstehend erhalten Sie Informationen zu Server-IDs und deren Funktionsweise.

Ferne Clients können den Server, mit dem sie Daten austauschen, identifizieren, falls der Server anhand einer eindeutigen ID erkannt werden kann. IBM Security Directory Integrator lässt die Angabe von eindeutigen Server-IDs zu, die es fernen Clients wie beispielsweise AMC ermöglichen, zu unterschiedlichen Zeitpunkten unter Verwendung derselben IP-Adresse und desselben Ports Verbindungen zu verschiedenen IBM Security Directory Integrator-Servern herzustellen. Damit Verbindungen zwar mit derselben ID und demselben Port, aber zu unterschiedlichen Zeitpunkten hergestellt werden können, müssen IBM Security Directory Integrator-Clientanwendungen diese Clientanwendungen als verschiedene IBM Security Directory Integrator-Server registrieren können, denen Sie unterschiedliche Daten und Datenbanken zuordnen können.

Benutzer müssen die eindeutigen IDs manuell zuordnen. Dies stellt sicher, dass jeder ferne Client (z. B. AMC) basierend auf der IP-Adresse, dem Port und der eindeutigen ID des IBM Security Directory Integrator-Servers eine Verbindung zu einem IBM Security Directory Integrator-Server herstellen kann. AMC registriert jeden Server mit einer eindeutigen ID. Daher kann ein IBM Security Directory Integrator-Server weder versehentlich noch vorsätzlich mehr als ein Mal registriert werden. Bei der manuellen Zuordnung von IDs müssen Benutzer sicherstellen, dass die verschiedenen IBM Security Directory Integrator-Server unterscheidbare IDs besitzen.

Sie können die Eigenschaft für die eindeutige Server-ID mit `com.ibm.di.server.id` definieren. Wenn Sie für einen bestimmten Server eine eindeutige Server-ID vergeben wollen, geben Sie in der Datei `global.properties` oder `solution.properties` auf dem Server, den Sie identifizieren wollen, eine eindeutige ID-Zeichenfolge an. Der Standardwert für die Eigenschaft `com.ibm.di.server.id` ist ein leerer Wert.

Ausnahmebedingung für kennwortgeschützte Konfigurationen

Die Server-API gibt eine Ausnahmebedingung aus, wenn Sie eine kennwortgeschützte Konfiguration ohne Angabe eines Kennworts verwenden. Nachstehend erhalten Sie weitere Informationen dazu.

Die Server-API von IBM Security Directory Integrator kann Serverprobleme erkennen und verarbeiten, wenn der Server von Clients kontaktiert wird, die versuchen, ohne Angabe eines Kennworts auf kennwortgeschützte Konfigurationen zuzugreifen. In diesem Fall wird eine Nachricht angezeigt, die den Benutzer über das Problem benachrichtigt. Die Fehlnachricht wird aufgerufen, wenn kein Kennwort angegeben wird oder wenn das eingegebene Kennwort falsch ist. Weitere Informationen finden Sie unter „AMC und verschlüsselte Konfigurationen“ auf Seite 288.

Server-RMI

Nachstehend erhalten Sie Informationen und Links zur Server-RMI.

Aufgrund des gestiegenen Bedarfs an Fernzugriffen auf alle IBM Security Directory Integrator-Server ist RMI (Remote Method Invocation) standardmäßig aktiviert. Zur Gewährleistung einer adäquaten Sicherheit ist beim Fernzugriff SSL (Secure Sockets Layer) für die Clientauthentifizierung erforderlich. Der SSL-Zugriff wird durch die Beispiele für den Schlüsselspeicher und den Truststore unterstützt, die mit IBM Security Directory Integrator implementiert werden. Weitere Informationen können Sie den Abschnitten „SSL-Clientauthentifizierung“ auf Seite 114 und „Zusammenfassung der Optionen für die Server-API-Authentifizierung“ auf Seite 135 entnehmen.

Zeitlimitintervall für Konfigurationsladevorgang

Mit der Eigenschaft `api.config.timeout` können Sie ein Zeitlimitintervall hinzufügen.

Falls eine Konfigurationsinstanz die Konfigurationsdatei nicht vollständig geladen hat, wenn die Server-API einen Aufruf ausgibt, gibt die Server-API ein Nullobjekt zurück. Fügen Sie zum Hinzufügen eines Zeitlimitintervalls die Eigenschaft `api.config.timeout` in der Datei `global.properties` hinzu. Das Intervall für das Laden der Konfigurationsdatei ist standardmäßig mit zwei Minuten definiert. Falls die Konfigurationsdatei innerhalb des Zeitintervalls nicht geladen wird, wird eine Ausnahmebedingung ausgelöst.

Kapitel 11. Eigenschaften

Mithilfe von Eigenschaften können Sie IBM Security Directory Integrator-Komponenten und den IBM Security Directory Integrator-Server konfigurieren.

Eigenschaften sind einfache Paare im Format `schlüsselwort:wert` mit Parametern, die außerhalb der Konfigurationsdateien in externen Eigenschaftendateien gespeichert sind. Dies versetzt Sie in die Lage, vertrauliche Informationen wie beispielsweise Kennwörter außerhalb der Konfigurationsdateien zu speichern. Die Datei `global.properties` ist die primäre Konfigurationsdatei für IBM Security Directory Integrator. Eigenschaften werden in der Datei `global.properties` oder in der Datei `solution.properties` definiert. Die Datei `solutions.property` ist eine modifizierbare Kopie der Datei `global.properties`. Sie kommt zum Einsatz, wenn der Server aus dem Lösungsverzeichnis heraus gestartet wird. Falls während der Installation ein Lösungsverzeichnis angegeben wird, bei dem es sich nicht um das Installationsverzeichnis handelt, wird im Lösungsverzeichnis von IBM Security Directory Integrator eine Kopie der Datei `global.properties` unter dem Namen `solution.properties` erstellt. Beide Dateien sind Textdateien und so geschrieben, dass sie für das Betriebssystem, das auf der Plattform ausgeführt wird, verständlich sind.

Eigenschaften sind Datencontainer mit einem einzelnen Wert, die Parameterinformationen enthalten, z. B. `true` oder `5000`. Mit Eingabefunktionen wie `getProperty()` und `setProperty()` können Sie über Scripts auf Eigenschaften zugreifen. Die Methoden "get" und "set" arbeiten direkt mit Eigenschaftswerten. Eintragsobjekte können ebenfalls Eigenschaften enthalten. Wie Attribute sind auch Eigenschaften Datencontainer. Attribute werden verwendet, um Dateninhalt zu speichern. Eigenschaften hingegen enthalten parametrische Informationen. Eigenschaftswerte und Attribute können jeden beliebigen Java-Objekttyp besitzen. Eigenschaften werden bei Folgendem nicht angezeigt:

- Attributzuordnungsauswahl
- Work-Eintrag-Listen

Mit Eigenschaften arbeiten

Nachstehend erhalten Sie Informationen zum Arbeiten mit Lösungsverzeichnissen auf unterschiedliche Weise.

In diesem Abschnitt werden die primären Konzepte vorgestellt, die Sie beim Arbeiten mit Eigenschaften kennen müssen. Eigenschaften, die in einer beliebigen Datei `properties` festgelegt sind, bilden für alle Benutzer auf diesem Computer eine Referenzkonfiguration für die gesamte IBM Security Directory Integrator-Installation. Falls Sie jedoch ein Lösungsverzeichnis verwenden, bei dem es sich nicht um das Installationsverzeichnis handelt, können Sie in Ihrem Lösungsverzeichnis eine Gruppe von Textdateien verwenden, die ihre Entsprechungen im Installationsverzeichnis spiegeln. Eine Eigenschaft, die in einer dieser Dateien aufgeführt ist, setzt die Einstellung in einer der globalen Eigenschaftendateien der Installation (inklusive der Dateien `global.properties` und `solution.properties`) außer Kraft. Darüber hinaus hat eine Java-Eigenschaft, die in einer Konfigurationsdatei festgelegt ist, die höchste Vorrangstellung und setzt alle Angaben in einer globalen Eigenschaftendatei oder in den Eigenschaftendateien im Lösungsverzeichnis außer Kraft.

Zur Angabe des Lösungsverzeichnisses haben Sie mehrere Möglichkeiten:

- Sie können die Umgebungsvariable `TDI_SOLDIR` vor dem Starten des Konfigurationseditors oder des Servers festlegen.
- Sie können den Parameter `-s` für das Script `ibmditk` zum Starten des Konfigurationseditors oder für das Script `ibmdisrv` zum Starten des IBM Security Directory Integrator-Servers angeben. Dieses Verfahren hat Vorrang vor der Angabe der Umgebungsvariablen `TDI_SOLDIR`. Falls die Umgebungsvariable `TDI_SOLDIR` mit dem Installationsverzeichnis identisch ist, werden alle Eigenschaftendateien aus dieser Position gelesen und die Anmerkungen zu Eigenschaftendateien im Lösungsverzeichnis gelten nicht.

In allen anderen Fällen kopiert der IBM Security Directory Integrator-Server bei seiner erstmaligen Ausführung alle Eigenschaftendateien in Ihr Lösungsverzeichnis (gegebenenfalls bereits vorhandene Dateien werden hierbei nicht überschrieben). Anschließend können Sie diese Dateien ganz nach Bedarf und ohne Einfluss auf die Eigenschaftendateien im Installationsverzeichnis anpassen. Die im Installationsverzeichnis verbliebenen Dateien bilden weiterhin eine Referenzkonfiguration für andere IBM Security Directory Integrator-Instanzen.

Anmerkung: Die Datei `global.properties` wird in Ihrem Lösungsverzeichnis in eine Datei namens `solutions.properties` kopiert. Andere Dateien (z. B. `Log4J.properties` sowie die Dateien in den Ordnern `amc` und `serverapi`) werden jeweils unter ihrem eigenen Namen kopiert.

Zur Dokumentation wird die ursprüngliche Datei `global.properties` aus dem Installationsverzeichnis in den Ordner `<lösungsverzeichnis>/etc` kopiert. Diese Datei wird zu **keinem** anderen Zweck verwendet.

Migration mit Eigenschaften und dem Tool "tdimiggb1" ausführen

Das Tool "tdimiggb1" hilft Ihnen dabei, Ihre Datei `global.properties` aus einer Version von IBM Security Directory Integrator auf eine höhere Version zu migrieren. Entsprechende Informationen enthält der Abschnitt Kapitel 5, „Migration“, auf Seite 67.

Globale Eigenschaften

Sie können globale Eigenschaften dazu verwenden, die Einstellungen für den IBM Security Directory Integrator-Server zu konfigurieren, die in einer Datei namens `global.properties` im Ordner `etc` Ihres Installationsverzeichnisses gespeichert sind.

Alle in der Datei `global.properties` enthaltenen Eigenschaften sind im vorliegenden Abschnitt mit ihren Standardwerten aufgeführt und erläutert. Am Beginn der jeweiligen Eigenschaftengruppen finden Sie gegebenenfalls einen Verweis auf eine detailliertere Dokumentation. Der Konfigurationseditor (`ibmditk`) und der IBM Security Directory Integrator-Server (`ibmdisrv`) lesen die Datei `global.properties` beim Start. Diese Datei wird vor dem Lesen einer Datei namens `solution.properties` aus Ihrem Lösungsverzeichnis gelesen und angewendet.

Tabelle 23. Einige wichtige globale Eigenschaften von IBM Security Directory Integrator

Zweck der Eigenschaft	Eigenschaft	Standardwert	Beschreibung
Eigene JAR- oder ZIP-Dateien hinzufügen	com.ibm.di.loader.userjars	c:\myjars	Die Eigenschaft gibt Verzeichnisse oder JAR-Dateien durch die Java-Eigenschaft "path.separator" (also ":" bei Linux und ";" bei Windows) getrennt an. Verzeichnisse werden durch das TDI-Ladeprogramm rekursiv nach JAR-Dateien durchsucht, die Klassen und Ressourcen enthalten. Nur Dateien mit einer Erweiterung .zip oder .jar werden gesucht.
Chiffrierwert definieren	com.ibm.di.securityTransformation	DES/ECB/NoPadding	Diese Eigenschaft definiert einen Chiffrierwert für die kennwortbasierte Verschlüsselung oder Entschlüsselung von IBM Security Directory Integrator-Konfigurationen. Sie wurde in IBM Security Directory Integrator 7.0 geändert.
Automatisches Laden der Konfiguration aktivieren	com.ibm.di.server.autoLoad	autoload.tdi	Es wird in dem Verzeichnis, das durch den Befehl "ibmdisrv -d" angegeben wird, nach Dateien "*.xml" gesucht. Jede Datei "*.xml", die in dem durch den Parameter "-d" angegebenen Verzeichnis gefunden wird, wird ausgeführt.

Lösungseigenschaften

Lösungseigenschaften überschreiben normalerweise die globalen Eigenschaften und befinden sich in einer Datei mit dem Namen `solution.properties` in Ihrem Lösungsverzeichnis. Nachstehend erhalten Sie weitere Informationen dazu.

Die Datei `solution.properties` ist standardmäßig eine Kopie der Datei `global.properties`. Beim Konfigurieren von IBM Security Directory Integrator sollten Sie die Datei `solutions.properties` bearbeiten, da sie von allen Eigenschaftendateien als Letzte gelesen wird. Wenn Sie möchten, können Sie Eigenschaften in Ihrer Datei `solution.properties` löschen und Konfigurationsanweisungen für Eigenschaften hinzufügen, mit denen Standardeinstellungen der Datei `global.properties` ausdrücklich überschrieben werden sollen.

Java-Eigenschaften

Nachstehend erhalten Sie Informationen zu Java-Eigenschaften.

Java-Eigenschaften sind Variablen und Einstellungen der Java Virtual Machine (JVM). Eigenschaften für die Java-Protokolldatei (Jlog-Datei) sind im Abschnitt „Nützliche JLOG-Parameter“ auf Seite 266 erläutert.

Anmerkung: Eine Java-Eigenschaft, die in einer Konfigurationsdatei festgelegt ist, hat die höchste Vorrangstellung und setzt alle Angaben in einer globalen Eigenschaftendatei oder in den Eigenschaftendateien im Lösungsverzeichnis außer Kraft.

Tabelle 24. Java-Eigenschaften

Eigenschaft	Standardwert	Beschreibung
javax.net.debug	Keiner	Legt den Debugmodus für den JSSE-Provider fest.
com.ibm.di.javacmd	Keiner	Überschreibt den Java-Interpreter.
com.ibm.di.installDir	Keiner	Verwendet diesen Pfad zur ausführbaren Java-Datei, wenn Fertigungslinien aus dem Konfigurationseditor heraus ausgeführt werden.

Tabella 24. Java-Eigenschaften (Forts.)

Eigenschaft	Standardwert	Beschreibung
com.ibm.di.jvmdir	tdi-stammverzeichnis/jvm	Definiert den Verzeichnispfad, in dem die von IBM Security Directory Integrator verwendete JRE installiert ist.
com.ibm.di.server.maxThreadsRunning	500	Legt diese Anzahl von Threads in IBM Security Directory Integrator fest. Der Wert muss größer als 3 sein, damit er wirksam ist.
com.ibm.di.server.securemode	false	Legt den Modus fest, in dem IBM Security Directory Integrator ausgeführt wird (Standardmodus oder gesicherter Modus).
com.ibm.di.server.keystore	testserver.jks	Gibt den Schlüsselspeicher für das SSL-Zertifikat des Servers an. Wurde in IBM Security Directory Integrator 7.0 umbenannt.
com.ibm.di.server.key.alias	server	Gibt den Schlüsselaliasnamen für das SSL-Zertifikat des Servers an. Wurde in IBM Security Directory Integrator 7.0 umbenannt.
{protect}-api.keystore.password	server (standardmäßig verschlüsselt)	Gibt das Kennwort für den Schlüsselspeicher der Server-API an. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
{protect}-api.key.password		Gibt das Kennwort für den Schlüssel an. Wenn diese Eigenschaft nicht angegeben ist, wird das Kennwort für den Schlüsselspeicher des Servers verwendet. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.server.encryption.keystore	testserver.jks	Gibt den Schlüsselspeicher für den Verschlüsselungsschlüssel des Servers an. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.server.encryption.key.alias	server	Gibt den Schlüsselaliasnamen für den Verschlüsselungsschlüssel des Servers an. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.server.encryption.keystoretype	jks	Gibt den Typ des Schlüsselspeichers an, in dem sich der vom Server für die Verschlüsselung verwendete Schlüssel befindet. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.server.encryption.transformation	RSA	Gibt die Verschlüsselungsumsetzung an, die vom Server für die Verschlüsselung verwendet wird. Diese Eigenschaft kann entweder auf "RSA" (Verschlüsselung mit öffentlichem Schlüssel) oder auf eine Umsetzung mit geheimem Schlüssel gesetzt sein. Wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.server.fipsmode.on	false	Aktiviert oder inaktiviert die FIPS-Standards in IBM Security Directory Integrator. Wenn diese Eigenschaft auf "true" gesetzt ist, wird IBM Security Directory Integrator im FIPS-konformen Modus ausgeführt. Sie wurde in IBM Security Directory Integrator 7.0 hinzugefügt.
com.ibm.di.default.bind.address	*	Die Standardbindungsadresse für den gesamten IBM Security Directory Integrator-Server, also die Komponenten und die Server-API.

Systemeigenschaften

Systemeigenschaften sind nicht in einer externen Eigenschaftendatei wie `solution.properties`, sondern im Systemspeicher gespeichert. Nachstehend erhalten Sie weitere Informationen sowie einen Link dazu.

Bestimmte Systemeigenschaften und Java-Eigenschaften sind schreibgeschützt. Diese Systemeigenschaften werden in den entsprechenden Eigenschaftsspeichern (z. B.

Systemspeicher) angezeigt. Ein Versuch, diese schreibgeschützten Eigenschaften zu modifizieren, bleibt wirkungslos. Weitere Informationen finden Sie in Kapitel 12, „Systemspeicher“, auf Seite 219.

Kapitel 12. Systemspeicher

Nachstehend erhalten Sie Informationen zum Systemspeicher und dessen Funktionsweise.

IBM Security Directory Integrator unterstützt die permanente Speicherung (also die Speicherung von Objekten über JVM-Neustarts hinaus) durch eine eng verbundene relationale Datenbank, die als "Systemspeicher" bezeichnet wird.

Als Produkt für die Implementierung des Systemspeichers wird standardmäßig die vollständig in Java implementierte relationale Datenbank "IBM Security Directory Integrator" (früher "Cloudscape" genannt) verwendet.

Der Systemspeicher speichert die folgenden Objekte:

- Deltatabellen
- Eigenschaftsspeicher des Benutzers
- Kennwortspeicher

Standardposition des Systemspeichers

Die Standardposition der IBM Security Directory Integrator-Systemspeicherdatenbank im Netzmodus ist Ihr Lösungsverzeichnis. Deshalb können Sie für jedes Ihrer Lösungsverzeichnisse einen Systemspeicher haben.

Für die gemeinsame Nutzung eines einzigen Systemspeichers in allen Lösungsverzeichnissen müssen Sie den Wert `$lösungsverzeichnis$` durch das eigentliche `tdi-installationsverzeichnis` in der Eigenschaft `com.ibm.di.store.database` der Datei `global.properties` und der Datei `solution.properties` ersetzen, falls diese bereits erstellt wurden.

Aktualisieren Sie bei der Erstellung eines Lösungsverzeichnisses die folgenden Eigenschaften in der Datei `solution.properties` mit eindeutigen Werten, um Konflikte mit anderen Lösungsverzeichniseinstellungen zu vermeiden:

- `com.ibm.di.store.port=1527`
- `api.remote.naming.port=1099`
- `web.server.port=1098`
- Systemwarteschlangenport oder Active MQ-Port in `<lösungsverzeichnis>/etc/activemq.xml`

Anmerkung:

Das folgende Beispiel beschreibt die Auswirkung der Angabe ein und desselben Werts für die Eigenschaft `com.ibm.di.store.port` in mehreren Lösungsverzeichnissen.

Es gibt zwei Lösungsverzeichnisse (`lösungsverzeichnis1`, `lösungsverzeichnis2`) mit denselben Werten für `com.ibm.di.store.port` (1527, 1527) und eindeutigen Werten für `api.remote.naming.port` (1099, 41099).

Beim Start des Servers im Verzeichnis "lösungsverzeichnis1" wird der Server an Port 1099 und der Systemspeicher an Port 1527 gestartet, innerhalb des Verzeichnisses "lösungsverzeichnis1".

Beim Start des Servers im Verzeichnis "lösungsverzeichnis2" wird der Server an Port 41099 gestartet; er stellt eine Verbindung zum Systemspeicher her, welcher bereits an Port 1527 innerhalb des Verzeichnisses "lösungsverzeichnis1" empfangsbereit ist.

Im Rest dieses Abschnitts werden Betriebsaspekte beim Einsatz von IBM Security Directory Integrator behandelt, insbesondere bei der Verwendung von IBM Security Directory Integrator als Systemspeicher.

Anmerkung: Damit verschlüsselte Kennwortwerte gespeichert werden können, müssen Sie im Hinblick auf Managementsysteme für relationale Datenbanken anderer Anbieter die Felder relativ groß lassen. Ein typisches kompaktes Kennwort kann bis zu 178 Zeichen verwenden. Dies ist sowohl vom Schlüssel des Servers als auch von der Länge der unverschlüsselten Daten (in Byte) abhängig, die Sie speichern wollen. Da es sich um eine Blockverschlüsselung handelt, belegt ein längeres Kennwort möglicherweise denselben Platz oder aber auch das Zwei- oder Dreifache. Auch die Größe des Blocks ist vom Schlüssel des Servers abhängig. Um die benötigte Größe zu ermitteln, können Sie beispielsweise das (geschützte) Kennwort zunächst in einer Datei speichern und anschließend in dieser Datei feststellen, wie viele Zeichen verwendet wurden.

IBM Security Directory Integrator kann in zwei unterschiedlichen Modi ausgeführt werden, nämlich im *integrierten Modus* und im *Netzmodus*. Gemäß der Angabe in der Datei `global.properties` wird IBM Security Directory Integrator standardmäßig im *Netzmodus* ausgeführt.

Die IBM Security Directory Integrator-Releases vor Version 7.0 verwendeten als Systemspeicher IBM Security Directory Integrator (damals "Cloudscape" genannt) im *integrierten Modus*. Die Ausführung von IBM Security Directory Integrator im integrierten Modus ist mit einigen Nachteilen verbunden. Im integrierten Modus wird IBM Security Directory Integrator erforderlichenfalls als separater Thread in der JVM ausgeführt. Der Start und die Beendigung von IBM Security Directory Integrator erfolgen im integrierten Modus automatisch. Bei einer derartigen Ausführung fordert dieser IBM Security Directory Integrator-Thread jedoch einen exklusiven Zugriff auf die Datenbankdateien an. Dies kann problematisch werden, wenn verschiedene JVMs, die jeweils einen eigenen IBM Security Directory Integrator-Thread enthalten, versuchen, auf denselben Systemspeicher zuzugreifen.

Im integrierten Modus können die folgenden Aktionen das Starten einer neuen, unabhängigen JVM verursachen und lösen somit einen Zugriffskonflikt aus, weil gleichzeitig mehrere JVMs aktiv sind:

- Befehlszeilenaufruf des IBM Security Directory Integrator-Servers mit einer Konfigurationsdatei, die die Ausführung einer oder mehrerer Fertigungslinien auslöst
- Start des Konfigurationseditors (grafische Benutzerschnittstelle)
- Start einer Fertigungslinie aus dem Konfigurationseditor heraus

Keine dieser Aktionen bewirkt selbst das Starten des IBM Security Directory Integrator-Threads. Der IBM Security Directory Integrator-Thread wird jedoch gestartet, wenn ein Zugriff auf eines der Objekte im Systemspeicher erforderlich ist (z. B. vom Systemspeicher unterstützte Objekte wie Deltatabellen oder der Eigenschaftsspeicher des Benutzers).

Zur Lösung der oben dargestellten Zugriffskonflikte kann IBM Security Directory Integrator im *Netzmodus* ausgeführt werden. Dieser Modus ermöglicht einen gleichzeitigen Zugriff auf den Systemspeicher. Zur Vermeidung von Sicherheitsproblemen im Netzmodus sollten Sie auch die Benutzerauthentifizierung in IBM Security Directory Integrator aktivieren. IBM Security Directory Integrator verwendet den Sicherheitsprovider BUILTIN für IBM Security Directory Integrator, um die Sicherheit auf Datenbankebene zu gewährleisten. Der Provider BUILTIN stellt sicher, dass nur gültige Benutzer auf die IBM Security Directory Integrator-Datenbank zugreifen können. Wenn Sie IBM Security Directory Integrator im Netzmodus konfiguriert haben, können Sie mit mehreren Instanzen von IBM Security Directory Integrator-Datenbanken arbeiten, die als Systemspeicher gestartet werden. Außerdem können Sie eine IBM Security Directory Integrator-Instanz so konfigurieren, dass sie eine bestimmte Konfigurationsdateiinstanz verwendet.

Anmerkung: Abhängig davon, wie IBM Security Directory Integrator gestartet wurde, sind Instanzen von IBM Security Directory Integrator im Netzmodus auch dann weiterhin aktiv, nachdem alle anderen IBM Security Directory Integrator-Prozesse beendet wurden.

Wenn Sie die Eigenschaft `derby.drda.startNetworkServer` auf "true" setzen (Standardeinstellung in der Datei `global.properties`), wird beim Starten von IBM Security Directory Integrator automatisch der Netzserver gestartet (in diesem Kontext wird IBM Security Directory Integrator gestartet, nachdem der integrierte Treiber geladen wurde). Möglicherweise müssen Sie IBM Security Directory Integrator bei Bedarf manuell beenden.

Cloudscape-Befehlszeilendienstprogramm

Ziehen Sie zur Erleichterung der Arbeit mit der IBM Security Directory Integrator-Datenbank die Erstellung eines Scripts ("dbserver") in Betracht, das die folgende Zeile enthält (das vorliegende Beispiel gilt für Unix/Linux):

```
export DB_JAR_DIR=jars/3rdparty/IBM
export DB_CLASSPATH=$DB_JAR_DIR/derby.jar:$DB_JAR_DIR/derbyclient.jar:\
$DB_JAR_DIR/derbynet.jar:$DB_JAR_DIR/derbytools.jar
java -classpath $DB_CLASSPATH org.apache.derby.drda.NetworkServerControl "$@"
```

Möglicherweise müssen Sie die beiden mittleren Zeilen an der Stelle "\" verbinden.

Analog dazu würde die Datei "dbserver.bat" unter Windows wie folgt aussehen:

```
set DB_JAR_DIR=jars/3rdparty/IBM
set DB_CLASSPATH=%DB_JAR_DIR%\derby.jar;%DB_JAR_DIR%\derbyclient.jar;\
%DB_JAR_DIR%\derbynet.jar;%DB_JAR_DIR%\derbytools.jar;
java -classpath %DB_CLASSPATH% org.apache.derby.drda.NetworkServerControl %*
```

Anmerkung: Der Start des Scripts muss im Installationspfad von IBM Security Directory Integrator als Arbeitsverzeichnis erfolgen, da der folgende Klassenpfad relativ zu diesem Verzeichnis ist.

Nachstehend finden Sie ein Beispiel für die Syntax dieses Dienstprogrammscripts:

```
Show all available commands: ./dbserver

Start DBServer ./dbserver start -p 1527

Stop DBServer ./dbserver shutdown
```

Die vollständige Liste der Teilbefehle, die Sie für das Script "dbserver" ausführen können und die an IBM Security Directory Integrator gesendet werden, lautet wie folgt:

- `start [-h <host>] [-p <portnummer>]`: Startet den Netzserver auf dem angegebenen Port/Host oder auf dem lokalen Host, Port 1527, wenn kein Host/Port angegeben wird und keine Eigenschaften zum Überschreiben der Standardwerte festgelegt werden. Standardmäßig ist der Netzserver für Verbindungen nur auf dem System empfangsbereit, auf dem er ausgeführt wird. Verwenden Sie `-h 0.0.0.0` für die Empfangsbereitschaft auf allen Benutzerschnittstellen oder `-h <hostname>` für die Empfangsbereitschaft auf einer bestimmten Benutzerschnittstelle auf einem System mit mehreren IPs.
- `shutdown [-h <host>] [-p <portnummer>]`: Beendet den Netzserver auf dem angegebenen Host und Port oder auf dem lokalen Host und Port 1527 (Standardwert), wenn kein Host oder Port angegeben wird.
- `ping [-h <host>] [-p <portnummer>]`: Testet, ob der Netzserver betriebsbereit ist.
- `sysinfo [-h <host>] [-p <portnummer>]`: Druckt den Klassenpfad und Versionsinformationen zum Netzserver, zur JVM und zum Cloudscape-Server.
- `runtimeinfo [-h <host>] [-p <portnummer>]`: Druckt umfangreiche Debugging-Informationen zu Sitzungen, Threads, vorbereiteten Anweisungen und zur Speichernutzung für aktiven Netzserver.
- `logconnections {on | off} [-h <host>] [-p <portnummer>]`: Aktiviert und deaktiviert die Protokollierung des Auf- und Abbaus von Verbindungen. Der Verbindungsauf- und -abbau wird in der Datei `derby.log` protokolliert. Die Standardeinstellung ist "off".
- `maxthreads <max> [-h <host>] [-p <portnummer>]`: Legt fest, wie viele Threads maximal für Verbindungen verwendet werden können. Der Standardwert ist 0 (unbegrenzt).
- `timeslice <millisekunden> [-h <host>] [-p <portnummer>]`: Legt fest, wie lange jede Sitzung einen Verbindungsthread verwenden kann, bevor sie einer wartenden Sitzung den Vortritt lässt. Der Standardwert ist 0 (kein Vortritt).
- `trace {on | off} [-s <sitzungs-id>] [-h <host>] [-p <portnummer>]`: Aktiviert und deaktiviert die DRDA-Traceerstellung für die angegebene Sitzung oder - wenn keine Sitzung angegeben wird - für alle Sitzungen. Die Standardeinstellung ist "off".
- `tracedirectory <traceverzeichnis> [-h <host>] [-p <portnummer>]`: Ändert die Speicherposition neuer Tracedateien. Bei Sitzungen mit bereits aktivierter Traceerstellung bleiben die Tracedateien an der bisherigen Speicherposition. Die Standardeinstellung ist `cloudscape.system.home`.

Bei der Ausführung im Netzmodus ist die IBM Security Directory Integrator-Datenbank natürlich über das Netz erreichbar, nicht nur für IBM Security Directory Integrator-Instanzen, sondern auch für andere Anwendungen, die die entsprechenden Treiber verwenden. Die für einen solchen Zugriff erforderlichen Berechtigungsnachweise sind in der Datei `global.properties` definiert und müssen möglicherweise an die Anforderungen an Ihrem speziellen Standort angepasst werden. Achten Sie besonders auf die Parameter für den Benutzernamen und das Kennwort, da diese die Datenintegrität und -sicherheit bestimmen.

Wenn Sie bei der Ausführung von IBM Security Directory Integrator oft zwischen dediziertem Modus und Netzmodus wechseln, lohnt es sich unter Umständen, in Ihrem Dateisystem zwei verschiedene "Prototypdateien" `global.properties` zu erstellen, mit den korrekten Parametern für jeweils einen Modus. Kopieren Sie vor dem Start einer Serverinstanz je nach Bedarf die entsprechende Datei "global.properties" an die Speicherposition. Als Alternative können Sie separate Lösungsverzeichnisse verwenden. In einem Lösungsverzeichnis kann sich eine Datei namens

solution.properties befinden, deren Eigenschaftswerte die Werte überschreiben, die systemweit in der Datei global.properties definiert sind.

Eigenschaftsspeicher

Der Kennwortspeicher und der Eigenschaftsspeicher des Benutzers sind System-speichertypen.

Kennwortspeicher

Der *Kennwortspeicher* ist ein externes Repository, in dem ein Wert gespeichert wird, der durch die Änderung des Wertes für eine Komponente in der Kennwortsyntax entsteht. Der Kennwortschutzmechanismus ist direkt mit den Konfigurationsfenstern verbunden, die für einen Benutzer angezeigt werden. Die Konfigurationsfenster (also die Formulare) enthalten für jeden Parameter eine Beschreibung und seine Syntax. Einer der Typen für die Syntax ist der Typ *password*. Dieser Typ bewirkt, dass der Konfigurationseditor für die Bearbeitung ein Kennworttextfeld verwendet. Dieses externe Repository für Kennwörter wird im Konfigurationseditor auf der Seite *Eigenschaften (Kennwortspeicher)* konfiguriert und ist in der Konfigurationsdatei für die aktuelle IBM Security Directory Integrator-Lösung angegeben. Falls kein solcher Eigenschaftsspeicher konfiguriert ist, wird das Kennwort in der Konfigurationsdatei als unverschlüsselter Text gespeichert.

Wenn ein Standardkennwort konfiguriert ist, wird beim erstmaligen Speichern eines Parameters mit dem Typ "protected" oder "password" ein eindeutiger Eigenschaftsname generiert. Dieser Schlüssel wird als Schlüssel im Kennwortspeicher verwendet. Derselbe Eigenschaftsname wird als Standardeigenschaftsreferenz in die Konfigurationsdatei geschrieben. Wenn der Wert zu einem späteren Zeitpunkt abgerufen wird, findet eine Standardeigenschaftsauflösung statt, um den tatsächlichen Wert aus dem Kennwortspeicher abzurufen.

Falls ein Kennwortspeicher angegeben ist, wird für das Kennwort ein eindeutiger Schlüssel generiert und das Kennwort wird unter diesem Schlüssel im Kennwortspeicher in verschlüsselter Form abgelegt. In der Konfigurationsdatei wird das Kennwort nur durch diesen Schlüssel referenziert.

Eigenschaftsspeicher des Benutzers

Der Eigenschaftsspeicher des Benutzers ist eine Systemspeichertabelle, die zur Verwaltung serialisierter Java-Objekte verwendet wird, denen ein Schlüsselwert zugeordnet wurde. Hier werden persistente Komponentenparameter und -eigenschaften wie beispielsweise der **Speicher für Iteratorstatus** sowie von Ihnen gespeicherte Daten verwaltet. Der Systemspeicher implementiert Eigenschaftsspeicher des Benutzers als einen seiner drei Typen von persistenten Speichern für IBM Security Directory Integrator-Komponenten. Informationen zu IBM Security Directory Integrator-Benutzerschnittstellen, mit denen Sie Eigenschaften aus einem Eigenschaftsspeicher auswählen können, finden Sie unter „Lösungsansicht hinzufügen“ auf Seite 299.

Managementsystem für relationale Datenbanken eines anderen Anbieters als Systemspeicher

Sie können den Systemspeicher so konfigurieren, dass anstelle der im Produktpaket enthaltenen Datenbank (Apache Derby) andere mehrbenutzerfähige Managementsysteme für relationale Datenbanken verwendet werden.

Hierzu werden entsprechende SQL-DDL-Anweisungen und Treiberparameter als Systemeigenschaften in der Datei `global.properties` oder `solution.properties` angegeben. Die Vertriebsversion der Datei `global.properties` im Verzeichnis `tdi-installationsverzeichnis/etc` enthält auf Kommentar gesetzte Beispielanweisungen für die unterstützten Konfigurationen von IBM DB2, Oracle und MS SQL*Server.

Es ist ebenfalls möglich, passende Schablonen zu nutzen, die in den Konfigurationseditor integriert sind. Rufen Sie hierzu das entsprechende Dokument des IBM Security Directory Integrator-Servers auf. Klicken Sie im Teilfenster "Server" mit der rechten Maustaste auf den Server und wählen Sie die Option **Systemspeichereinstellungen bearbeiten** aus. Der Header **Serversystemspeicher** im Fenster ist ein kontextabhängiges Menü. Es enthält Optionen für Derby Embedded, Derby Networked, Oracle, DB2, MS SQL*Server 2005+ und IBM solidDB.

Anmerkung: Ein Systemspeicher kann im Konfigurationseditor auch projektbezogen konfiguriert werden. Die entsprechenden Einstellungen werden dann beim Exportieren des Projekts in der Konfigurationsdatei gespeichert und haben Vorrang vor dem für den Server definierten Systemspeicher.

Parameter für den JDBC-Treiber stellen einen Pfad zur Datenbank bereit. Zusätzliche Eigenschaften werden zur Angabe von angepasstem SQL für bestimmte Operationen verwendet, die IBM Security Directory Integrator im Systemspeicher ausführen können muss. Pro Eigenschaft können mehrere SQL-Anweisungen angegeben werden. Die einzelnen Anweisungen sollten durch ein Semikolon beendet werden. Im Folgenden ist ein Beispiel für eine Eigenschaft angegeben. Bitte beachten Sie, dass aus Darstellungsgründen die Anweisungen in diesem Dokument in mehrere Zeilen umgebrochen werden. In Ihrer Eigenschaftendatei müssen jedoch alle Anweisungen für eine jeweilige Eigenschaft in einer einzigen Zeile angegeben werden.

```
com.ibm.di.store.create.delta systable=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
SEQUENCEID int, VERSION int);
ALTER TABLE {0} ADD CONSTRAINT IDI_DS_{UNIQUE} PRIMARY KEY (ID)
```

Hierbei wird {0} durch den Tabellennamen ersetzt.

{UNIQUE} ist eine spezielle Variable, mit deren Hilfe ein eindeutiger Name generiert werden kann, der auf der aktuellen Systemzeit basiert.

Der folgende Abschnitt enthält zu jedem unterstützten Managementsystem für relationale Datenbanken Beispiele für Verbindungsparameter und Anweisungen.

Oracle

Bei Verwendung von Oracle müssen Sie die Clientbibliothek für den JDBC-Treiber (`ojdbc14.jar`) im Verzeichnis "`tdi-installationsverzeichnis/jars`" ablegen.

JDBC-Verbindungsparameter

```
com.ibm.di.store.database=jdbc:oracle:thin:@itdidev.in.ibm.com:1521:itimdb
com.ibm.di.store.jdbc.driver=oracle.jdbc.OracleDriver
com.ibm.di.store.jdbc.urlprefix=jdbc:oracle:thin:
com.ibm.di.store.jdbc.user=SYSTEM
{protect}-com.ibm.di.store.jdbc.password=password
```

Hierbei steht `itimdb` für die SID der Datenbank, die als Systemspeicher verwendet wird.

Anweisungen für die Tabellenerstellung

```

com.ibm.di.store.create.delta.systable=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
SEQUENCEID int, VERSION int);ALTER TABLE {0} ADD CONSTRAINT IDI_CS_{UNIQUE} PRIMARY KEY (ID)
com.ibm.di.store.create.delta.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
SEQUENCEID int, ENTRY BLOB );ALTER TABLE {0} ADD CONSTRAINT IDI_DS_{UNIQUE} Primary Key (ID)
com.ibm.di.store.create.property.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
ENTRY BLOB );ALTER TABLE {0} ADD CONSTRAINT IDI_PS_{UNIQUE} Primary Key (ID)
com.ibm.di.store.create.sandbox.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
ENTRY BLOB )
com.ibm.di.store.create.recal.conops=CREATE TABLE {0} (METHOD varchar(VARCHAR_LENGTH), RESULT BLOB,
ERROR BLOB)

```

MS SQL Server

Bei Verwendung von MS SQL Server sollten Sie eine Reihe von Microsoft-Clientbibliotheken im Verzeichnis *tdi-installationsverzeichnis/jars* installieren.

JDBC-Verbindungsparameter

```

com.ibm.di.store.database=jdbc:Microsoft:sqlserver://localhost:1433;DatabaseName=master;selectMethod=cursor;
com.ibm.di.store.jdbc.driver=com.microsoft.jdbc.sqlserver.SQLServerDriver
com.ibm.di.store.jdbc.user=sa
com.ibm.di.store.jdbc.password=passw0rd

```

Die obigen Verbindungsparameter werden mit den folgenden JAR-Dateien für Microsoft JDBC verwendet:

1. Msutil.jar
2. MsBase.jar
3. MSsqlserver.jar

Anmerkung: Bei Microsoft SQL Server 2008 muss im Verzeichnis *tdi-installationsverzeichnis/jars* die Treiber-JAR-Datei *sqljdbc.jar* gespeichert sein (es ist nur eine Datei erforderlich). Diese Datei finden Sie in Ihrer SQL Server 2008-Installation an der Position `<microsoft_sql_server_2008-installationsverzeichnis>/sqljdbc_<version>/<sprache>/sqljdbc.jar`. Die anzugebenden JDBC-Verbindungsparameter lauten wie folgt:

```

com.ibm.di.store.database=jdbc:sqlserver://localhost:1433;DatabaseName=name;selectMethod=cursor;
com.ibm.di.store.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
com.ibm.di.store.jdbc.user=sa
com.ibm.di.store.jdbc.password=passw0rd

```

Die Eigenschaft "selectMethod" ist für die JDBC-URL optional. Wenn diese Eigenschaft auf "cursor" gesetzt ist, wird ein Datenbankcursor erstellt. Dies ist hilfreich, wenn sehr große Ergebnismengen gelesen werden, die für den Hauptspeicher der Clients zu umfangreich sind.

Das Standardverhalten der Eigenschaft "selectMethod" ist nicht "cursor", sondern "direct". Bei diesem Wert werden Ergebnismengen im Hauptspeicher der Clients beibehalten, was schnellere Leistungswerte ergibt. Sofern der Hauptspeicher kein problematischer Aspekt ist, ist es daher besser, das Standardverhalten "direct" zu verwenden. Weitere Informationen finden Sie unter der folgenden Adresse: [http://msdn.microsoft.com/en-us/library/ms378988\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms378988(SQL.90).aspx).

JDBC-Verbindungsparameter (für JSQLConnect-Treiber)

```

com.ibm.di.store.database= jdbc:JSQLConnect://itdidriver/database=reqpro
com.ibm.di.store.jdbc.driver= com.jnetdirect.jsql.JSQLDriver
com.ibm.di.store.jdbc.urlprefix= jdbc:JSQLConnect:
com.ibm.di.store.jdbc.user=administrator
{protect}-com.ibm.di.store.jdbc.password=password

```

Diese Verbindungsparameter werden bei JSQLConnect-Treibern verwendet. Sie müssen die Datei "JSQLConnect.jar" herunterladen und in das Verzeichnis *tdi-installationsverzeichnis/jars* kopieren.

Anweisungen für die Tabellenerstellung

Der Datentyp für MS SQL ist IMAGE.

```
com.ibm.di.store.create.delta.systable=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
SEQUENCEID int, VERSION int);
ALTER TABLE {0} ADD CONSTRAINT IDI_MYCONSTRAINT {UNIQUE} PRIMARY KEY (ID)
com.ibm.di.store.create.delta.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
SEQUENCEID int, ENTRY IMAGE );
ALTER TABLE {0} ADD CONSTRAINT IDI_DS_{UNIQUE} Primary Key (ID)
com.ibm.di.store.create.property.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
ENTRY IMAGE );
ALTER TABLE {0} ADD CONSTRAINT IDI_PS_{UNIQUE} Primary Key (ID)
com.ibm.di.store.create.sandbox.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
ENTRY IMAGE)
com.ibm.di.store.create.recal.conops=CREATE TABLE {0} (METHOD varchar(VARCHAR_LENGTH),
RESULT IMAGE, ERROR IMAGE)
```

IBM DB2

Die hier aufgeführten Parameter und Anweisungen enthalten Informationen zur Funktionsweise von DB2.

JDBC-Verbindungsparameter

```
com.ibm.di.store.database=jdbc:db2:net://localhost:50000/idadb
com.ibm.di.store.jdbc.driver=com.ibm.db2.jcc.DB2Driver
com.ibm.di.store.jdbc.urlprefix= jdbc:db2:net:
com.ibm.di.store.jdbc.user=db2admin
{protect}-com.ibm.di.store.jdbc.password=db2admin
```

Hierbei steht *idadb* in der Datenbank-URL für den Standardsubsystemnamen einer DB2-Instanz.

Anweisungen für die Tabellenerstellung

```
com.ibm.di.store.create.delta.systable=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
SEQUENCEID int, VERSION int);
ALTER TABLE {0} ADD CONSTRAINT IDI_MYCONSTRAINT {UNIQUE} PRIMARY KEY (ID)
com.ibm.di.store.create.delta.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
SEQUENCEID int, ENTRY BLOB );
ALTER TABLE {0} ADD CONSTRAINT IDI_DS_{UNIQUE} Primary Key (ID)
com.ibm.di.store.create.property.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
ENTRY BLOB );ALTER TABLE {0} ADD CONSTRAINT IDI_PS_{UNIQUE} Primary Key (ID)
com.ibm.di.store.create.sandbox.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL,
ENTRY BLOB )
```

IBM solidDB

IBMsolidDB macht es erforderlich, dass die Datei "SolidDriver2.0.jar" im Verzeichnis *tdi-installationsverzeichnis/jars* gespeichert wird.

Diese JAR-Datei erhalten Sie in der IBMsolidDB-Installation (an der Position *soliddb-installationsverzeichnis/jdbc/SolidDriver2.0.jar*).

JDBC-Verbindungsparameter

```
com.ibm.di.store.database=jdbc:solid://localhost:1315
com.ibm.di.store.jdbc.driver=solid.jdbc.SolidDriver
com.ibm.di.store.jdbc.urlprefix=jdbc:solid:
com.ibm.di.store.jdbc.user=dba
{protect}-com.ibm.di.store.jdbc.password=dba
```

Anweisungen für die Tabellenerstellung

```
com.ibm.di.store.create.delta.systable=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH)
PRIMARY KEY NOT NULL, SEQUENCEID int, VERSION int)
com.ibm.di.store.create.delta.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH)
PRIMARY KEY NOT NULL, SEQUENCEID int, ENTRY BLOB)
com.ibm.di.store.create.property.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH)
PRIMARY KEY NOT NULL, ENTRY BLOB)
com.ibm.di.store.create.sandbox.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH)
NOT NULL, ENTRY BLOB)
com.ibm.di.store.create.recal.conops=CREATE TABLE {0} (METHOD VARCHAR(VARCHAR_LENGTH),
RESULT BLOB, ERROR BLOB)
```

Derby als Systemspeicher verwenden

Sie können Derby als Systemspeicher verwenden.

Im Rest dieses Abschnitts werden Betriebsaspekte beim Einsatz von Derby behandelt, insbesondere bei der Verwendung von Derby als Systemspeicher.

Anmerkung: Damit verschlüsselte Kennwortwerte gespeichert werden können, müssen Sie im Hinblick auf Managementsysteme für relationale Datenbanken anderer Anbieter die Felder relativ groß lassen. Ein typisches kompaktes Kennwort kann bis zu 178 Zeichen verwenden. Dies ist sowohl vom Schlüssel des Servers als auch von der Länge der unverschlüsselten Daten (in Byte) abhängig, die Sie speichern wollen. Da es sich um eine Blockverschlüsselung handelt, belegt ein längeres Kennwort möglicherweise denselben Platz oder aber auch das Zwei- oder Dreifache. Auch die Größe des Blocks ist vom Schlüssel des Servers abhängig. Um die benötigte Größe zu ermitteln, können Sie beispielsweise das (geschützte) Kennwort zunächst in einer Datei speichern und anschließend in dieser Datei feststellen, wie viele Zeichen verwendet wurden.

Derby kann in zwei unterschiedlichen Modi ausgeführt werden, nämlich im *integrierten Modus* und im *Netzmodus*. Gemäß der Angabe in der Datei `global.properties` wird Derby standardmäßig im *Netzmodus* ausgeführt.

Die IBM Security Directory Integrator-Releases vor Version 7.0 verwendeten als Systemspeicher Derby (damals "Cloudscape" genannt) im integrierten Modus. Die Ausführung von Derby im integrierten Modus ist mit einigen Nachteilen verbunden. Im integrierten Modus wird Derby erforderlichenfalls als separater Thread in der JVM ausgeführt. Der Start und die Beendigung von Derby erfolgen im integrierten Modus automatisch. Bei einer derartigen Ausführung fordert dieser Derby-Thread jedoch einen exklusiven Zugriff auf die Datenbankdateien an. Dies kann problematisch werden, wenn verschiedene JVMs, die jeweils einen eigenen Derby-Thread enthalten, versuchen, auf denselben Systemspeicher zuzugreifen.

Im integrierten Modus können die folgenden Aktionen das Starten einer neuen, unabhängigen JVM verursachen und lösen somit einen Zugriffskonflikt aus, weil gleichzeitig mehrere JVMs aktiv sind:

- Befehlszeilenaufruf des IBM Security Directory Integrator-Servers mit einer Konfigurationsdatei, die die Ausführung einer oder mehrerer Fertigungslinien auslöst
- Start des Konfigurationseditors (grafische Benutzerschnittstelle)
- Start einer Fertigungslinie aus dem Konfigurationseditor heraus

Keine dieser Aktionen bewirkt selbst das Starten des Derby-Threads. Der Derby-Thread wird jedoch gestartet, wenn ein Zugriff auf eines der Objekte im Systemspeicher erforderlich ist (z. B. vom Systemspeicher unterstützte Objekte wie Deltatabellen oder der Eigenschaftsspeicher des Benutzers).

Zur Lösung der oben dargestellten Zugriffskonflikte kann Derby im Netzmodus ausgeführt werden. Dieser Modus ermöglicht einen gleichzeitigen Zugriff auf den Systemspeicher. Zur Vermeidung von Sicherheitsproblemen im Netzmodus sollten Sie auch die Benutzerauthentifizierung in Derby aktivieren. IBM Security Directory Integrator verwendet den Sicherheitsprovider BUILTIN für Derby, um die Sicherheit auf Datenbankebene zu gewährleisten. Der Provider BUILTIN stellt sicher, dass nur gültige Benutzer auf die Derby-Datenbank zugreifen können. Wenn Sie Derby im Netzmodus konfiguriert haben, können Sie mit mehreren Instanzen von Derby-Datenbanken arbeiten, die als Systemspeicher gestartet werden. Außerdem können Sie eine Derby-Instanz so konfigurieren, dass sie eine bestimmte Konfigurationsdateiinstanz verwendet.

Anmerkung: Abhängig davon, wie Derby gestartet wurde, sind Instanzen von Derby im Netzmodus auch dann weiterhin aktiv, nachdem alle anderen IBM Security Directory Integrator-Prozesse beendet wurden. Wenn Sie die Eigenschaft `derby.drda.startNetworkServer` auf "true" setzen (Standardeinstellung in der Datei `global.properties`), wird beim Starten von Derby automatisch der Netzserver gestartet (in diesem Kontext wird Derby gestartet, nachdem der integrierte Treiber geladen wurde). Möglicherweise müssen Sie Derby bei Bedarf manuell beenden.

Apache Derby-Instanzen konfigurieren

Nachstehend erhalten Sie Informationen zum Konfigurieren von Apache Derby-Instanzen.

Zum Konfigurieren und Verwalten mehrerer Derby-Instanzen sowie zum Starten, Stoppen und erneuten Starten von Derby-Servern im Netzmodus enthält der Konfigurationseditor von IBM Security Directory Integrator eine Menüoption mit der Bezeichnung **Systemspeicher** (als Teil der Konfiguration **Lösungsprotokollierung und Einstellungen** eines Projekts). Viele der aufgeführten Konfigurationsoptionen übernehmen Standardwerte aus der Datei `global.properties`, die für Vorversionen von IBM Security Directory Integrator die Konfigurationsbasis bildete.

Die Menüoption **Systemspeicher** bietet auch die Möglichkeit, den Systemspeicher so zu konfigurieren, dass andere Datenbanken (z. B. IBM DB2) als Back-End-Managementsystem für relationale Datenbanken verwendet werden. Weitere Informationen enthält der Abschnitt "Systemspeichereinstellungen" unter **Konfigurationseditor -> Lösungsprotokollierung und Einstellungen** in der Veröffentlichung *Directory Integrator - Konfiguration*.

Apache Derby im Netzmodus starten

Mit den hier aufgeführten Anweisungen können Sie Apache Derby im Netzmodus starten.

Falls die Eigenschaft `com.ibm.di.store.hostname` auf `localhost` gesetzt ist, sind Fernverbindungen nicht zulässig. Ist die Eigenschaft `com.ibm.di.store.hostname` auf die IP-Adresse des lokalen Computers gesetzt, auf dem IBM Security Directory Integrator ausgeführt wird, können ferne Clients unter Verwendung der IP-Adresse auf diese Derby-Instanz zugreifen. Sie können nur den Netzserver für den lokalen Computer starten.

Tabelle 25. Apache Derby im Netzmodus starten

Eigenschaft	Standardwert	Beschreibung
<code>com.ibm.di.store.start.mode</code>	automatic	Der Modus für das Starten des Derby-Serverprozesses, wenn dies erforderlich ist. Kann auf "automatic" oder auf "manual" gesetzt sein.
<code>com.ibm.di.store.hostname</code>	localhost	Die URL des Derby-Servers.
<code>com.ibm.di.store.port</code>	1527	Der Port für die Verbindung zum Derby-Server.
<code>com.ibm.di.store.sysibm</code>	true	Gibt an, ob das Schema SYSIBM verwendet wird oder nicht. Gültige Werte sind "true" oder "false".
<code>com.ibm.di.store.varchar.length</code>	512	Der <code>varchar(length)</code> -Wert für die ID-Spalten, die in den Tabellen des Systemspeichers und des Systemspeicherconnectors (PES) verwendet werden.
<code>com.ibm.di.store.database</code>	<code>jdbc:derby://localhost:1527/\$soldir\$/TDISysStore;create=true</code>	Legt das Lösungsverzeichnis als Standardposition der Systemspeicherdatenbank fest. Anmerkung: Ersetzen Sie den Wert "\$soldir\$" nicht durch den absoluten Pfad des Lösungsverzeichnisses. Der Pfad wird bei der Laufzeit in JVM automatisch aktualisiert.

Benutzerauthentifizierung im Systemspeicher aktivieren

Sie können die folgenden Eigenschaften in der Datei `global.properties` nach den Eigenschaften für den Netzmodus des Systemspeichers hinzufügen.

Tabelle 26. Benutzerauthentifizierung im Systemspeicher aktivieren

Eigenschaft	Standardwert	Beschreibung
<code>derby.connection.requireAuthentication</code>	<code>true</code>	Aktiviert die Benutzerauthentifizierung für den Systemspeicher.
<code>derby.authentication.provider</code>	<code>BUILTIN</code>	Legt <code>BUILTIN</code> als Provider für die Benutzerauthentifizierung fest. Dies ist der grundlegendste und einfachste Authentifizierungsprovider von Derby.
<code>derby.database.defaultConnectionMode</code>	<code>fullAccess</code>	Definiert die Zugriffsebene für den Systemspeicherbenutzer. Die unterschiedlichen, von Derby unterstützten Zugriffsebenen sind "fullAccess" (uneingeschränkter Zugriff), "readOnly" (Lesezugriff) und "noAccess" (kein Zugriff).

Erstellungsanweisungen für Systemspeichertabellen

Sie können SQL-Anweisungen zur Tabellenerstellung (`CREATE TABLE`) für die hier aufgeführten Objekte konfigurieren:

- Deltasystemtabelle
- Deltatabelle
- Eigenschaftentabelle
- Sandboxtabellen
- Tabelle für Fertigungslinienaufzeichnung
- Tombstone Manager-Tabelle
- Spaltenname `ibmsnap_commitseq`

Tabelle 27. Erstellungsanweisungen für Systemspeicher

Eigenschaft	Standardwert	Beschreibung
<code>com.ibm.di.store.create.delta.systable</code>	<pre>CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL, SEQUENCEID int, VERSION int); ALTER TABLE {0} ADD CONSTRAINT IDI_CS_{UNIQUE} PRIMARY KEY (ID)</pre>	SQL-Anweisungen <code>CREATE TABLE</code> für die Deltasystemtabelle.
<code>com.ibm.di.store.create.delta.store</code>	<pre>CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL, SEQUENCEID int, ENTRY BLOB); ALTER TABLE {0} ADD CONSTRAINT IDI_DS_{UNIQUE} Primary Key (ID)</pre>	SQL-Anweisungen <code>CREATE TABLE</code> für die Deltatabelle.
<code>com.ibm.di.store.create.property.store</code>	<pre>CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL, ENTRY BLOB); ALTER TABLE {0} ADD CONSTRAINT IDI_PS_{UNIQUE} Primary Key (ID)</pre>	SQL-Anweisungen <code>CREATE TABLE</code> für die Eigenschaftentabelle.
<code>com.ibm.di.store.create.sandbox.store</code>	<pre>CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL, ENTRY BLOB)</pre>	SQL-Anweisungen <code>CREATE TABLE</code> für die Sandboxtabellen.

Tabelle 27. Erstellungsanweisungen für Systemspeicher (Forts.)

Eigenschaft	Standardwert	Beschreibung
com.ibm.di.store.create.recal.conops	CREATE TABLE {0} (METHOD varchar (VARCHAR_LENGTH), RESULT BLOB, ERROR BLOB)	SQL-Anweisungen CREATE TABLE für die Fertigungslinienaufzeichnung.
com.ibm.di.store.create.tombstones	CREATE TABLE IDI_TOMBSTONE (ID INT GENERATED ALWAYS AS IDENTITY, COMPONENT_TYPE_ID INT, EVENT_TYPE_ID INT, START_TIME_TIMESTAMP, CREATED_ON_TIMESTAMP, COMPONENT_NAME VARCHAR(1024), CONFIGURATION VARCHAR(1024), EXIT_CODE INT, ERROR_DESCR VARCHAR(1024), STATS LONG VARCHAR FOR BIT DATA, GUID VARCHAR(1024) NOT NULL, USER_MESSAGE VARCHAR(1024), UNIQUE (ID, GUID))	Geben Sie die SQL-Anweisung für die Erstellung der Tombstone Manager-Tabelle an. Behalten Sie dieselben Tabellennamen und Feldnamen bei.
com.ibm.di.conn.rdbmschlog.cdcolname	ibmsnap_commitseq	Geben Sie an, dass der Spaltenname "ibmsnap_commitseq" durch den RDBMS-Änderungsprotokollconnector verwendet wird.

Apache Derby-Datenbanken sichern

Mit der hier aufgeführten Methode können Sie Apache Derby-Datenbanken sichern.

Ein weiterer Aspekt, den Sie beachten müssen, ist die **Sicherung** der Daten, die in einer Derby-Datenbank enthalten sind. Das empfohlene (und einfachste) Verfahren ist im Folgenden erläutert:

- Beenden Sie die Derby-Datenbank (beenden Sie bei einer Ausführung im integrierten Modus alle IBM Security Directory Integrator-Instanzen und Konfigurationseditorinstanzen).
- Kopieren Sie das gesamte Derby-Verzeichnis in Ihrem IBM Security Directory Integrator-Ausgangsverzeichnis (bzw. das Derby-Verzeichnis, auf das die Datei `global.properties` verweist) an eine andere Position und stellen Sie sicher, dass diese Daten dort sicher sind.
- Starten Sie die Derby-Datenbank erneut (bei Ausführung im Netzmodus).

Zum Wiederherstellen einer Datenbank kehren Sie die Quelle und das Ziel der Kopieroperation in den obigen Schritten um.

Fehlerbehebung für Apache Derby

Der vorliegende Abschnitt ist nicht dazu gedacht, umfassende Fehlerbehebungsrichtlinien für Derby bereitzustellen, sondern behandelt lediglich eine Reihe von Symptomen, die unter Umständen bei der Verwendung von Derby als Basisdatenbank in IBM Security Directory Integrator zu beobachten sind.

Es handelt sich um die folgenden Elemente:

Schema "SYSIBM" ist nicht vorhanden

Frage

Bei dem Versuch, Derby im Netzmodus zu verwenden, treten Probleme auf. Die Datenbank kann zwar gestartet und mit "sysinfo" und "testconnection" abgefragt werden, aber bei dem Versuch, IBM Security Directory Integrator auszuführen und den Systemspeicher zu öffnen, wird der folgende Fehler ausgegeben:

```
[com.ibm.db2.jcc.a.SQLException: Schema 'SYSIBM' does not exist]
```

Wie kann dieser Fehler behoben werden?

Erläuterung

Dieser Fehler wird ausgegeben, weil versucht wird, eine im integrierten Modus erstellte Datenbank in einem Server mit Netzmodus zu starten, ohne dass der Server mit dem Parameter "-ld" gestartet wurde. Damit ein Derby-Server im Netzmodus eine Datenbank öffnen kann, die im integrierten Modus erstellt wurde, MUSS das Schema SYSIBM geladen worden sein. Das Schema SYSIBM ist ein spezielles, vom Derby-Server geladenes Schema. In diesem Schema sind vorbereitete Anweisungen gespeichert, die Ergebnismengen für die Ermittlung von Metadateninformationen zurückgeben.

Korrekturmaßnahme

Um dieses Problem zu lösen, starten Sie den Derby-Netzserver mit dem Parameter "-ld". Beispiel:

```
./dbserver start -p 1527 -ld
```

Weitere Derby-Instanz wurde möglicherweise bereits gebootet

Es kann sein, dass manchmal der folgende Fehler ausgegeben wird, insbesondere bei Verwendung von Derby im integrierten Modus:

```
[ERROR XSDB6: Another instance of Derby may have already booted the database D:\tdi60\Derby.]
```

Erläuterung

Derby versucht zu verhindern, dass zwei Derby-Instanzen dieselbe Datenbank (hier D:\tdi60\Derby) booten. Dies kann vorkommen, wenn Sie zwei Fertigungslinien ausführen, die versuchen, dieselbe im integrierten Modus ausgeführte Derby-Datenbank zu aktualisieren. Dieser Fehler kann außerdem auftreten, wenn eine Verbindung zur Datenbank nicht geschlossen wurde.

Korrekturmaßnahme

Wenn zwei Fertigungslinien dieselbe Derby-Datenbank aktualisieren sollen, sollte Derby korrekterweise im Netzmodus ausgeführt werden. Bei dieser Betriebsart ist diese Einschränkung nicht gegeben.

Als Ausweidlösung können Sie die Datenbank schließen, indem Sie die Option **Serverspeicher durchsuchen** verwenden und dann auf die Schaltfläche **Schließen** klicken. Auch wenn die Datenbank nicht geöffnet ist, kann das bloße Öffnen und erneute Schließen über die Option **Serverspeicher durchsuchen** zur Lösung dieses Problems hilfreich sein.

Künftige Versionen von IBM Security Directory Integrator versuchen, diese Situation automatisch zu verarbeiten und Derby nach Bedarf zu starten und zu stoppen.

Kann DB2 als Systemspeicher verwendet werden?

Es ist möglich, DB2 in IBM Security Directory Integrator anstelle des im Produktpaket enthaltenen Derby-Datenbanksystems als Systemspeicher zu verwenden. Damit eine ordnungsgemäße Funktionsweise gewährleistet ist, müssen hierzu jedoch einige Systemeigenschaftendateien modifiziert werden. Sie müssen den Abschnitt für den Derby-Netzmodus durch einen Abschnitt wie den Folgenden ersetzen (fügen Sie die für Ihre Installation korrekten Parameter ein).

In der Standarddatei `global.properties` finden Sie einige Anweisungen `CREATE TABLE` für die Verwendung und Konfiguration des Systemspeichers. Wenn Sie die richtige Syntax verwenden, können Sie andere Datenbanken als Derby für den Systemspeicher verwenden. Die folgende Syntax gilt bei DB2:

```
## Location of the DB2 database (networked mode)
com.ibm.di.store.database=jdbc:db2://168.199.48.4:3700/tdidb
com.ibm.di.store.jdbc.driver=com.ibm.db2.jcc.DB2Driver
com.ibm.di.store.jdbc.urlprefix=jdbc:db2:
com.ibm.di.store.jdbc.user=db2inst1
com.ibm.di.store.jdbc.password=*****
com.ibm.di.store.start.mode=automatic
com.ibm.di.store.port=3700
com.ibm.di.store.sysibm=true

# the varchar(length) for the ID columns used in system store and PES Connector tables
com.ibm.di.store.varchar.length=512

# create statements for DB2 system store tables
com.ibm.di.store.create.delta.systable=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH)
NOT NULL, SEQUENCEID int, VERSION int)
com.ibm.di.store.create.delta.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH)
NOT NULL, SEQUENCEID int, ENTRY BLOB )
com.ibm.di.store.create.property.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH)
NOT NULL, ENTRY BLOB )
com.ibm.di.store.create.sandbox.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH)
NOT NULL, ENTRY BLOB )
```

Anmerkung: Jede Anweisung `com.ibm.di.store.create.xxx` muss in einer einzigen Zeile angegeben werden (im Beispiel sind die Zeilen zur besseren Lesbarkeit umgebrochen).

Warum können zum Derby-Netzserver keine Fernverbindungen hergestellt werden?

Dies kann daran liegen, dass beim Starten des Derby-Servers der Wert "localhost" als Hostname übergeben wurde. In diesem Fall sind Fernverbindungen zu Derby nicht zulässig. Stoppen Sie den Derby-Server und starten Sie ihn erneut, indem Sie die IP-Adresse des Computers als Hostnamenparameter angeben. Hierzu können Sie das Servereinstellungsfenster **Serversystemspeicher** im Konfigurationseditor verwenden (dieses Fenster ist über das Kontextmenü eines Servers in der Ansicht "Server" verfügbar).

Weitere Details finden Sie unter der folgenden Adresse: <http://db.apache.org/derby/docs/10.5/adminguide/tadmincbdjhhd.html>

Kapitel 13. Befehlszeilenooptionen

Bei Befehlszeilenooptionen muss der Wert unmittelbar nach der Option angegeben werden. Geben Sie zwischen der Option und dem Wert kein Leerzeichen an.

Es gibt Optionen für die folgenden Elemente:

- „Konfigurationseditor“
- „Server“ auf Seite 234
- „CLI - Dienstprogramm "tdisrvctl"“ auf Seite 238

Konfigurationseditor

Nachstehend erhalten Sie Informationen zum Konfigurationseditor und den erforderlichen Eigenschaften.

Der Konfigurationseditor wird unter Verwendung des Wrapper-Scripts `ibmditk` gestartet. Dieses Script ruft das Eclipse-Startprogramm für IBM Security Directory Integrator (`ce/eclipsece/miadmin`) mit den richtigen Einstellungen für die Java-VM und die Eigenschaft für die IBM Security Directory Integrator-Installationsposition auf. Beide Angaben sind erforderlich, damit der aktuelle Konfigurationseditor ausgeführt wird.

Das Eclipse-Startprogramm (`ce/eclipsece/miadmin`) ist ein Eclipse-Standardstartprogramm, das eigene Befehlszeilenparameter verwendet. Eine vollständige Beschreibung der Eclipse-Befehlszeilenooptionen finden Sie im Abschnitt "Eclipse-Befehlszeilenooptionen".

```
"%TDI_HOME_DIR%\ce\eclipsece\miadmin" -vm "%TDI_JAVA_BIN_DIR%\javaw" -vmargs -Dcom.ibm.di.loader.IDILoader.path=%TDI_HOME_DIR% %*
```

Das obige Fragment des Scripts `ibmditk` zeigt die beiden erforderlichen Parameter (Eclipse-Befehlszeilenparameter), die der Konfigurationseditor benötigt.

Interessant ist besonders die Befehlszeilenooption `"-data"`, mit der die Position des zu verwendenden Arbeitsbereichs angegeben wird. Falls Sie mehrere Instanzen des Konfigurationseditors ausführen wollen, müssen Sie für jede Instanz des Konfigurationseditors einen anderen Arbeitsbereich angeben, da der Arbeitsbereich von einer Instanz jeweils gesperrt wird. Beispiel:

```
ibmditk -data c:/arbeitsbereich_instanz1
```

Der obige Befehl startet den Konfigurationseditor mit der Arbeitsbereichsposition `c:/arbeitsbereich_instanz1`.

Option für Serverbeendigung

Dies ist eine Befehlszeilenooption, die versucht, alle aktiven Server zu stoppen, von denen dasselbe Installationsverzeichnis wie vom Konfigurationseditor verwendet wird. Wenn diese Option in der Befehlszeile mit

```
ibmditk -tdishutdown
```

angegeben wird, startet der Konfigurationseditor, sucht nach allen definierten Servern im IBM Security Directory Integrator-Serverprojekt, filtert diejenigen Server heraus, die nicht dasselbe Installationsverzeichnis wie der Konfigurationseditor

verwenden, und versucht, den Stoppvorgang durchzuführen. Anschließend beendet der Konfigurationseditor die Java-VM mit dem Exit-Code 0. Es gibt keine Garantie, dass die Server, die der Konfigurationseditor zu stoppen versuchte, auch tatsächlich gestoppt wurden. Einige Server bleiben auch über den Zeitraum hinaus bestehen, den der Konfigurationseditor für die Ausführung dieses Befehls benötigt, und einige Server verweigern möglicherweise einfach aus verschiedenen Gründen den Stoppvorgang.

Option für Perspektive

Dies ist eine Befehlszeilenoption, die das Öffnen des Konfigurationseditors in einer alternativen Perspektive anfordert. Gegenwärtig wird neben der Standardperspektive lediglich die Easy-ETL-Perspektive unterstützt. Der Konfigurationseditor wird bei Verwendung der folgenden Option in dieser Perspektive gestartet:

```
ibmditk -perspective com.ibm.tdi.rcp.perspective.etl
```

Server

Nachstehend erhalten Sie Informationen zum Arbeiten mit dem Server.

Die folgenden Befehlszeilenoptionen können für den IBM Security Directory Integrator-Server verwendet werden (ibmdisrv [optionen]).

Beispiel:

```
ibmdisrv -c"C:\demos\rs.xml" -r"Access2LDAP" -l"c:\metamerge\mydemo.log"
```

Anmerkung:

1. Zwischen dem Optionsbuchstaben und dem Wert wird kein Leerzeichen angegeben. Verwenden Sie doppelte Anführungszeichen, wenn Sie mögliche Leerzeichen oder Kommas in den Werten beibehalten wollen.
2. Die Ausführungsbedingungen für die Windows-Shell lassen maximal neun (9) Argumente aus der nachfolgenden Liste zu. Bei anderen Plattformen gibt es keine Einschränkungen.
3. Verwenden Sie im Namen der Konfigurationsdatei kein Komma (,).

-s <verzeichnis>

Gibt das Arbeitsverzeichnis an, in dem sich die Lösung befindet. Dieses Verzeichnis wird als Lösungsverzeichnis bezeichnet. Alle relativen Dateireferenzen in IBM Security Directory Integrator, in Ihren Konfigurationen usw. werden auf diese Position bezogen. Dieser Parameter muss als Erster angegeben werden.

Falls das angegebene Verzeichnis nicht vorhanden ist, wird es durch den IBM Security Directory Integrator-Server erstellt und mit einer Reihe von Eigenschaftendateien gefüllt, die auf den Dateien im Installationsverzeichnis basieren und die Sie an Ihre Anforderungen anpassen können. Weitere Informationen finden Sie im Abschnitt Anhang A, „Beispiele für Eigenschaftendateien“, auf Seite 383.

-c <datei...>

Gibt eine oder mehrere Konfigurationsdateien an. Wenn Sie diese Option nicht angeben, werden die Elemente im Ordner "Autostart" geladen und gestartet (sofern nicht durch die Angabe von **-D** unterdrückt). Platzhalter (wie z. B. in *.xml) sind ebenfalls zulässig.

Anmerkung: Das Übergeben mehrerer Konfigurationsdateien ist nur dann zulässig, wenn die Option "-d" ebenfalls angegeben ist.

-n <codierung>

Gibt die Codierung an, die zum Schreiben von Konfigurationsdateien verwendet wird. Es muss sich um eine in Java2 gültige Zeichensatz-ID handeln. Eine vollständige Liste der Werte enthält die IANA-Zeichensatzregistry (<http://www.iana.org/assignments/character-sets>). Bitte beachten Sie, dass Java2 lediglich eine Teilgruppe dieser Zeichensätze unterstützt.

-r <fertigungslinie...>

Gibt eine Liste mit Namen von Fertigungslinien an, die gestartet werden sollen. Um die Fertigungslinien **a** und **b** zu starten, verwenden Sie den Befehl **-r a b**. Andere Syntaxformate wie **-ra,b** und **-ra -rb** werden ebenfalls unterstützt.

Anmerkung: Falls Sie Include-Dateien und Namensbereiche verwenden, kann die Fertigungslinie "myNamespace:/AssemblyLines/alName" sein (unter der Voraussetzung, dass der Namensbereich **myNamespace** und die Fertigungslinie **alName** heißt).

-T<name>

Aktiviert die JLOG-Traceerstellung in der Datei trace<name>.log unter dem Verzeichnis <tivoli_common_directory>/TDI/logs/. Standardmäßig wird der Trace im Hauptspeicher abgelegt (dort kann er im Fall einer nicht behandelten Ausnahmebedingung durch die Traceback-Routinen von JFF-DC abgerufen werden).

-D Dieser Parameter inaktiviert den Start von Elementen im Ordner "Auto-start".

-w Falls die Option "-r" (oder "-t") angegeben ist, bewirkt dieser Parameter, dass IBM Security Directory Integrator bei jeder Fertigungslinie den Abschluss abwartet, bevor die nächste Fertigungslinie gestartet wird. Wenn dieser Parameter nicht angegeben ist, startet IBM Security Directory Integrator alle durch den Parameter "-r" angegebenen Fertigungslinien parallel. Nach der Beendigung der letzten Fertigungslinie wird der Server gestoppt.

-e Bei Angabe dieser Option wird der Server im gesicherten Modus ausgeführt. Alle Konfigurationsdateien sowie die Server-API-Registry werden unter Verwendung des speziellen Hauptkennworts dieses Servers entschlüsselt und verschlüsselt.

-v Zeigt Versionsinformationen an und führt dann eine Beendigung durch. Dies wird nur in der Protokolldatei protokolliert.

-P <kennwort>

Gibt das Kennwort an, falls die Konfigurationsdatei(en) verschlüsselt ist/sind.

-p Erstellt beim Start einen Speicherauszug der Java-Eigenschaften. Bitte beachten Sie, dass Sie dennoch eine Konfigurationsdatei angeben müssen, die gelesen wird, bevor der Speicherauszug der Java-Eigenschaften erstellt wird.

-d Startet einen "Dämon" oder eine *Konfigurationsinstanz* auf diesem System.

Wenn Sie den Start mit der Option "-d" ausführen, wird 1 anonyme Instanz (der Dämon) gestartet, die für jede in der Befehlszeile angegebene Konfigurationsdatei 1 Konfigurationsinstanz startet. Auf diese Weise können Sie in einem Arbeitsgang mehrere Konfigurationsinstanzen starten. In der Befehlszeile können Sie 0 oder mehr Konfigurationsdateien angeben. Die Angabe von auszuführenden Fertigungslinien in diesem Modus ist nicht sinnvoll, da nicht angegeben werden kann, in welcher Konfigurationsdatei sich

die Fertigungslinie befindet. Sie können Fertigungslinien jedoch automatisch starten, da diese zu der Konfigurationsinstanz gehören, die den automatischen Start angibt.

Falls Sie den Start ohne die Option "-d" ausführen, erhalten Sie 1 Konfigurationsinstanz, die die in der Befehlszeile angegebene Konfigurationsdatei lädt. Sie müssen in der Befehlszeile genau 1 Konfigurationsdatei angeben. (Falls Sie mehrere Konfigurationsdateien verwenden müssen, können diese bei der Standardeingabe über eine Befehlsverkettung angegeben werden.) In diesem Modus können Sie beliebig viele auszuführende Fertigungslinien angeben. Dies ist das traditionelle Verfahren für die Ausführung des Servers.

-q Verwendet 1 Argument für den Modus. Der Modus 1 bedeutet eine Ausführung im Aufzeichnungsmodus, der Modus 2 bedeutet eine Ausführung im Wiedergabemodus.

-l <datei>

Gibt die Protokolldatei an (Standardkonsolenausgabe). Diese Option hat nur geringe Wirkung, da wenige Nachrichten in der Konsole ausgegeben werden. Um die Protokolldatei für den Großteil der Protokollierung zu ändern, ändern Sie die Datei `log4j.properties`.

-R Inaktiviert die ferne API ungeachtet der Einstellung in der Datei `global.properties`.

-W Startet alle Konfigurationen in demselben Thread. Konfigurationen werden nicht beendet, sondern sind für unbegrenzte Zeit im Wartestatus.

-M Startet Fertigungslinien im Simulationsmodus.

-S Diese Option ist nur zur internen Verwendung für die Kommunikation zwischen dem Konfigurationseditor und einem Server bestimmt. Sie wird verwendet, um Konfigurationsdateien zwischen dem Konfigurationseditor und dem Server zu übergeben. Verwenden Sie diese Option bei Ihren eigenen Befehlsangaben nicht.

-f ext_eig1=datei1, ext_eig2=datei2

Hierbei ist `ext_eig` der Name des externen Eigenschaftsspeichers. `datei` gibt an, aus welcher Datei die Eigenschaften gelesen werden. Diese Option gibt einen benutzerdefinierten externen Eigenschaftsspeicher an, der beim Starten eines IBM Security Directory Integrator-Servers verwendet werden kann. Dieser optionale Befehlszeilenparameter `-f` kann mit den Serverstartskripts "ibmdisrv" verwendet werden. `ext_eig` ist der Name des externen Eigenschaftsspeichers. `datei` gibt an, aus welcher Datei die Eigenschaften gelesen werden. Wenn die Option `-f` verwendet wird, um über die Befehlszeile eine Eigenschaftendatei anzugeben, ändert der Server die Konfiguration des Eigenschaftsspeichers nur im Hauptspeicher, nimmt also diese Änderung nicht permanent durch eine Änderung der auf Platte gespeicherten IBM Security Directory Integrator-Konfigurationsdatei vor. Diese Änderung ist nur für die aktuelle Ausführung des IBM Security Directory Integrator-Servers gültig.

Falls in der Befehlszeile Eigenschaftendateien angegeben werden, sind sie nur für die Konfigurationsinstanzen gültig, die mit der Befehlszeilenoption `-c` angegeben wurden (diese werden beim Start des IBM Security Directory Integrator-Servers geladen). Auf Konfigurationsinstanzen, die nicht explizit mit der Befehlszeilenoption `-c` benannt werden (also beispielsweise durch den Client der fernen Server-API geladene Konfigurationsinstanzen) haben die in der Befehlszeile angegebenen Eigenschaftendateien keinen Einfluss.

Falls ein Eigenschaftsspeicher, dessen Name mit dem Befehlszeilenswitch `-f` angegeben ist, in einer Konfigurationsinstanz nicht gefunden wird, wird im Serverprotokoll (Datei "ibmdi.log" im Installationsverzeichnis) eine Fehlermeldung protokolliert. Wird der Name eines Eigenschaftsspeichers mehrfach mit dem Befehlszeilenswitch `-f` angegeben, führt dies dazu, dass zum einen eine Warnung protokolliert wird und zum zweiten die zuletzt angegebene Datei wirksam ist. Diese Funktionsweise ist in der Java-Klasse "com.ibm.di.server.RS" implementiert (bei der Scripterstellung wird dies durch die Variable "main" referenziert). Nach dem Aufruf der Methode "reload()" wird das Objekt "MetamergeConfig" geladen und für jeden in der Befehlszeile angegebenen Eigenschaftsspeicher wird das entsprechende Objekt "PropertyStoreConfig" aktualisiert.

Anmerkung:

Das Kopieren/Einfügen von Konfigurationsobjekten (Fertigungslinien, Connectors, Funktionskomponenten usw.) wird jedoch vollständig unterstützt. Sie können ohne Weiteres Fertigungslinien und Komponenten kopieren und anschließend in eine andere Konfiguration einfügen. Auch der Austausch von Fertigungslinien und Komponenten mittels IM-Chats, E-Mails und Textdateien ist möglich, da der Kopierpuffer mit der XML-Definition der IBM Security Directory Integrator-Konfiguration für das ausgewählte Element gefüllt wird. Die auf diese Weise ermöglichte einfache und komfortable Übergabe von Elementen ist ein wertvolles Werkzeug für die Unterstützung und Onlinehilfe (z. B. ICT/NotesBuddy, Foren etc.).

Anmerkung:

Achten Sie darauf, den gesamten Knoten `<MetamergeConfig>` - inklusive Start- und Endtag - in Ihrem Kopierbefehl auszuwählen.

- i Diese Option gibt an, dass der IBM Security Directory Integrator-Server alle Eigenschaften aus der Datei `global.properties` ignoriert und ausschließlich die Datei `solution.properties` liest. Sie kann verwendet werden, wenn die Datei `global.properties` nicht lesbar ist, beispielsweise dann, wenn die Codierung, mit der der IBM Security Directory Integrator-Server gestartet wird, von der Codierung der Datei `global.properties` abweicht.
- ? Gibt eine Nachricht mit einem *Verwendungshinweis* aus, der zu allen Optionen Kurzinformationen enthält.
- j <datei>
Diese Option wird zum Lesen von Regressionsinformationen aus der angegebenen Datei verwendet und wird mit den Details aus der Fertigungslinie verglichen. Werden Abweichungen festgestellt, werden Warnungen in die Protokolldatei geschrieben. Diese Option ist nur bei der Ausführung einer einzigen Fertigungslinie hilfreich.
- J<datei>
Diese Option wird zum Schreiben der Regressionsinformationen der Fertigungslinie in die angegebene Datei verwendet.
- k<datei>
Diese Option wird zum Ignorieren des Work-Eintrags beim Lesen der Regressionsinformationen verwendet.

Anmerkung:

1. Wenn eine Eigenschaft als Parameter für einen Connector, einen Parser oder eine Funktionskomponente verwendet wird und diese Eigenschaft nicht im Eigenschaftsspeicher vorhanden ist, wird eine Warnung protokolliert.
2. Sie können eine Konfigurationsdatei in den Server laden, ohne die Fertigungslinien starten zu müssen. Warnungen werden für die nicht vorhandenen Eigenschaften, die verwendet werden, protokolliert.

Wenn IBM Security Directory Integrator beendet wird, wird einer der folgenden Exit-Codes zurückgegeben:

- 0 Kein Fehler. Die Operation wurde erfolgreich ausgeführt.
- 1
- Die Protokolldatei (Parameter "-l") kann nicht geöffnet werden.
 - Die Konfigurationsdatei kann nicht geöffnet werden.
 - Eine Fertigungslinie ist fehlgeschlagen. Dies gilt nur, wenn der Server nicht im Dämonmodus (also ohne die Option "-d") ausgeführt wird. Beispiel: `ibmdirsv -c rs.xml -r a1` oder `ibmdirsv -c rs.xml -r a11,a12,a13`.
- 2 (Obsolet) Nach der automatischen Ausführung erfolgt eine Beendigung. Wenn Sie IBM Security Directory Integrator unter Angabe der Option "-w" starten, führt der Server die im Parameter "-r" angegebenen Fertigungslinien aus und wird anschließend beendet.
- Anmerkung:** Fertigungslinien, die über den Konfigurationseditor ausgeführt werden, werden anders gestartet und nicht mit dem Status 2 beendet.
- 9 (Obsolet) Die Lizenz ist abgelaufen oder ungültig.

Anmerkung: Falls der Server durch eine Verwaltungsanforderung beendet wird und ein angepasster Exitcode angegeben ist, wird dieser angepasste Code als Exit-Code des Servers verwendet.

CLI - Dienstprogramm "tdisrvctl"

Die IBM Security Directory Integrator-CLI (Command Line Interface - Befehlszeilenschnittstelle), die auch als "Dienstprogramm *tdisrvctl*" bezeichnet wird, ist für die ferne Verwaltung von Konfigurationen, Fertigungslinien usw. konzipiert.

Dieses Dienstprogramm stellt unter Verwendung der fernen Server-API eine Verbindung zu einem fernen IBM Security Directory Integrator-Server her und führt die angeforderten Operationen aus. Da es sich um eine Clientanwendung handelt, die eine Schnittstelle mit einem fernen Server bildet, unterliegt sie denselben Verbindungs-, Authentifizierungs- und Autorisierungsaspekten, die in Kapitel 6, „Sicherheit“, auf Seite 103 beschrieben sind.

Sie bietet zahlreiche Befehlszeilenoptionen für die folgenden Funktionen:

- IBM Security Directory Integrator-Konfigurationen starten, stoppen oder erneut laden
- Fertigungslinien in einer bestimmten Konfiguration starten oder stoppen
- Liste der im Server geladenen Konfigurationen anzeigen
- Server beenden
- Konfigurationsbericht anzeigen

- Konfigurationseigenschaften über TDI-p (Eigenschaftenframework von IBM Security Directory Integrator) verwalten
- Angepasste Benachrichtigungsereignisse senden
- Verfügbare Fertigungslinienoperationen anzeigen
- Tombstones für beendete Konfigurationen und Fertigungslinien anzeigen
- Details des IBM Security Directory Integrator-Servers anzeigen

Anmerkung:

1. Das Befehlszeilendienstprogramm wird im Ordner *tdi-installationsverzeichnis/bin* ausgeliefert.
2. RMI (Remote Method Invocation) ist in der Datei "global.properties" standardmäßig (`api.remote.on=true`) mit aktivierter Sicherheit (`api.remote.ssl.on=true`) aktiviert. Wenn `api.remote.ssl.on=true` angegeben ist, müssen die allgemeinen Optionen für den Schlüsselspeicher und den Truststore (`general_options`) in den Befehl eingeschlossen werden. Beispiel:

```
tdisrvctl.bat -h mytdiserver.com -p 1099 ..\serverapi\testadmin.jks -P administrator -T ..\testserver.jks -W server -op srvinfo
```

Wenn die RMI-Sicherheit inaktiviert ist (`api.remote.ssl.on=false`), muss die IP-Adresse des Clients, der das Dienstprogramm "tdisrvctl" ausführt, in der Eigenschaft `api.remote.nonssl.hosts` definiert sein. In diesem Fall kann der Befehl "tdisrvctl" von dem angegebenen Client ohne die Parameter für Schlüsselspeicher und Truststore ausgeführt werden. Beispiel:

```
tdisrvctl.bat -h mytdiserver.com -p 1099 -op srvinfo
```

3. Der ferne IBM Security Directory Integrator-Server muss aktiv sein.

Befehlszeilenreferenz

Nachstehend erhalten Sie Informationen zur Verwendung der Befehlszeilenreferenz.

Der Befehl hat die folgende Syntax:

```
tdisrvctl [allgemeine_optionen] -op operation [operationsspezifische_optionen]
```

Hierbei kann für **allgemeine_optionen** Folgendes angegeben werden:

-h host	Geben Sie die IP-Adresse oder den Hostnamen des fernen Servers ein (Standardwert ist "localhost").
-K schlüsselspeicher	Geben Sie den Namen der Datenbankdatei mit dem SSL-Schlüssel ein.
-p port	Geben Sie die Portnummer ein (Standardwert ist 1099).
-P schlüsselkennwort	Geben Sie das Kennwort der Schlüsseldatei ein.
-s	Geben Sie das Arbeitsverzeichnis an, in dem sich das Lösungsverzeichnis befindet.
-T truststore	Geben Sie den Namen der Datenbankdatei mit dem SSL-Truststore ein.
-u benutzer-id	Geben Sie den Benutzernamen (für die angepasste Authentifizierung) ein.
-v	Diese Option bewirkt eine Ausführung im ausführlichen Modus.
-w benutzerkennwort	Geben Sie das Benutzerkennwort ein (für die angepasste Authentifizierung).
-W trust-datei-kennwort	Geben Sie das Kennwort für die Trust-Datei ein.
-?	Bei Verwendung dieser Option wird die Befehlssyntax angezeigt.

Für **operation** kann Folgendes verwendet werden:

event	Angepasste Benachrichtigungsereignisse senden.
prop	Konfigurationseigenschaften verwalten.
queryop	Fertigungslinienoperationen abfragen.
reload	Aktive Konfigurationen erneut laden.
report	Konfigurationsbericht generieren oder Liste der Konfigurationen auf dem fernen Server erstellen.
shutdown	Server beenden.
srvinfo	IBM Security Directory Integrator-Serverinformationen anzeigen.
status	Status von Konfigurationen oder Fertigungslinien anzeigen.
start	Bestimmte Konfiguration oder Fertigungslinien starten.
stop	Bestimmte Konfiguration oder Fertigungslinien stoppen.
tombstone	Tombstonedatensätze für bestimmte Konfiguration oder Fertigungslinie anzeigen.
deletetombstone	Tombstonedatensatz löschen.
debug	Debug für Komponenten einer aktiven Fertigungslinie ausführen.

Für jede einzelne Option können Sie mit einem Befehl wie dem Folgenden den Hilfetext anzeigen:

```
tdisrvctl -op operation -?
```

Operationen

Nachstehend ist eine Liste mit den Operationen aufgeführt.

event Verwenden Sie diese Option, um angepasste Benachrichtigungsereignisse an einen bestimmten Server zu senden. Alle für das bestimmte Ereignis registrierten Listener empfangen diese Benachrichtigung. Auf diese Weise können IBM Security Directory Integrator-Administratoren Listeneranwendungen auf der Grundlage von geplanten angepassten Ereignissen auslösen.

Die Operation "event" hat die folgende Syntax:

```
qdisrvctl [allgemeine_optionen] -op event -e ereignisname [-s quelle ] [-d daten]
```

Hierbei gilt Folgendes:

-e ereignisname	Der Name des zu sendenden Ereignisses.
-s quelle	Der Name der Quelle, die das Ereignis aufruft (Standardwert ist "tdisrvctl").
-d daten	Die an einen Ereignislistener zu übergebenden Daten (Standardwert ist "null").

Beispiel:

Der folgende Befehl sendet ein Ereignis "user.process.X.completed" von "admin".

```
tdisrvctl -h itditest -op event -e "process.X.completed" -s admin -d "Durch Administrator ausgelöstes Ereignis"
```

Anmerkung: Alle durch das Dienstprogramm "tdisrvctl" mit der Option "-e" gesendeten Ereignisse werden mit dem Präfix "user" versehen.

prop Die Option "prop" macht die Eigenschaften einer Konfiguration über TDI-p zugänglich. Mit ihrer Hilfe kann ein Benutzer die Eigenschaften einer bestimmten Konfiguration abrufen, festlegen und anzeigen.

Die Operation "prop" hat die folgende Syntax:


```

tdisrvctl [allgemeine_optionen] -op prop -c konfigurationsname
[ -l ] |
[-o eigenschaftsspeicher]
[-g schlüssel | all] |
[-s schlüssel=wert] [-e] |
[-d schlüssel] ]

```

Hierbei gilt Folgendes:

-c konfigurationsname	Der Name der Konfiguration, mit der gearbeitet werden soll.
-l	Erstellt eine Liste aller konfigurierten Eigenschaftsspeicher.
-o eigenschaftsspeicher	Der Name des Eigenschaftsspeichers, mit dem gearbeitet werden soll.
-g schlüssel	Ruft den Wert des angegebenen Schlüssels ab (das Schlüsselwort "all" führt zum Abruf aller Schlüssel).
-s schlüssel=wert	Legt den "schlüssel" mit dem angegebenen "wert" fest.
-e	Verschlüsselt den Wert beim Ablegen im Speicher (diese Option kann nur zusammen mit der Option "-s" verwendet werden).
-d schlüssel	Löscht den angegebenen "schlüssel" aus dem Speicher.

Anmerkung:

1. Die Optionen "-l", "-g", "-s" und "-d" schließen sich gegenseitig aus und können nicht kombiniert werden.
2. Die Option "-e" kann nur zusammen mit der Option "-s" verwendet werden.
3. Die Verwaltung von Eigenschaften, die im **Kennwortspeicher** gespeichert sind, wird NICHT unterstützt.
4. Bei Angabe der Option "-c" muss der VOLLSTÄNDIGE Konfigurationsdateipfad auf dem fernen Server oder aber ein relativer Pfad angegeben werden, der sich auf den Ordner "configs" bezieht. Zum Anzeigen der relativen Pfade verwenden Sie die Option "report" des Dienstprogramms "tdisrvctl":

```
tdisrvctl -op report -l
```

Beispiele:

Der folgende Befehl zeigt eine Liste aller Eigenschaftsspeicher für die Konfiguration "C1.xml" an:

```
tdisrvctl -op prop -c C1.xml -l
```

Der folgende Befehl ruft eine Liste aller Eigenschaften für die Konfiguration "C1.xml" ab:

```
tdisrvctl -op prop -c C1.xml -g all
```

Der folgende Befehl ruft eine Liste aller Eigenschaften für die Konfiguration "C1.xml" aus dem Speicher "MyStore" ab:

```
tdisrvctl -op prop -c C1.xml -o MyStore -g all
```

Der folgende Befehl legt eine Eigenschaft "MY_PROP" mit dem Wert "MY_VALUE" für die Konfiguration "C1.xml" im Speicher "MyStore" fest und kennzeichnet sie als geschützte Eigenschaft:

```
tdisrvctl -op prop -c C1.xml -o MyStore -s MY_PROP=MY_VALUE -e
```

queryop

Die Option "queryop" gibt die Liste der Fertigungslinienoperationen zurück, die in einer Fertigungslinie zugänglich sind.

Diese Option ist in einer Umgebung für die Scripterstellung hilfreich. Ein Entwickler von IBM Security Directory Integrator-Lösungen kann ein Script schreiben, mit dem zugängliche Operationen automatisch abgefragt werden, und die Ergebnisse anschließend zum Starten einer Fertigungslinie mit einer bestimmten Operation unter Verwendung des Parameters **-r -alop** der Startoperation verwenden. Die Ausgabe dieser Operation kann in einer scriptgesteuerten Umgebung ohne großen Aufwand durchsucht oder mit einem Token versehen werden.

Die Operation **queryop** hat die folgende Syntax:

```
tdisrvctl [allgemeine_optionen] -op queryop -c <konfigurationsdatei> -r <fertigungsliniennamen>
```

Hierbei gilt Folgendes:

konfigurationsdatei	Der Name der Konfigurationsdatei.
fertigungsliniennamen	Der Name der Fertigungslinie.

Ausgabe:

```
ALOp:{attr_1;attr_2...attr_n;}
```

Anmerkung: Bei Angabe der Option "-c" muss der VOLLSTÄNDIGE Konfigurationsdateipfad auf dem fernen Server oder aber ein relativer Pfad angegeben werden, der sich auf den Ordner "configs" bezieht. Zum Anzeigen der relativen Pfade verwenden Sie die Option "report" des Dienstprogramms "tdisrvctl":

```
tdisrvctl -op report -l
```

Beispiele:

Der folgende Befehl ruft die in einer Fertigungslinie zugänglichen *Operationen* ab:

```
tdisrvctl -h itditest -T trust.kdb  
-W secret -op queryop  
-c examples/ADCustomConnector.xml  
-r ADAssemblyLine
```

Ausgabe für das Beispiel:

```
$initialize: {ldapurl;loginPasswd;loginUserName}
```

reload Diese Option kann verwendet werden, um Konfigurationen erneut zu laden, die auf einem bestimmten Server aktiv sind.

Die Operation "reload" hat die folgende Syntax:

```
tdisrvctl [allgemeine_optionen] -op reload -c [konfigurationsliste]
```

Hierbei gilt Folgendes:

konfigurationsliste	Eine durch Kommas getrennte Liste der Konfigurationen, die erneut geladen werden sollen.
----------------------------	--

Anmerkung: Bei Angabe der Option "-c" muss der VOLLSTÄNDIGE Konfigurationsdateipfad auf dem fernen Server oder aber ein relativer Pfad angegeben werden, der sich auf den Ordner "configs" bezieht. Zum Anzeigen der relativen Pfade verwenden Sie die Option "report" des Dienstprogramms "tdisrvctl":

```
tdisrvctl -op report -l
```

Beispiel:

Der folgende Befehl lädt die Konfigurationen "C1.xml", "C2.xml" und "C3.xml" auf dem fernen Host "itditest" erneut:

```
tdisrvctl -h itditest -T trust.jks -W secret -op reload -c C1.xml,C2.xml,C3.xml
```

report Mit dieser Option kann ein Bericht für eine bestimmte Konfiguration generiert oder eine Liste der Konfigurationen erstellt werden, die im Ordner "config" des fernen Servers verfügbar sind.

Der Konfigurationsbericht listet Details der jeweiligen Konfiguration auf. Hierzu gehören Fertigungslinien, Connectors und Parser in den einzelnen Fertigungslinien, Connectorbibliothek, Parserbibliothek, Scriptbibliothek, Funktionsbibliothek. Bei Verwendung dieser Option erhalten Sie auf einen Blick alle Details einer bestimmten Konfiguration.

Mithilfe der Option für die Konfigurationsliste kann der Benutzer die Liste der auf dem fernen Server verfügbaren Konfigurationen und deren genaue Namen ermitteln. Natürlich werden nur Konfigurationen angezeigt, die sich im Ordner "config" des fernen Servers befinden (Informationen zur Eigenschaft **api.config.folder** finden Sie in der Datei `global.properties`). Eine Liste der Konfigurationen, die sich an anderen Stellen auf dem System befinden, kann mit diesem Befehl nicht abgerufen werden.

Die Operation "report" hat die folgende Syntax:

```
tdisrvctl [allgemeine_optionen] -op report [-c konfiguration | -l]
```

Hierbei gilt Folgendes:

-c konfiguration	Der Name der Konfiguration, für die ein Bericht generiert werden soll.
-l	Erstellt eine Liste der Konfigurationen im Ordner "config" des fernen Servers.

Für jeden Connector oder jede Funktionskomponente, der/die Teil einer Fertigungslinie ist, werden die folgenden Details angezeigt:

```
Name          : count
Modus          : Iterator
Status        : Aktiviert
Debug         : Inaktiviert
Schablone     : system:/Connectors/ibmdi.Timer
Parser        : [parent]
Kommentar     : Kein
```

Anmerkung:

1. Die angegebene Konfiguration muss bereits auf dem fernen Server geladen sein.
2. Sie können entweder die Option "-c" oder die Option "-l" angeben. Die gleichzeitige Verwendung beider Optionen ist nicht zulässig.
3. Bei Angabe der Option "-c" muss der VOLLSTÄNDIGE Konfigurationsdateipfad auf dem fernen Server oder aber ein relativer Pfad angegeben werden, der sich auf den Ordner "configs" bezieht.
4. Beim Argument für die Option "-c" muss die Groß-/Kleinschreibung beachtet werden. Das Argument muss exakt mit dem Namen der Konfigurationsdatei übereinstimmen, der für die Serverinstanz bekannt ist und der beispielsweise durch den Befehl "tdisrvctl -op status" gemeldet wird.

Beispiele:

Der folgende Befehl ruft eine Liste der Details für die Konfiguration "C1.xml" auf dem fernen Server ab:

```
tdisrvctl -h ferner_server -op report -c C1.xml
```

Der folgende Befehl ruft eine Liste der im Ordner "config" auf dem fernen Server verfügbaren Konfigurationen ab:

```
tdisrvctl -h ferner_server -op report -l
```

shutdown

Mit dieser Option kann der IBM Security Directory Integrator-Server beendet werden.

Der Befehl hat das folgende Format:

```
tdisrvctl [allgemeine_optionen] -op shutdown [-o rückkehrcode] [-f]
```

Hierbei gilt Folgendes:

- o rückkehrcode Der Rückkehrcode, mit dem der ferne IBM Security Directory Integrator-Server beendet werden soll.
- f Erzwingt ein kontrolliertes Beenden und beendet alle Fertigungslinien.

Beispiele:

Der folgende Befehl beendet den lokalen IBM Security Directory Integrator-Server:

```
tdisrvctl -op shutdown
```

Der folgende Befehl beendet den lokalen IBM Security Directory Integrator-Server mit einem kontrollierten Beenden aller Fertigungslinien:

```
tdisrvctl -op shutdown -f
```

Der folgende Befehl beendet den auf dem fernen Host "itditest" ausgeführten Server, der für SSL (nur Serverauthentifizierung) konfiguriert ist:

```
tdisrvctl -h itditest -T trust.kdb -W secret -op shutdown
```

srvinfos

Mit dieser Option werden die Informationen zu einem IBM Security Directory Integrator-Server angezeigt.

Der Befehl hat die folgende Syntax:

```
tdisrvctl [allgemeine_optionen] -op srvinfos
```

Beispiel:

Der folgende Befehl zeigt die Serverinformationen für einen auf "localhost" ausgeführten IBM Security Directory Integrator-Server an:

```
tdisrvctl -h localhost -op srvinfos
```

status Mit dieser Option kann der Status von Fertigungslinien angezeigt werden.

Die Operation "status" hat die folgende Syntax:

```
tdisrvctl [allgemeine_optionen] -op status -c [konfigurationsliste | all]
-r [fertigungslinienliste | all]
-listen
```

Hierbei gilt Folgendes:

- konfigurationsliste** Eine durch Kommas getrennte Liste der Konfigurationen oder das Schlüsselwort "all".
- fertigungslinienliste** Eine durch Kommas getrennte Liste der Fertigungslinien oder das Schlüsselwort "all".
- listen** Gibt an, dass mit dem Empfang der Protokolle einer aktiven Konfiguration oder Fertigungslinie begonnen werden soll.

Anmerkung:

1. Es muss mindestens eine der Optionen "-c" oder "-r" angegeben werden.
2. Das Schlüsselwort "all" gibt an, dass der Befehl alle Konfigurationen oder Fertigungslinien berücksichtigen soll.
3. Für die Option "-listen" muss genau 1 Konfiguration oder Fertigungslinie angegeben werden.
4. Bei Angabe der Option "-c" muss der VOLLSTÄNDIGE Konfigurationsdateipfad auf dem fernen Server oder aber ein relativer Pfad angegeben werden, der sich auf den Ordner "configs" bezieht. Zum Anzeigen der relativen Pfade verwenden Sie die Option "report" des Dienstprogramms "tdisrvctl":

```
tdisrvctl -op report -l
```

Beispiele:

Der folgende Befehl zeigt den Status aller Konfigurationen und Fertigungslinien an:

```
tdisrvctl [allgemeine_optionen] -op status -c all -r all
```

Zu diesem Zweck könnte auch der folgende Befehl eingegeben werden:

```
tdisrvctl [allgemeine_optionen] -op status
```

Der folgende Befehl zeigt den Status der Fertigungslinien "AL1" und "AL2" an:

```
tdisrvctl -h itditest -op status -c c1.xml -r AL1,AL2
```

Ausgabe:

```
(Format: "komponententyp # komponentenname # RUNNING oder STOPPED # statistik"):
1 # AL1 # RUNNING # [get:571] [add:571] [del:3] [requests:2333]....
1 # AL2 # STOPPED #
```

Die Komponententypen lauten:

- 0 = Konfiguration
- 1 = Fertigungslinie

Die Statistik enthält die folgenden Details (nur bei Fertigungslinien gültig):

- Attribut "add": Gesamtzahl der ausgeführten Operationen "add"
- Attribut "mod": Gesamtzahl der ausgeführten Operationen "modify"
- Attribut "del": Gesamtzahl der ausgeführten Operationen "delete"
- Attribut "get": Gesamtzahl der ausgeführten Operationen "getNext" (Iterationen)
- Attribut "requests": Gesamtzahl der akzeptierten Anforderungen, wenn die Fertigungslinie einen Servermodusconnector enthält
- Attribut "callReply": Gesamtzahl der ausgeführten Operationen "callReply"
- Attribut "err": Gesamtzahl der festgestellten Fehler
- Attribut "skip": Gesamtzahl der ausgeführten Operationen "skip"
- Attribut "lookup": Gesamtzahl der ausgeführten Operationen "lookup"
- Attribut "ignore": Gesamtzahl der ausgeführten Operationen "ignore"
- Attribut "reconnect": Gesamtzahl der ausgeführten Operationen "reconnect"

- Attribut "exception": Ausnahmebedingungstext, falls die Komponente mit einer Ausnahmebedingung beendet wurde

Der folgende Befehl zeigt die Details der (aktiven und gestoppten) Konfigurationen auf einem bestimmten Server an:

```
tdisrvctl -h itditest -op status -c all
```

Der folgende Befehl zeigt die Details einer aktiven Fertigungslinie auf einem bestimmten Server an und startet den Empfang ihrer Protokolle:

```
tdisrvctl -h itditest -op status -c rs.xml -r all -listen
```

start Mit dieser Option können eine Konfiguration oder Fertigungslinien gestartet werden.

Die Operation "start" hat die folgende Syntax:

```
tdisrvctl [allgemeine_optionen] -op start -c [konfiguration]
-e [kennwort]
-r [fertigungslinienliste | all] -alop <fertigungslinienoperation>
[erforderliches_attribut_1;
erforderliches_attribut_2; ... erforderliches_attribut_n] | [-f dateiname]
-s [simulationsmodus]
-m [ausführungsname] -o [eigenschaftsspeicher_1=dateiname_1,
eigenschaftsspeicher_2=dateiname_2...]
-t [temporäre_konfigurationsinstanz]
-listen
-sync
```

Hierbei gilt Folgendes:

-c konfiguration	Der Name der Konfiguration, die gestartet werden soll.
-e kennwort	Das Kennwort der Konfigurationsdatei, falls diese verschlüsselt ist.
-r fertigungslinienliste	Eine durch Kommas getrennte Liste der zu startenden Fertigungslinien oder das Schlüsselwort "all".
-o eigenschaftendateiliste	Eine durch Kommas getrennte Liste der Namen und Werte für Eigenschaftsspeicher.
-alop fertigungslinienoperation	Die spezielle Fertigungslinienoperation und eine Liste der erforderlichen Attribute für die angegebene Operation.
-f dateiname	Der Name der Datei, in der die Eingabeattribute und deren Werte für die Operation konfiguriert sind.
-s simulationsmodus	Führt die angegebene Fertigungslinie im Simulationsmodus aus.
-m mehrinstanz	Führt mehrere Instanzen derselben Konfiguration unter verschiedenen Ausführungsnamen aus.
-t temporäre_konfigurationsinstanz	Startet die temporäre Konfigurationsinstanz aus dem XML in der angegebenen Konfigurationsdatei.
-listen	Empfängt die Protokolle der angegebenen Konfiguration oder Fertigungslinie.
-sync	Führt die Fertigungslinie synchron aus.

Anmerkung:

1. Die Angabe der Option "-c" ist verbindlich. -
2. Das Schlüsselwort "all" gibt an, dass der Befehl alle Fertigungslinien berücksichtigen soll.
3. Die Angabe einer Liste der erforderlichen Attribute ist bei der Option "-alop" verbindlich.

4. Die Option "-alop" kann nicht zusammen mit der Option "-r *all*" verwendet werden. Sie kann nur für eine bestimmte Fertigungslinie verwendet werden.
5. Bei Ausführung einer temporären Konfiguration mit dem Lösungs- oder Ausführungsnamen kann nicht geprüft werden, ob auf dem Server bereits eine andere Konfiguration mit diesem Namen aktiv ist, was eine Ausnahmebedingung auslösen würde. Sie können die aktiven Konfigurationsinstanzen mit dem Befehl **status** prüfen.
6. Bei der Verwendung der Option "-t" wird davon ausgegangen, dass sich die mit der Option "-c" angegebene Konfiguration auf der Clientmaschine befindet.
7. Falls die Option "-t" verwendet wird und in der Option "-c" ein relativer Wert für die Konfiguration angegeben ist, wird im aktuellen Ordner nach der Konfiguration gesucht.
8. Für die Option "-listen" muss genau 1 Konfiguration oder Fertigungslinie angegeben werden.
9. Mit der Option "-listen" wird eine Fertigungslinie synchron ausgeführt. Eine Kombination mit der Option "-sync" ist nicht erforderlich.
10. Bei der Verwendung der Option "-sync" muss genau eine Fertigungslinie angegeben werden.

Beispiele:

1. Der folgende Befehl startet die Fertigungslinien "AL1" und "AL2" der Konfiguration "C1" auf dem fernen Server "itditest":

```
tdisrvctl -h itditest -T trust.kdb -W secret -op start -c C1.xml -r AL1,AL2
```

Bei Verwendung der Option "-r" muss die Option "-c" ebenfalls angegeben werden. Dies liegt daran, dass die im Befehl angegebenen Fertigungslinien zu einer der Konfigurationen in der Option "-c" gehören *müssen*.

2. Der folgende Befehl startet die Fertigungslinie "AL1" auf dem fernen Server "itditest" mit der Fertigungslinienoperation:

```
tdisrvctl -h itditest -T trust.kdb -W secret -op start  
-c examples/ADCustomConnector.xml  
-r ADAssemblyLine  
-alop $initialize {ldapurl:ldap://9.182.190.149:390;loginPasswd:password;loginUsrname:cn=root}
```

3. Der folgende Befehl startet die Fertigungslinie "AL1" auf dem fernen Server "itditest" mit der Fertigungslinienoperation "update":

```
tdisrvctl -h itditest -T trust.kdb -W secret -op start  
-c examples/ADCustomConnector.xml -r ADAssemblyLine  
-alop search {$init.ldapurl:ldap://9.182.190.149:390;$init.loginPasswd:password;  
$init.loginUsrname:cn=root;searchBase:o=ibm,c=us}
```

Anmerkung: Alle Initialisierungsattribute müssen mit dem Präfix "\$init" versehen werden.

- 4.

```
tdisrvctl -h itditest -T trust.kdb -W secret -op start -c examples/ADCustomConnector.xml  
-r ADAssemblyLine -alop search -f inputFile
```

Eingabedateiformat:

```
=====
schlüssel1:wert1
schlüssel2:wert2
```

5. Der folgende Befehl führt die Fertigungslinie "AL1" im Simulationsmodus aus:

```
tdisrvctl -h itditest -T trust.kdb -W secret -op start -c examples/ADCustomConnector.xml -r AL1 -s
```

6. Der folgende Befehl lädt mehrere Konfigurationsinstanzen:

```
tdisrvctl -op start -c C1.xml -m test -f PropertyStoreName=TestProp.properties, PropStore2=profile2 ... -r AL1,AL2
```

7. Der folgende Befehl führt eine temporäre Konfigurationsinstanz aus:

```
tdisrvctl -op start -c C1.xml -t -r AL1
```

8. Der folgende Befehl startet eine Konfiguration auf einem bestimmten Server und empfängt ihre Protokolle:

```
tdisrvctl -h itditest -op start -c rs.xml -listen
```

9. Der folgende Befehl startet eine Fertigungslinie auf einem bestimmten Server und empfängt ihre Protokolle:

```
tdisrvctl -h itditest -op start -c rs.xml -r AL1 -listen
```

10. Der folgende Befehl führt eine Fertigungslinie auf einem bestimmten Server synchron aus:

```
tdisrvctl -h itditest -op start -c rs.xml -r AL1 -sync
```

stop Die Operation "stop" hat die folgende Syntax:

```
tdisrvctl [allgemeine_optionen] -op stop -c [konfiguration]  
-r [fertigungslinienliste | all]
```

Hierbei gilt Folgendes:

-c konfiguration	Der Name der Konfiguration.
-r fertigungslinienliste	Eine durch Kommas getrennte Liste der zu stoppenden Fertigungslinien oder das Schlüsselwort "all".
-f	Erzwingt ein kontrolliertes Beenden von Fertigungslinien.

Anmerkung:

1. Die Angabe der Option "-c" ist verbindlich.
2. Bei Angabe der Option "-c" muss der VOLLSTÄNDIGE Konfigurationsdateipfad auf dem fernen Server oder aber ein relativer Pfad angegeben werden, der sich auf den Ordner "configs" bezieht. Zum Anzeigen der relativen Pfade verwenden Sie die Option "report" des Dienstprogramms "tdisrvctl":

```
tdisrvctl -op report -l
```
3. Das Schlüsselwort "all" gibt an, dass der Befehl alle Fertigungslinien berücksichtigen soll.
4. Bei Verwendung der Option "-r" muss die Option "-c" ebenfalls angegeben werden. Dies liegt daran, dass die im Befehl angegebenen Fertigungslinien zu einer der Konfigurationen in der Option "-c" gehören *müssen*.
5. Die Verwendung der Option "-f" ist optional.
6. Beim Argument für die Option "-c" muss die Groß-/Kleinschreibung beachtet werden. Das Argument muss exakt mit dem Namen der Konfigurationsdatei übereinstimmen, der für die Serverinstanz bekannt ist und der beispielsweise durch den Befehl "tdisrvctl -op status" gemeldet wird.

Beispiel:

Der folgende Befehl stoppt die Fertigungslinien "AL1" und "AL2" der Konfiguration "C1" auf dem fernen Server "itditest":

```
tdisrvctl -h itditest -T trust.jks -W secret -op stop -c C1.xml -r AL1,AL2
```

tombstone

Mit dieser Option können Tombstonedetails von zuvor ausgeführten Konfigurationen, Fertigungslinien und EventHandlern (Langzeitdaten) angezeigt werden.

Die Operation "tombstone" hat die folgende Syntax:

```
tdisrvctl [allgemeine_optionen] -op tombstone -c [konfiguration]
[-r [fertigungsliniename] ]
  [-age n]
    [[attributliste] | all ]
```

Hierbei gilt Folgendes:

-age n	Der Tombstonedatensatz soll die vergangenen "n" Tage berücksichtigen (Standardwert ist 1 Tag).
-c konfiguration	Der Name der Konfiguration.
-r fertigungsliniename	Der Name der Fertigungslinie.
all	Es sollen alle Tombstone-Attribute angezeigt werden.

attributliste:

-ct	Komponententyp.
-cn	Komponentenname.
-guid	GUID des Tombstone-Eintrags.
-et	Ereignistyp.
-ex	Exit-Code.
-stime	Startzeit der Komponente.
-ctime	Erstellungszeit des Tombstones.
-desc	Fehlerbeschreibung.
-um	Benutzernachricht.
-stat	Statistik (nur bei Fertigungslinien gültig).

Anmerkung:

1. Die Angabe der Option "-c" ist verbindlich.
2. Bei Angabe der Option "-c" muss der VOLLSTÄNDIGE Konfigurationsdateipfad auf dem fernen Server oder aber ein relativer Pfad angegeben werden, der sich auf den Ordner "configs" bezieht. Zum Anzeigen der relativen Pfade verwenden Sie die Option "report" des Dienstprogramms "tdisrvctl":

```
tdisrvctl -op report -l
```
3. Beim Argument für die Option "-c" muss die Groß-/Kleinschreibung beachtet werden. Das Argument muss exakt mit dem Namen der Konfigurationsdatei übereinstimmen, der für die Serverinstanz bekannt ist und der beispielsweise durch den Befehl "tdisrvctl -op status" gemeldet wird.

Beispiele:

1. Der folgende Befehl zeigt die Tombstone-Einträge der letzten 2 Tage (alle Attribute) für die Konfiguration "C1.xml" an:

```
tdisrvctl [allgemeine_optionen] -op tombstone -c C1.xml -age 2 all
```
2. Der folgende Befehl zeigt die Tombstone-Einträge der letzten 3 Tage für die Konfiguration "C1" an:

```
tdisrvctl -h itdiserver -op tombstone -c C1 -age 3 all
```
3. Der folgende Befehl zeigt die Tombstone-Einträge der letzten 24 Stunden (bestimmte Attribute) für die Konfiguration "C1" an:

```
tdisrvctl -h itdiserver -op tombstone -c C1 -ct -ctime -cn -um
```
4. Der folgende Befehl zeigt den Tombstone-Eintrag für die Fertigungslinie "AL1" von "rs.xml" an:

```
tdisrvctl -h itdiserver -op tombstone -c C1 -r AL1
```

deletetombstone

Mit dieser Option können Tombstone-Einträge für zuvor ausgeführte Fertigungslinien gelöscht werden.

Die Operation "deletetombstone" hat die folgende Syntax:

```
tdisrvctl [allgemeine_optionen] -op deletetombstone -guid <guid-nummer>
```

Hierbei gilt Folgendes:

-guid "guid-nummer" ist die eindeutige Kennung für den zu löschenden Tombstone. Die GUID für einen Tombstone kann durch das Anzeigen des Tombstone-Inhalts abgerufen werden. Details zum Abrufen der GUID finden Sie in den Informationen zur Option tombstone.

debug Mit dieser Option können die Debugmoduswerte von Connectors und Funktionskomponenten einer aktiven Fertigungslinie festgelegt werden. Wenn Sie den Debugmodus eines Connectors mit angegebenem Parser festlegen, wird der Debugmodus des Parsers ebenfalls mit demselben Wert initialisiert.

Die Operation "debug" hat die folgende Syntax:

```
tdisrvctl [allgemeine_optionen] -op debug -c konfiguration  
          -r fertigungslinie  
          [-alc fertigungslinienkomponente]  
          -on/off
```

Hierbei gilt Folgendes:

-c konfiguration	Der Name der Konfiguration.
-r fertigungslinie	Der Name der Fertigungslinie.
-alc fertigungslinienkomponente	Der Name der Fertigungslinienkomponente.
-on	Mit diesem Parameter wird das Debug aktiviert.
-off	Mit diesem Parameter wird das Debug inaktiviert.

Anmerkung:

1. Die Angabe der Optionen "-c" und "-r" ist verbindlich und erfordert die Angabe genau 1 Konfiguration bzw. Fertigungslinie.
2. Bei Angabe der Option "-c" muss der VOLLSTÄNDIGE Konfigurationsdateipfad auf dem fernen Server oder aber ein relativer Pfad angegeben werden, der sich auf den Ordner "configs" bezieht. Zum Anzeigen der relativen Pfade verwenden Sie die Option "report" des Dienstprogramms "tdisrvctl":

```
tdisrvctl -op report -l
```
3. Beim Argument für die Option "-c" muss die Groß-/Kleinschreibung beachtet werden. Das Argument muss exakt mit dem Namen der Konfigurationsdatei übereinstimmen, der für die Serverinstanz bekannt ist und der beispielsweise durch den Befehl "tdisrvctl -op status" gemeldet wird.
4. Wenn die Option **-alc** nicht angegeben wird, sind alle Komponenten der angegebenen Fertigungslinie betroffen.

Beispiele:

1. Der folgende Befehl zeigt den Debugmoduswert der Komponenten in den Fertigungslinien einer angegebenen Konfiguration an:

```
tdisrvctl -op report -c C1
```

2. Der folgende Befehl aktiviert den Debugmodus für alle Komponenten in der aktiven Fertigungslinie "a12":

```
tdisrvctl -op debug -c C1-r a12 -on C1-r a12 -on
```

3. Der folgende Befehl inaktiviert den Debugmodus für angegebene Komponenten in der aktiven Fertigungslinie "a13":

```
tdisrvctl -op debug -c C1-r a13 -a1c comp1,comp2 -off
```

Weitere zu beachtende Punkte

- Falls der Benutzer die Option **-T** oder die Option **-K** angibt, bedeutet dies, dass das Befehlszeilendienstprogramm SSL verwenden muss.
- Wird die Option **-h** (Host) nicht angegeben, sucht die CLI nach der Umgebungsvariablen **TDI_RSRV**. Ist die Variable "TDI_RSRV" leer oder nicht definiert, wird der Standardwert "localhost" verwendet. Dies gilt ebenfalls für die Option **-p** (Port): Falls **-p** nicht angegeben ist, wird nach der Variablen **TDI_RPORT** gesucht. Ist diese Variable nicht definiert, wird der Standardwert "1099" verwendet.
- Der Befehl "tdisrvctl" gibt den Exit-Code 0 zurück, falls die Operation erfolgreich ausgeführt wurde. Schlägt die Operation fehl, wird ein Exit-Code zurückgegeben, der nicht 0 ist. Mögliche Ursachen für das Fehlschlagen einer Operation:
 - Es kann keine Verbindung zum fernen Server aufgebaut werden.
 - Der ferne Server hat einen Fehler gemeldet (wahrscheinlich im Zusammenhang mit der ausgeführten Operation).
 - Eine synchron ausgeführte Fertigungslinie ist fehlgeschlagen (Informationen hierzu finden Sie in den Angaben über die Option "-sync" der Option "start").
- Das Befehlszeilendienstprogramm "tdisrvctl" verwendet Log4J-Protokollierungs-APIs für die Protokollierung von Fehlernachrichten. Die Log4J-Konfigurationsdatei ist im Startscript (Datei ".bat" oder ".sh") angegeben. Der Befehl verwendet eine Datei namens `tdisrvctl-log4j.properties`, um die Log4J-Protokollierung zu konfigurieren. Falls das Lösungsverzeichnis angegeben ist, legt der Befehl eine Umgebungsvariable für den Verweis auf die Protokollkonfigurationsdatei im Lösungsverzeichnis fest. Ist das Lösungsverzeichnis nicht angegeben, verwendet der Befehl die Protokollkonfigurationsdatei aus dem Installationsverzeichnis.
- Die Datei `tdisrvctl-log4j.properties` enthält den vollständigen Pfad der Position, an der die Protokolle erstellt werden sollen. Die Protokolldateien werden standardmäßig im Verzeichnis *tdi-installationsverzeichnis/logs* erstellt. Die Position kann bei Bedarf angepasst werden.
- Alle gemeldeten Fehlernachrichten und Warnungen werden mit einem Fehlercodepräfix angezeigt. Anhand dieses Fehlercodes können Sie in der Veröffentlichung *Nachrichten* nach einer Erläuterung der Fehlernachricht sowie einer Bedieneraktion suchen.

Kapitel 14. Protokollierung und Debug

Nachstehend erhalten Sie Informationen zur Protokollierung und zum Debugging.

IBM Security Directory Integrator verwendet eine Protokollierungsklasse, um Nachrichten in einer Reihe unterschiedlicher Protokollkanäle aufzuzeichnen. Alle IBM Security Directory Integrator-Komponenten verwenden diese Protokollierungsklasse, die ihrerseits ein dem Branchenstandard entsprechendes Protokollierungstool (Log4J) aufruft. Log4J bietet eine Vielzahl von Ausgabekanälen und Formaten. Es gibt allerdings andere Protokolldienstprogramme mit überschneidenden und zusätzlichen Ausgabekanälen, die Sie möglicherweise als IBM Security Directory Integrator-Benutzer benötigen. Hierbei handelt es sich häufig um Open-Source-Bibliotheken, die nicht im Produktpaket von IBM Security Directory Integrator enthalten sind. Damit solche Protokolldienstprogramme von Drittanbietern integriert werden können, ist die Protokollierungskomponente von IBM Security Directory Integrator so modelliert, dass sie als Proxy zwischen IBM Security Directory Integrator und den eigentlichen Protokollierungsimplementierungen, den so genannten Implementierungen von LogInterface, agiert. Weitere Informationen zum Erstellen, Konfigurieren und Programmieren eigener Klassen "LogInterface" enthält der Abschnitt zum Erstellen zusätzlicher Protokollfunktionen in den *Referenzinformationen*.

Anmerkung: Die Protokollierung wird in IBM Security Directory Integrator durch die Konfiguration der Eigenschaft `com.ibm.di.logging.enabled` aktiviert oder inaktiviert. Zum Aktivieren der Protokollierung verwenden Sie die Einstellung `com.ibm.di.logging.enabled=true` (Standardeinstellung). Um die Protokollierung vollständig zu inaktivieren, verwenden Sie `com.ibm.di.logging.enabled=false`.

Im weiteren Verlauf dieses Abschnitts wird erläutert, wie Sie die im Produktpaket von IBM Security Directory Integrator enthaltene Protokollierungsklasse `com.ibm.di.log.TDILog4J` verwenden.

Die Protokollierung und das Debug durch das System erfolgt größtenteils über das Taskobjekt (die aktuelle Fertigungslinie). Die Protokollierung kann entweder explizit (im Script) erfolgen oder durch die zahlreichen Komponenten selbst vorgenommen werden.

Die Log4J-Protokollierungssteuerkomponente ist ein sehr flexibles Framework, mit dem Sie eine Protokollierung in Dateien, im Ereignisprotokoll (eventlog) und im Systemprotokoll (syslog) vornehmen können. Die Protokollierung kann optimiert und so an die meisten Anforderungen angepasst werden. Von großem Nutzen kann sie bei der Fehlerbehebung oder beim Debug für eine Lösung sein. Dank der vorgenannten Protokollierungsklasse verfügt IBM Security Directory Integrator über zusätzliche Tracefunktionen (siehe Kapitel 15, „Traceerstellung und First-Failure Data Capture“, auf Seite 263). In den meisten Fällen ist jedoch die hier beschriebene Protokollierungsfunktionalität ausreichend.

Für IBM Security Directory Integrator-Komponenten gibt es spezielle Fehlerbehebungsrichtlinien. Prüfen Sie stets, ob die Abschnitte zu *Referenz* und *Fehlerbehebung und Unterstützung* im IBM Knowledge Center for IBM Security Directory Integrator weitere Informationen zu der jeweiligen Komponente enthalten.

Das Protokollschema für den Server (ibmdisrv) wird durch die Datei `Log4J.properties` und Elemente der Konfigurationsdatei beschrieben (siehe „Log4J-Standardparameter“ auf Seite 260).

Anmerkung: Jede der vorgenannten Eigenschaftendateien kann sich im Lösungsverzeichnis befinden. In diesem Fall setzen diese Dateien die Werte in der Datei außer Kraft, die sich im Installationsverzeichnis befindet.

Sie können eigene Appender zur Verwendung durch die Log4J-Protokollierungssteuerkomponente erstellen. Hierzu definieren Sie diese in der Datei `Log4J.properties`. Sie können die in Log4J integrierten Treiber verwenden, beispielsweise den Standardtreiber, der mit der folgenden Anweisung definiert ist:
`Log4J.appender.Default=org.apache.Log4J.FileAppender`

Der Ausdruck `org.apache.Log4J.FileAppender` definiert diesen Appender so, dass die Klasse "FileAppender" verwendet wird. Zusätzliche Log4J-kompatible Treiber sind im Internet verfügbar, beispielsweise Treiber, die eine Protokollierung unter Verwendung von JMS oder JDBC ausführen können. Zur Verwendung solcher Treiber müssen diese im Verzeichnis `jars` der IBM Security Directory Integrator-Installation installiert werden. Anschließend können unter Verwendung dieser zusätzlichen Treiber in der Datei `Log4J.properties` Appender definiert werden. Weitere Informationen finden Sie unter der Adresse <http://jakarta.apache.org/log4j/docs>.

Neben dem Einsatz der integrierten Protokollierung von IBM Security Directory Integrator können Sie durch das Hinzufügen von Script-Code in Ihrer Fertigungslinie eine Protokollierung vornehmen. Dies ist detaillierter im IBM Knowledge Center for IBM Security Directory Integrator im Abschnitt *Konfiguration* beschrieben. Dort erfahren Sie auch, wie der interaktive Debugger eingesetzt wird.

Scriptbasierte Protokollierung

Sie können mit JavaScript jederzeit und an jeder Stelle, an der eine Scripterstellung möglich ist (Hooks, Scriptkomponenten usw.), Nachrichten an die konfigurierten Protokollfunktionen der Fertigungslinie absetzen.

Die verfügbaren expliziten Aufrufe von `logmsg()` (also **`task.logmsg()`** & **`main.logmsg()`**) können mit einem optionalen Zeichenfolgeparameter die Log4J-Ebene angeben, auf der die Nachrichten protokolliert werden sollen. Standardeinstellung ist INFO. Falls die durch den Benutzer angegebene Protokollebene für Log4J ungültig ist, wird die Nachricht auf der Ebene DEBUG protokolliert. Zu den Ebenen gehören DEBUG, INFO, WARN, ERROR und FATAL.

Bei Verwendung des Aufrufs

```
task.logmsg()
```

werden Ihre Nachrichten zusammen mit den anderen Nachrichten aus der Fertigungslinie protokolliert. Wenn Sie Ihre Fertigungslinie über den Konfigurationseditor ausführen, erfolgt die Protokollierung im Ausgabefenster des Konfigurationseditors. Verwendet Ihre Fertigungslinie auch andere Protokollierungsmethoden, werden die Nachrichten dort ebenfalls ausgegeben.

Bei Verwendung des Aufrufs

```
main.logmsg()
```

wird Ihre Nachricht zusammen mit anderen Nachrichten aus der Konfigurationsinstanz protokolliert. Die Protokollierung erfolgt in der/den Protokolldatei(en) oder

anderen durch die Konfigurationsinstanz erstellten Protokollfunktionen, die normalerweise im Konfigurationseditor nicht angezeigt werden.

Protokollierung mit Log4J-Standardklasse

Nachstehend erhalten Sie Informationen zur Protokollierung mit der Log4J-Standardklasse.

Die Konfiguration der Standardprotokollierung von IBM Security Directory Integrator, für die Apache Log4J verwendet wird, erfolgt global (mit der Datei `Log4J.properties`, die globale Standardwerte für `ServerTasks` angibt) oder aber mit dem Konfigurationseditor spezifisch für jede Fertigungslinie oder Konfigurationsdatei in ihrer Gesamtheit. Um dieses Flexibilitäts- und Anpassungsniveau bereitzustellen, wird die Java-Log4J-API verwendet.

In diesem Abschnitt sind nur die Parameter erläutert, die beschreiben, auf welche Weise Nachrichten protokolliert werden.

Alle Konfigurationsfenster für die Protokollierung funktionieren identisch: Für jedes Fenster können Sie eines oder mehrere Protokollschemas definieren. Diese sind zusätzlich zu allen in der Datei `Log4J.properties` festgelegten Standardwerten (siehe „Log4J-Standardparameter“ auf Seite 260) gleichzeitig aktiv.

Viele (jedoch nicht alle) Protokollfunktionen unterstützen eine Option für die Zeichencodierung und steuern damit, in welchem Zeichensatz die Protokolldateien geschrieben werden. Es gibt viele verschiedene Zeichensätze. Eine informelle Übersicht finden Sie unter der Adresse <http://download.oracle.com/javase/6/docs/technotes/guides/intl/encoding.doc.html>.

Die folgenden Protokollschemas sind möglich:

FileRollerAppender

Es kann vorkommen, dass Sie die Protokollierung in Dateien vornehmen, aber eine begrenzte Anzahl von Dateien verwenden wollen, da diese Ihre Datenträger belegen können. Das Schema "FileRollerAppender" generiert für jede Ausführung des Servers eine neue Datei. Das System speichert nur die angegebene Anzahl der vorherigen Protokolle. Falls Ihr Protokoll "mylog.txt" heißt und Sie 2 Generierungen anfordern, sind nach 3 Ausführungen eine Datei "mylog.txt" (für die zuletzt erfolgte Ausführung) sowie die Dateien "mylog.txt.1" und "mylog.txt.2" vorhanden, wobei die Datei "mylog.txt.2" das älteste Protokoll darstellt. Ab diesem Zeitpunkt werden keine weiteren Dateien, sondern nur neuere Versionen mit demselben Namen erstellt. Bewahren Sie zwei Generierungen von Sicherungsdateien auf.

Für das Schema **FileRollerAppender** gibt es die folgenden Parameter:

Dateipfad

Der Name der Datei, in der die Protokollierung erfolgen soll. Der Pfad ist relativ und bezieht sich auf die Position, an der IBM Security Directory Integrator installiert ist. Das spezielle, in Dateinamen verwendete Makro {0} wird durch den Namen des Servers ersetzt. Analog wird das in Dateinamen verwendete Makro {1} durch eine eindeutige Kennung ersetzt, die das System automatisch generiert. Das Makro {1} ist für den Sonderfall der Verwendung von "FileRollerAppender" unerheblich, jedoch dann wichtig, wenn Sie eindeutige Dateinamen wünschen.

Anzahl der Sicherungsdateien

Falls Sie den Wert "mylog.txt" als Dateipfad verwenden und zwei Sicherungsdateien auswählen, werden die Dateien für die beiden vorherigen Ausführungen in "mylog.txt.1" und "mylog.txt.2" umbenannt, sobald eine dritte Ausführung stattfindet.

Layout

Bestimmt das Format der Protokollnachricht. Mögliche Optionen:

- **Muster** (Diese Option wird verwendet, wenn Sie die Weise anpassen wollen, in der die Nachrichten protokolliert werden.)
- **Einfach** (Dieses Format enthält lediglich die Protokollebene und die Nachricht.)
- **HTML** (Dieses Format erstellt eine HTML-Datei, die einige (relative) Zeitinformationen, Threadinformationen, die Protokollebene, die Kategorie und die Nachricht enthält.)
- **XML** (Dieses Format hat Ähnlichkeit mit dem Format "HTML", generiert jedoch (mit dem Namensbereichspräfix "Log4J") eine XML-Datei.)

Muster

Wird nur verwendet, wenn für **Layout** die Einstellung **Muster** angegeben ist. Informationen hierzu finden Sie unter „Eigene Protokollstrategien erstellen“ auf Seite 261.

Protokollebene

Gibt die Bewertungsstufe der Protokollnachrichten an. Mögliche Optionen (mit absteigendem Informationsumfang):

- **DEBUG** (Debugnachricht)
- **INFO** (Informationsnachricht)
- **WARN** (Warnung)
- **ERROR** (Fehler)
- **FATAL** (Schwerwiegender Fehler)

Zeichencodierung

Die zu verwendende Zeichencodierung, z. B. "Cp1252", "ISO-8859-1" usw.

Protokoll aktiviert

Klicken Sie auf diese Option, um die Verwendung dieses Appenders zu aktivieren.

ConsoleAppender

Nimmt die Protokollierung in der Konsole (Standardausgabe) vor. Dies ist das Fenster, in dem Sie den Server (ibmdisrv) gestartet haben, oder das Taskausführungsfenster im Konfigurationseditor (ibmditk). Für das Schema **ConsoleAppender** gibt es die folgenden Parameter:

Layout

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Muster

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Protokollebene

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Protokoll aktiviert

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

FileAppender

Nimmt die Protokollierung in einer Datei vor. Für das Schema **FileAppender** gibt es die folgenden Parameter:

Dateipfad

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

An Datei anhängen

Klicken Sie auf diese Option, damit die Protokolldaten an eine Datei angehängt werden. Wenn diese Option nicht aktiviert ist, wird die Datei überschrieben.

Layout

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Muster

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Protokollebene

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Zeichencodierung

Die zu verwendende Zeichencodierung, z. B. "Cp1252", "ISO-8859-1" usw.

Protokoll aktiviert

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Dies ist der standardmäßig konfigurierte Appender. Weitere Informationen finden Sie unter „Log4J-Standardparameter“ auf Seite 260.

SyslogAppender

Dieses Schema ermöglicht IBM Security Directory Integrator die Protokollierung im UNIX-Systemprotokoll (Syslog). Für das Schema **SyslogAppender** gibt es die folgenden Parameter:

Hostname/IP-Adresse

Gibt den Host an, auf dem die Protokollierung erfolgen soll.

Systemprotokollfunktion (Syslog)

Gibt gültige Funktionen an, die im Dropdown zu finden sind. Muss durch den Host unterstützt werden, auf dem die Protokollierung erfolgt.

Funktionszeichenfolge drucken

Wenn diese Option aktiviert ist, enthalten gedruckte Nachrichten den Funktionsnamen der Anwendung.

Layout

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Muster

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Protokollebene

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Protokoll aktiviert

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

NTEventLog

Dieses Schema ermöglicht Anwendungen die Protokollierung im Windows NT-Ereignisprotokoll (auf Windows-Plattformen). Für das Schema **NTEventLog** gibt es die folgenden Parameter:

Quelle

Der im NT-Ereignisprotokoll enthaltene Name der "Quelle". Normalerweise ist dies der Titel der Anwendung, die die Protokollierung ausführt.

Layout

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Muster

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Protokollebene

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Protokoll aktiviert

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

DailyRollingFileAppender

Der Appender für die tägliche Fortsetzungsdatei wechselt an jedem Tag turnusmäßig die Protokolldatei. Wenn die Ausgabedatei gewechselt wird, erhält sie einen Namen, der aus dem Basisnamen zuzüglich einer Datums-musterzeichenfolge besteht, also "dateiname.jjjj-mm-tt". Der Appender wird normalerweise verwendet, wenn der Parameter **An Datei anhängen** aktiviert ist. Für das Schema **DailyRollingFileAppender** gibt es die folgenden Parameter:

Dateipfad

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

An Datei anhängen

Abhängig von der ausgewählten Einstellung wird eine neue Datei erstellt oder die Daten werden an eine vorhandene Datei angehängt. Dies ist normalerweise bei Verwendung von **DailyRollingFileAppender** gewünscht.

Datumsmuster

Gibt an, wie häufig die Datei gewechselt wird. Wählen Sie im Dropdown-Feld die Auflösung aus (Minuten bis Monate). Beispiel: Falls der Dateipfad mit "example.log" und das Datumsmuster mit '. 'jjjj-mm-tt angegeben ist, wird die Protokolldatei "example.log" am 31.10.2003 um Mitternacht in die Datei "example.log.2003-10-31" kopiert. Die Protokollierung für den 31.10.2003 wird in der Datei "example.log" fortgesetzt, bis sie am nächsten Tag turnusmäßig gewechselt wird.

Layout

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Muster

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Protokollebene

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Zeichencodierung

Die zu verwendende Zeichencodierung, z. B. "Cp1252", "ISO-8859-1" usw.

Protokoll aktiviert

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Ziehen Sie auch das Beispiel unter „Protokollierung mit Log4J-Standardklasse“ auf Seite 255 hinzu.

SystemLogAppender

Dieser Appender erstellt Protokolldateien in einer Kataloghierarchie unter dem Verzeichnis *tdi-installationsverzeichnis/system_logs*. Für jede Konfigurationsdatei gibt es ein entsprechendes Verzeichnis mit Protokolldateien namens *AL_xxx*. Hierbei steht "xxx" für den Namen der ausgeführten Fertigungslinie.

Für diesen Appender gibt es die folgenden Parameter:

Muster

Gibt das Format des Protokolls gemäß der Definition durch LOG4J an. Der Standardwert lautet folgendermaßen:

```
"%d{ISO8601} %-5p [%c] - %m%n"
```

Folgende Werte sind zusätzlich im Feld verfügbar:

```
"%d{HH:mm:ss} %p [%t] - %m%n"  
"%p [%t] %c %d{HH:mm:ss,SSS} - %m%n"
```

Protokollebene

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Zeichencodierung

Die zu verwendende Zeichencodierung, z. B. "Cp1252", "ISO-8859-1" usw.

Protokoll aktiviert

Siehe vorstehende Angaben zum Schema **FileRollerAppender**.

Protokollebenen und Protokollebenensteuerung

Hier werden die verschiedenen Protokollebenen aufgeführt.

Die folgenden Protokollebenen sind möglich:

- ALL (Alle)
- DEBUG (Debugnachricht)
- INFO (Informationsnachricht)
- WARN (Warnung)
- ERROR (Fehler)
- FATAL (Schwerwiegender Fehler)
- OFF (Inaktiviert)

Bei der Ebene "ALL" wird alles protokolliert. Bei den Ebenen "DEBUG", "INFO", "WARN", "ERROR" und "FATAL" werden die Nachrichten immer stärker gefiltert. Bei Verwendung der Ebene "OFF" erfolgt keine Protokollierung.

Sie können Protokollnachrichten für das System- oder das Fertigungslinienprotokoll unter Verwendung der Methode "logmsg()" aus JavaScript an jeder Stelle absetzen, an der IBM Security Directory Integrator die Scripterstellung zulässt. Die Methode kann einen oder zwei Parameter verwenden. Weitere Informationen enthält die Java-API-Dokumentation für die Deklaration von "logmsg()" (Paket **com.ibm.di.server**, Klasse *AssemblyLine* oder Klasse *RS*).

Die Schnittstelle für die Methode "logmsg()" (sowohl Hauptmethode als auch Task) mit zusätzlichem Protokollebenenparameter ist **logmsg (String logLevel, String msg)**. Gültige Werte für "logLevel" sind "FATAL", "ERROR", "WARN", "INFO", "DE-

BUG". Sie entsprechen den für die Protokollappender verfügbaren Protokollebenen. Jeder nicht erkannte Wert wird als "DEBUG" behandelt.

Bitte beachten Sie, dass die JavaScript-Aufrufe für `logmsg()` von IBM Security Directory Integrator standardmäßig eine Protokollierung der Ebene "INFO" vornehmen. Dies bedeutet, dass bei Verwendung der Protokollebene "WARN" oder einer niedrigeren Ebene "logmsg" sowie alle Einstellungen für "Ausführliches Protokoll" ausgeschaltet werden. Mit dem Parameter für die Ebene des Aufrufs von "logmsg()" können Sie jedoch die Protokollebene für einzelne Aufrufe von "logmsg()" überschreiben.

Log4J-Standardparameter

Sie können mithilfe der Standardkonfiguration den Inhalt ändern.

Nach der Installation von IBM Security Directory Integrator wird *FileAppender* als Standardprotokollfunktion verwendet. Wenn Sie die Standardprotokollfunktion ändern wollen, müssen Sie den Inhalt der Datei `log4j.properties` im Ordner *tdi-installationsverzeichnis/etc* ändern. Die Standardkonfiguration lautet folgendermaßen:

```
# This is the default logger, you will see that it logs to ibmdi.log
log4j.appender.Default=org.apache.log4j.FileAppender
log4j.appender.Default.file=logs/ibmdi.log
log4j.appender.Default.layout=org.apache.log4j.PatternLayout
log4j.appender.Default.layout.ConversionPattern=%d{ISO8601} %-5p [%c] - %m%n
log4j.appender.Default.append=false
```

Die Protokollfunktion "FileAppender" schneidet den Inhalt der Datei `ibmdi.log` (Position: *tdi-installationsverzeichnis/logs*) bei jedem Starten des IBM Security Directory Integrator-Servers ab. Wenn Sie dieses Verhalten ändern wollen, müssen Sie die Eigenschaft `log4j.appender.Default.append` in die Einstellung "true" ändern.

Die Datei "log4j.properties" enthält zusätzlich zwei Beispiele dafür, wie die Standardprotokollfunktion in "RollingFileAppender" oder "DailyRollingFileAppender" geändert wird. Falls Sie eine dieser Funktionen verwenden wollen, müssen Sie lediglich die Kommentarzeichen bei der gewünschten Funktion entfernen und für die Protokollfunktion "FileAppender" Kommentarzeichen setzen:

```
#####ROLLING FILE SIZE APPENDER
##RollingFileAppender rolls over log files when they reach a certain size specified by the
##MaxFileSize parameter

#log4j.appender.Default=org.apache.log4j.RollingFileAppender
#log4j.appender.Default.File=logs/ibmdi.log
#log4j.appender.Default.Append=true
#log4j.appender.Default.MaxFileSize=10MB
#log4j.appender.Default.MaxBackupIndex=10
#log4j.appender.Default.layout=org.apache.log4j.PatternLayout
#log4j.appender.Default.layout.ConversionPattern=%d{ISO8601} %-5p [%c] - %m%n

#####DAILY OUTPUT LOG4J SETTINGS
## With the DailyRollingFileAppender the underlying file is rolled over at a user chosen frequency.
##The rolling schedule is specified by the DatePattern option

#log4j.appender.Default=org.apache.log4j.DailyRollingFileAppender
#log4j.appender.Default.file=logs/ibmdi.log
#log4j.appender.Default.DatePattern='.'yyyy-MM-dd
#log4j.appender.Default.layout=org.apache.log4j.PatternLayout
#log4j.appender.Default.layout.ConversionPattern=%d{ISO8601} %-5p [%c] - %m%n
```

Im Folgenden werden einige der Parameter erläutert, die in der Datei "Log4J.properties" (für "ibmdisrv" und "ibmditk") verwendet werden.

Die vollständige Dokumentation finden Sie unter der Adresse <http://jakarta.apache.org/log4j/docs>.

Log4J.rootCategory=DEBUG, Default

DEBUG ist die Protokollebene für den benannten Appender (bei Log4J "Default" genannt). Falls Sie für die Protokollebene die Einstellung "OFF" oder eine höhere Ebene als "INFO" verwenden, erhalten Sie keine Ausgabe von den Scriptprotokollnachrichten (siehe anschließende Protokollbegriffe):

Log4J.appender.Default

Definiert, welchen Appendertyp der benannte Appender "Default" besitzt. Gültige Werte:

- FileRollerAppender (Generiert für jede Ausführung des Servers eine neue Datei.)
- ConsoleAppender (Nimmt die Protokollierung in der Konsole vor.)
- FileAppender (Nimmt die Protokollierung in einer Datei vor.)
- SyslogAppender (Nimmt die Protokollierung im UNIX-Systemprotokoll vor.)
- NTEventLog (Nimmt die Protokollierung im Windows NT-Ereignisprotokoll vor.)
- DailyRollingFileAppender (Speichert alte Dateien mit einer Datumszeitmarke in deren Namen.)
- SystemLogAppender (In einer Ordnerstruktur unter "*stammverzeichnis/system_logs*".)

Log4J.appender.Default.file

Die Standardprotokolldatei für "FileAppender". Die Position ist relativ und bezieht sich auf das Installationsverzeichnis (Standardwert ist "ibmdi.log").

Log4J.logger.com.ibm.di.*

Die Protokollebene für verschiedene IBM Security Directory Integrator-Komponenten. Bitte beachten Sie, dass beispielsweise "ibmditk" die Protokollebene für den eigentlichen Konfigurationseditor von IBM Security Directory Integrator (und nicht für die in ihm ausgeführten Prozesse) anzeigt. Ändern Sie diese Werte nicht.

Eigene Protokollstrategien erstellen

Mit diesem Framework können Sie die Protokollierung für die verschiedenen Fertigungslinien differenzierter gestalten.

Mit diesem Framework können Sie die Protokollierung für die verschiedenen Fertigungslinien differenzierter gestalten.

Anmerkung: Diese Informationen sind für Benutzer gedacht, die weiterhin die Datei `global.properties` verwenden wollen, um die Ausgabe der Protokollierung anzupassen. Sie können die Ausgabe der Protokollierung auch über den Konfigurationseditor (ibmditk) anpassen.

Im folgenden Abschnitt wird ein Protokollschema namens "CONSOLE" definiert, das später durch bestimmte Fertigungslinien verwendet werden kann:

```
Log4J.appender.CONSOLE=org.apache.Log4J.ConsoleAppender
Log4J.appender.CONSOLE.layout=org.apache.Log4J.PatternLayout
Log4J.appender.CONSOLE.layout.ConversionPattern=%d [%t] %-5p - %m%n0
```

Damit die Fertigungslinie "myAL" dieses Schema verwenden kann, benötigen Sie die folgenden Zeilen:

Eine Beschreibung der Parameter für "ConversionPattern" enthält die vollständige Dokumentation für Log4J (Version 1.2). Die folgende Liste enthält einige der Parameter:

- %d** Datum/Zeit abhängig vom Format.
- %p** Priorität.
- %c** Kategorie.

Anmerkung: Dieser Parameter hat normalerweise das Format *typ.fertigungsliniename.xxx*. Für *typ* kann "EventHandler" oder "AssemblyLine" angegeben sein, *fertigungsliniename* ist der Name der Fertigungslinie (oder des durch den Ersteller benannten EventHandlers) und *xxx* ist eine eindeutige ID für den Thread. Die Angabe **%c{2}** gibt den *Namen der Fertigungslinie & die eindeutige ID* aus.

- %m** Nachricht.
- %n** Zeilenumbruch.
- %t** Threadname.

Kapitel 15. Traceerstellung und First-Failure Data Capture

Hierbei handelt es sich um eine Protokollierungsbibliothek, die Ähnlichkeit mit Log4j hat, jedoch innerhalb von IBM Security Directory Integrator speziell für Traceerstellung und First-Failure Data Capture (FFDC, Erfassung von Fehlerdaten beim ersten Auftreten) eingesetzt wird.

Zusätzlich zu der vom Benutzer konfigurierbaren Funktionalität, die in Kapitel 14, „Protokollierung und Debug“, auf Seite 253 beschrieben ist, ist IBM Security Directory Integrator in seinem gesamten Code mit Anweisungen für die Traceerstellung unter Verwendung des JLOG-Frameworks ausgestattet. In welchem Umfang dies für Sie als Endbenutzer erkennbar ist, ist von einer Reihe von Konfigurationsoptionen in der globalen Konfigurationsdatei `jlog.properties` sowie von der Serverbefehlszeilenoption `-T` abhängig.

Anmerkung: Normalerweise sollten Sie in der Lage sein, die Fehlerbehebung, das Debug und die Unterstützung für Ihre Lösungen unter Verwendung der Protokollierungsoptionen zu realisieren, die im Abschnitt Kapitel 14, „Protokollierung und Debug“, auf Seite 253 beschrieben sind. Wenn Sie sich jedoch aus einem bestimmten Grund mit der IBM Unterstützungsfunktion in Verbindung setzen, werden Sie möglicherweise aufgefordert, einige Parameter zu ändern, die sich auf die hier beschriebene Traceerstellungsfunktion beziehen, um den Unterstützungsprozess zu vereinfachen.

Erweiterungen für die Traceerstellung

Die meisten Connectors und Parser besitzen Anweisungen für den Trace-Einstieg und -Ausstieg. Es wurden zu den aufgeführten Elementen auf dem IBM Security Directory Integrator-Server Traceanweisungen hinzugefügt:

Listenelemente:

- Einstiegs- und Ausstiegspunkte für Methoden
- Interaktionen mit Drittanbietersoftware
- Threaderststellungen

Wissenswertes über die Traceerstellung

Die Traceerstellung erfolgt im Code von IBM Security Directory Integrator unter Verwendung des JLOG-Objekts "PDLogger". Das Objekt "PDLogger" (Problem Determination Logger = Protokollfunktion für Problembestimmung) protokolliert Nachrichten im Logxml-Format (ein Tivoli-Standard), das die IBM Unterstützungsfunktion kennt und für das sie Verarbeitungstools besitzt.

Die Basisebene der aufgezeichneten Informationen gemäß der Verarbeitung durch die PDLogger-APIs lautet:

Datum | Zeit | Klassenname | Methodenname | Maschinename | IP | {Einstieg/Ausstieg/Ausnahmebedingung} | [Parameter]

Die Basistraceinformationen sind Zeit, Ebene (Minimum, Medium, Maximum), Position im Code (also Methodenname) sowie Einstieg/Ausstieg. Das Zeichen "|" wird lediglich zu Dokumentationszwecken verwendet und ist nicht Bestandteil des tatsächlichen Protokolls.

Aus den folgenden Gründen werden für die Traceerstellung nicht die Log4J-Appender eingesetzt:

1. Die Traceerstellung muss immer aktiviert werden.
2. Es ist nicht sinnvoll, mehrere Traces im Server zu aktivieren (bei Verwendung der Appender könnte es für jede Fertigungslinie mehrere geben).

Das Objekt "PDLogger" ist an den JLOG-Handler **SnapMemory** sowie an **JlogSnapHandler** angehängt.

Der Handler **SnapMemory** protokolliert Tracenachrichten im Hauptspeicher. Für den Auslöser eines Protokollereignisses (d. h. ein Vorkommen einer bestimmten Tracenachricht für die Protokollebene gemäß der Definition durch den Filter `jlog.levelflt.level` oder ein Anwendungsabsturz oder das Vorkommen einer bestimmten TMS-XML-Nachrichten-ID) wird der Tracespeicherpuffer durch **JlogSnapHandler** in eine Datei geschrieben.

Damit Traceerstellungs- und Protokollnachrichten in IBM Security Directory Integrator über alle IBM Produkte hinweg eindeutig sind, wird ihnen das eindeutige Präfix **CTGDI** vorangestellt.

Allen Fehlernachrichten wird eine eindeutige TMSXML-Nachrichten-ID als Präfix vorangestellt, die die Fehlerursache und die Bedieneraktion angibt.

Allen Informationsnachrichten wird ebenfalls eine eindeutige TMSXML-Nachrichten-ID als Präfix vorangestellt, die unter Umständen (jedoch nicht immer) die Bedieneraktion angibt.

Traceerstellung konfigurieren

Sie können die gewünschte Tracestufe mit der Eigenschaft `jlog.logger.level` in der Datei `jlog.properties` festlegen.

Als Tracestufe kann eine der folgenden JLOG-Protokollebenen angegeben werden (diese sind in der folgenden Liste in absteigender Wertigkeit angegeben):

- FATAL
- ERROR
- WARNING
- INFO
- DEBUG_MIN
- DEBUG_MID
- DEBUG_MAX

Die Standardstufe ist FATAL.

Die Standardtracestufe wird, ebenso wie die Angabe, ob für die Traceerstellung eine Datei oder der Hauptspeicher verwendet werden soll, in der Standarddatei `jlog.properties` definiert. Diese Datei befindet sich im Ordner `tdi-installationsverzeichnis/etc`. Falls Sie ein Lösungsverzeichnis verwenden, befindet sie sich im Ordner `tdi-lösungsverzeichnis/etc`.

Tracestufen dynamisch festlegen

Mit IBM Security Directory Integrator werden die Scripts "LogCmd.bat" (für Windows) und "LogCmd.sh" (für UNIX-Systeme) ausgeliefert. Sie können damit die Tracestufen dynamisch festlegen.

Die JLOG-Protokollfunktion startet einen Befehlsserver am Standardport (9992), der für die vom Befehlszeildienstprogramm "logcmd" gesendeten Protokollbefehle empfangsbereit ist.

Damit das Script "logcmd" ordnungsgemäß ausgeführt wird, muss zunächst der Befehlsserver gestartet worden sein. Zum Starten des Protokollbefehlsservers müssen Sie "jlog.noLogCmd=false" in der Datei `jlog.properties` festlegen.

Der Empfangsport dieses Servers kann geändert werden, indem die Eigenschaft "jlog.logCmdPort" in der Datei `jlog.properties` auf den gewünschten Wert gesetzt wird. Weitere Informationen zu diesen Eigenschaften enthalten die Kommentare in der Datei `jlog.properties`.

Der Befehl "logcmd" hat die folgende Syntax:

```
logcmd -o portnummer { [-h] | [help] |  
                    [list {knotenname} ] |  
                    [config knotenname] |  
                    [set knotenname schlüsselname=wert |  
                    [remove knotenname {schlüsselname} ] |  
                    [dump handlername] | [save {all} ] }
```

Hierbei gilt Folgendes:

-o portnummer

Die Portnummer, über die die Verbindung zum Protokollbefehlsserver hergestellt wird. Falls kein Port angegeben wird, wird von der Verwendung des Standardports (9992) ausgegangen.

-h | help

Zeigt Informationen zur Syntax des Befehls an.

list Listet die Namen aller bekannten Protokollierungsobjekte (Knoten) auf.

list knotenname

Listet die Namen der untergeordneten Elemente des Knotennamens auf. Nicht alle Protokollierungsobjekte besitzen untergeordnete Elemente.

config knotenname

Listet alle Konfigurationseigenschaften für den Knoten auf.

set knotenname schlüsselname=wert

Legt einen Eigenschaftsschlüssel für den Knotennamen fest. Falls das Protokollierungsobjekt (knotenname) nicht vorhanden ist, wird das Protokollierungsobjekt erstellt und die Eigenschaft wird hinzugefügt.

remove knotenname

Entfernt das Konfigurationsobjekt (knotenname). Ein Protokollierungsobjekt, das aus dieser Konfiguration instanziiert wurde, wird durch das Entfernen des Konfigurationsknotens nicht beeinflusst.

remove knotenname schlüsselname

Entfernt die Konfigurationseigenschaft (schlüsselname) aus dem Protokollierungsobjekt (knotenname). Falls dieses Objekt eine hierarchische Vererbung von Eigenschaften unterstützt, wird durch einen nachfolgenden Befehl `logcmd config knotenname` der soeben entfernte Schlüssel möglicherweise angezeigt. In diesem Fall wurde er von einem Vorgänger geerbt.

save {all}

Speichert die Protokollierungskonfiguration im persistenten Speicher. Bei Angabe von **all** wird die gesamte Konfiguration gespeichert. Andernfalls werden nur diejenigen Konfigurationsknoten gespeichert, die ursprünglich aus der Datei geladen wurden.

Nützliche JLOG-Parameter

Nachstehend sind die JLOG-Parameter aufgeführt.

Eigenschaft	Wert	Beschreibung
jlog.snapmemory.queueCapacity	Standardwert: 10000	Die Anzahl der Protokollereignisse, die in der Warteschlange des Handlers "SnapMemory" gespeichert werden können.
jlog.snapmemory.dumpEvents	true	Wenn diese Eigenschaft auf "true" gesetzt ist, sendet der Handler unverzüglich alle eingereichten Ereignisse an seine Ausgabelistener. Anschließend kann die Eigenschaft auf "false" zurückgesetzt werden.
jlog.snapmemory.userSnapDir	CTGDI/FFDC/user/	Das Verzeichnis, in dem die Tracespeicherauszugsdatei abgelegt werden soll, wenn ein Benutzer unter Verwendung der Scripts "logcmd" eine FFDC-Aktion auslöst.
jlog.snapmemory.isSync	Standardwert: false	Wenn diese Eigenschaft auf "true" gesetzt ist, werden die Protokollereignisse synchron in der Momentaufnahmedatei ausgegeben. Dies startet keinen neuen Thread und bewirkt, dass die Protokollfunktion blockiert wird, bis die Momentaufnahme vollständig ist.
jlog.snapmemory.userSnapFile	userTrace.log	
jlog.snapmemory.triggerFilter	jlog.levelflt	Der Stufenfilter, der für die JFFDC-Aktion verwendet werden soll.
jlog.snapmemory.msgIds	*E	Der für die JFFDC-Aktion zu verwendende TMSXML-Nachrichtenfilter.
jlog.snapmemory.mode	PASSTHRU oder BLOCK (Standardwert: PASSTHRU)	Wenn die Eigenschaft für die Nachrichten-IDs (msgIDs) auf "BLOCK" gesetzt ist, werden die aufgelisteten IDs geblockt. Ist sie auf "PASSTHRU" gesetzt, werden die aufgelisteten IDs an den Filter gesendet.
Jlog.snapmemory.msgIDRepeatTime	10000 (in Millisekunden)	Gibt in Millisekunden die Mindestzeit nach der Übergabe eines Protokollereignisses mit einer angegebenen TMS-Nachrichten-ID an, bevor ein weiteres Protokollereignis mit derselben ID übergeben werden kann.

Der Standardwert für "jlog.snapmemory.triggerFilter" legt einen Auslöserfilter namens "jlog.levelflt" fest. Ein Attribut für einen solchen Filter ist die Nachrichtenbewertung, die einen der oben beschriebenen JLOG-Protokollwerte annimmt. Standardmäßig konfigurieren die Einträge

```
jlog.levelflt.className=com.ibm.log.LevelFilter  
jlog.levelflt.level=FATAL
```

den FFDC-Code so, dass der Hauptspeicherpuffer an das Traceprotokoll ausgegeben wird, wenn eine Tracenachricht mit der Bewertung "FATAL" auftritt. Die Eigenschaft "jlog.levelflt.level" kann ebenfalls jeden der anderen Werte für die Protokollebene annehmen. Sinnvoll sind jedoch lediglich die Werte "ERROR" oder "FATAL", da andernfalls der FFDC-Speicherauszug sehr umfangreich ist, was eine erhebliche Leistungsminderung des IBM Security Directory Integrator-Servers verursacht.

Kapitel 16. Verwaltung und Überwachung

Mit AMC können Sie IBM Security Directory Integrator-Konfigurationen und -Fertigungslinien über Fernzugriff starten, stoppen und verwalten. Nachstehend erhalten Sie Informationen zu AMC.

Die Benutzerschnittstelle "Administration and Monitoring Console" (AMC) von IBM Security Directory Integrator ist in Integrated Solutions Console (ISC) implementiert.

Bei IBM Security Directory Integrator wird zusammen mit AMC auch Action Manager ausgeliefert. Action Manager ist eine eigenständige Java-Anwendung, die mit der AMC-Datenbank interagiert und die ferne Server-API verwendet, um ferne Fertigungslinien zu verwalten.

AMC besteht aus einer Java-WAR-Datei (Webarchiv) und einer WAB-Datei (Webpaket), die in Integrated Solutions Console Standard Edition (ISC SE) und IBM Dashboard Application Services Hub implementiert werden können.

Die aktuelle Version von Action Manager, die im AMC-Paket von IBM Security Directory Integrator enthalten ist, unterstützt IBM Security Directory Integrator Version 7.1.1, 7.1 und 7.0. Ältere IBM Security Directory Integrator-Versionen als Version 7.0 werden nicht unterstützt.

Anmerkung: IBM Security Directory Integrator sowie mit diesem Produkt entwickelte und implementierte Lösungen können über die IBM Security Directory Integrator-Schnittstelle für Java Management Extension (JMX) auch mit IBM Tivoli Monitoring (ITM) Server und Portal oder IBM Netcool/OMNIBus überwacht werden. Anhand der in Anhang B, „Überwachung mit externen Tools“, auf Seite 393 dargestellten unterstützten Beispiele erfahren Sie, wie Sie diese Überwachung realisieren können.

Installation und Konfiguration

Mithilfe der hier bereitgestellten Links können Sie Administration and Monitoring Console installieren.

Informationen zur Installation von IBM Security Directory Integrator und AMC finden Sie unter „IBM Security Directory Integrator installieren“ auf Seite 9. Durch die Installation von AMC wird ebenfalls Action Manager installiert. Falls Sie für die Implementierung von AMC eine angepasste IBM Dashboard Application Services Hub-Instanz verwenden oder die Implementierung von AMC während der Installation verzögern wollen, finden Sie im Abschnitt „AMC in angepasster ISC SE-Instanz oder IBM Dashboard Application Services Hub implementieren“ auf Seite 51 Angaben über zusätzliche Implementierungsanforderungen.

AMC in Integrated Solutions Console implementieren

Mit den hier aufgeführten Anweisungen können Sie AMC in Integrated Solutions Console implementieren.

Informationen zu diesem Vorgang

Bei den folgenden Anweisungen wird davon ausgegangen, dass Sie mit den Installationsverfahren für IBM Security Directory Integrator vertraut sind. Informationen zur Installation von IBM Security Directory Integrator können Sie unter „Plattform-spezifisches Installationsprogramm von IBM Security Directory Integrator verwenden“ auf Seite 13 nachlesen. Durch die Installation von AMC wird ebenfalls Action Manager installiert.

Falls Sie AMC von IBM Security Directory Integrator automatisch in ISC implementieren wollen, wählen Sie eine der folgenden Optionen aus:

- Integrierte Instanz von ISC SE
- Vorhandene Instanz von ISC

Soll AMC bei der Installation nicht automatisch in ISC implementiert werden, wählen Sie die Option "Keine Angabe - Die Anwendung wird zu einem späteren Zeitpunkt manuell implementiert" aus.

Wenn Sie während der Installation von IBM Security Directory Integrator die Option "Integrierte Instanz von Integrated Solutions Console Standard Edition" oder "Vorhandene Instanz von Integrated Solutions Console" auswählen, installiert das Installationsprogramm ISC automatisch und implementiert AMC automatisch in ISC.

Gehen Sie folgendermaßen vor, um AMC in ISC SE zu installieren und zu implementieren:

1. Rufen Sie das Installationsprogramm von IBM Security Directory Integrator auf.
2. Wählen Sie während der Installation die Option für die **benutzerdefinierte** Installation aus. (Bei einer Installation des Typs "Standard" steht die Option für AMC nicht zur Verfügung.)
3. Wählen Sie im Installationsfenster "Produktkomponenten auswählen" die Optionen **Administration and Monitoring Console** sowie **Integrierte Webplattform** aus (die zweite Komponente schließt ISC SE ein).
4. Führen Sie die Installation von IBM Security Directory Integrator vollständig aus.

AMC mit dem Installationsprogramm von IBM Security Directory Integrator als Windows-Dienst oder UNIX-Prozess implementieren

Sie können AMC mit dem Installationsprogramm von IBM Security Directory Integrator als Windows-Dienst oder UNIX-Prozess implementieren.

Sofern die folgenden Bedingungen erfüllt sind, können Sie AMC als Windows-Dienst oder als UNIX-Prozess registrieren:

- Der Benutzer, der IBM Security Directory Integrator installiert, muss Administratorrechte besitzen (Administratorgruppe unter Windows oder Root unter UNIX).
- Für die Installation von AMC muss die Option "Integrierte Instanz von Integrated Solutions Console Standard Edition" ausgewählt worden sein.
- Sie haben die Option "AMC als Systemservice registrieren" ausgewählt und einen Namen für den Service vergeben. Der Standard servicename lautet "tdiamc".

AMC in einer vorhandenen IBM WebSphere Application Server-Umgebung implementieren

Sie können AMC in einer vorhandenen IBM WebSphere Application Server-Umgebung implementieren.

Für die Implementierung von AMC in der vorhandenen IBM WebSphere Application Server-Umgebung müssen Sie die Scriptdatei `tdiISCHome.bat` oder `tdiISCHome.sh` ändern, um die folgenden Parameter zu definieren:

- Setzen Sie den Parameter `TDI_ISC_RUNTIME` auf den Wert für IBM WebSphere Application Server.
- Setzen Sie den Parameter `TDI_ISC_HOME` auf das Verzeichnis "WAS_HOME".

Beispiel:

```
set TDI_ISC_RUNTIME=WAS
set TDI_ISC_HOME=C:\Program Files\IBM\WebSphere\AppServer
```

AMC und Action Manager starten sowie Anmeldung durchführen

Zum Starten und Stoppen von AMC und Action Manager können Sie die hier aufgeführten Scripts ausführen.

Informationen zu diesem Vorgang

Die Scripts werden im Ordner `tdi-installationsverzeichnis/bin/amc` bereitgestellt:

- Führen Sie zum Starten von AMC das Script `start_tdiamc` aus.
- Führen Sie zum Starten von Action Manager das Script `startAM` aus.

Weitere Informationen zu diesen Scripts enthält der Abschnitt „Befehlszeilendienstprogramme für AMC und Action Manager“ auf Seite 321.

Die obigen Scripts starten AMC und AM unter Verwendung einer Derby-Datenbank, die auf der Maschine lokal konfiguriert ist und im Netzmodus auf "localhost" an Port 1528 ausgeführt wird. Zusätzliche Angaben über alternative Einstellungen und Konfigurationen für AMC und AM finden Sie in den Abschnitten „AMC aktivieren“ auf Seite 270 und „Action Manager aktivieren“ auf Seite 281.

Sobald AMC gestartet wurde, können Sie über die folgende URL darauf zugreifen: `http://localhost:13100/ibm/console`. Weitere Informationen finden Sie im Abschnitt „Bei der Konsole an- und abmelden“ auf Seite 289.

Zum Stoppen von AMC und Action Manager können Sie das Script `stop_tdiamc` bzw. `stopAM` ausführen.

Anmerkung:

1. Angaben über das Hinzufügen von Benutzern und Benutzerrollen finden Sie im Abschnitt „AMC in ISC“ auf Seite 273.
2. Informationen zu Action Manager enthält der Abschnitt „Action Manager“ auf Seite 274.
3. Die Verwendung der einzelnen Anzeigen in AMC ist in der Onlinehilfe für die jeweiligen Anzeigen und im Abschnitt „AMC-Benutzerschnittstelle“ auf Seite 289 beschrieben.
4. In AMC werden IBM Security Directory Integrator-Server registriert, damit Lösungen aus IBM Security Directory Integrator-Konfigurationen verwaltet werden.

den können. AMC kann nur Konfigurationen finden, die der jeweilige IBM Security Directory Integrator-Server bei seinem Start geladen hat. Die ferne Server-API ist bei IBM Security Directory Integrator-Servern standardmäßig aktiviert. Stellen Sie sicher, dass der Ordner *tdi-installationsverzeichnis/configs* alle Konfigurationen enthält, die Sie verwalten und überwachen wollen (bzw. legen Sie diese Konfigurationen im Ordner "config" Ihres Lösungsverzeichnisses ab, wenn Ihr Server ein Lösungsverzeichnis verwendet).

5. Ein Beispiel für den Ablauf bei der Verwendung von AMC und Action Manager für einige einfache Tasks finden Sie im Abschnitt „Beispielablauf der Erstellung von Lösungsansicht und Regeln“ auf Seite 327.

AMC aktivieren

Mit den hier aufgeführten Schritten können Sie Administration and Monitoring Console konfigurieren.

Die Konfigurationsdatei für AMC heißt `amc.properties`. Sie befindet sich auf derselben Ebene wie das Verzeichnis `WEB-INF`. Diese Datei enthält die Datenbankkonfigurationseigenschaften, die LDAP-Eigenschaften, die SSL-bezogenen Eigenschaften und Hilfsserverdetails für AMC.

AMC verwendet standardmäßig Derby Version 10, um Daten zu speichern. Beim erstmaligen Starten von AMC erstellt AMC im Web-Server-Verzeichnis einen Ordner namens `tdiamcdb` und erstellt die für die AMC-Funktionen benötigten Tabellen. Auf die Derby-Datenbank kann entweder im Netzmodus oder im integrierten Modus zugegriffen werden. Im Lieferzustand von AMC ist Derby standardmäßig im Netzmodus konfiguriert. Die folgenden Eigenschaften in der Datei `amc.properties` sind anwendbar, wenn Derby für den Netzmodus konfiguriert ist:

```
com.ibm.di.amc.jdbc.database=jdbc:derby://localhost:1528/tdiamcdb;create=true
com.ibm.di.amc.jdbc.driver=org.apache.derby.jdbc.ClientDriver
com.ibm.di.amc.jdbc.urlprefix=jdbc:derby:
com.ibm.di.amc.jdbc.user=APP
com.ibm.di.amc.jdbc.password=APP
com.ibm.di.amc.jdbc.start.mode=automatic
com.ibm.di.amc.jdbc.host=localhost
com.ibm.di.amc.jdbc.port=1528
com.ibm.di.amc.jdbc.sysIBM=true
```

Die obige Eigenschaft `com.ibm.di.amc.jdbc.database` gibt an, dass Derby im Netzmodus auf "localhost" an Port 1528 ausgeführt wird. Die Datenbank, auf die zugegriffen wird, heißt `tdiamcdb`. Mit der Angabe `create=true` wird festgelegt, dass AMC die Datenbank erstellen soll, wenn sie nicht gefunden wird.

Nachdem Sie Ihre Umgebung eingerichtet haben, sollten Sie die Einstellung `create=true` in `create=false` ändern, damit AMC im Fall einer Modifizierung des Datenbankpfades die Datenbank nicht erneut erstellt, sondern stattdessen die Ausnahmebedingung ausgibt, dass die Datenbank nicht gefunden wurde. Außerdem sollten Sie darauf achten, dass für die Datenbank ein absoluter Pfad angegeben wird, damit später keine Unklarheiten bezüglich des Datenbankpfades entstehen.

Andere Datenbanken als Derby können Sie konfigurieren, indem Sie die entsprechenden Eigenschaften festlegen (siehe hierzu „Konsoleigenschaften“ auf Seite 296).

AMC stellt für Action Manager separate Start- und Beendigungsscripts bereit. AMC lässt die Ausführung von Action Manager über Fernzugriff zu und stellt ein separates Script zum Starten oder Beenden von Derby bereit.

AMC kann auch so konfiguriert werden, dass die Verbindung zur Derby-Datenbank im integrierten Modus hergestellt wird. In diesem Fall kann die separate An-

wendung "Action Manager", die ebenfalls mit der AMC-Datenbank kommuniziert, keine Verbindung zur AMC-Datenbank herstellen. Dies liegt daran, dass im integrierten Modus gleichzeitig jeweils nur eine einzige JVM eine Verbindung zur Derby-Datenbank herstellen darf. Das folgende Beispiel zeigt eine Datei `amc.properties`, in der Derby für den integrierten Modus konfiguriert ist:

```
##Location of the database (embedded mode)
configured for embedded mode:
##Location of the database (embedded mode)
com.ibm.di.amc.jdbc.database=tdiamcdb
com.ibm.di.amc.jdbc.driver=org.apache.derby.jdbc.EmbeddedDriver
com.ibm.di.amc.jdbc.urlprefix=jdbc:derby:
com.ibm.di.amc.jdbc.user=APP
com.ibm.di.amc.jdbc.password=APP
```

Die Eigenschaft `com.ibm.di.amc.jdbc.database` zeigt auf die Position der AMC-Datenbank. Es empfiehlt sich, für diesen Wert einen absoluten Pfad anzugeben, damit später keine Unklarheiten bezüglich des Datenbankpfades bestehen.

Action Manager über Fernzugriff ausführen

Beachten Sie die hier aufgeführten Schritte sowie die speziellen Anweisungen zur Ausführung von Action Manager über Fernzugriff.

Ab IBM Security Directory Integrator 7.0 kann Action Manager über Fernzugriff ausgeführt werden, ohne dass zuvor AMC gestartet werden muss. Derby, die Datenbank für AMC, muss im Netzmodus ausgeführt werden, damit Action Manager eine Verbindung zu ihr herstellen kann. IBM Security Directory Integrator 7.0 stellt außerdem Scripts für das Starten und Beenden des Derby-Datenspeichers bereit, damit ein Benutzer Action Manager ohne einen Start von AMC über Fernzugriff starten kann.

Anmerkung:

1. Bevor Sie Action Manager zum ersten Mal starten, muss AMC mindestens ein Mal ausgeführt worden sein. Dies liegt daran, dass AMC die für Action Manager erforderlichen Datenbanktabellen erstellt.
2. Die im vorliegenden Abschnitt beschriebenen Scripts befinden sich im Ordner `tdi-installationsverzeichnis\bin\amc\ActionManager\` des Installationsverzeichnisses auf dem fernen Computer.
3. Die Anweisungen in den folgenden Abschnitten zum Starten und Beenden gelten für die Ausführung von Action Manager und Derby auf unterschiedlichen fernen Computern ohne aktive AMC.
4. Um zu ermitteln, ob AMC, Action Manager oder Derby gestoppt wurde, überprüfen Sie die Protokolle.

AMC und Action Manager starten

Wenn Sie Action Manager und Derby bei aktiver AMC ausführen wollen, starten Sie AMC, indem Sie `start_tdiamc.bat(sh)` eingeben. Starten Sie anschließend Action Manager, indem Sie `startAM.bat(sh)` eingeben. Das Script `tidamc` ruft das Script `startNetworkServer.bat(sh)` auf und startet auf diese Weise die Derby-Datenbank im Netzmodus.

Anmerkung: Im Script "`startAM.bat(sh)`" ist der Klassenpfad für alle von Action Manager benötigten JAR-Dateien definiert. Es gibt zwei Variablen namens `CLASSPATH` und `DB_CLASSPATH`. Die Variable `DB_CLASSPATH` enthält die nach Pfaden getrennte Liste der JAR-Dateien, die für die Erzielung der JDBC-Konnektivität mit der Datenbank erforderlich sind. Wenn AMC für die Verwendung von Oracle,

MS SQL Server oder DB2 konfiguriert ist, sollten die entsprechenden JAR-Dateien dieser Datenbanken für JDBC zur Variablen DB_CLASSPATH hinzugefügt werden.

AMC und Derby beenden

Das Script `stop_tdiadc.bat (sh)` ruft das Script `stopNetworkServer.bat (sh)` auf. Dies stellt sicher, dass der Derby-Netzserver bei der Beendigung von AMC gestoppt wird.

Anmerkung: Falls Action Manager aktiv ist, sollte diese Anwendung zuerst beendet werden.

Action Manager über Fernzugriff starten

Bei den Anweisungen in diesem Abschnitt wird davon ausgegangen, dass Action Manager und Derby auf unterschiedlichen Computern ausgeführt werden.

1. Starten Sie Derby mit dem Script `startNetworkServer.bat (sh)`.
2. Starten Sie Action Manager mit dem Script `startAM.bat (sh)`.

Das Script `startNetworkServer` wird verwendet, um den Derby-Datenbankserver im Netzmodus zu starten. Der Derby-Server wird im Netzmodus an Port 1528 gestartet. Der ausgewählte Port weicht vom Standardport für Derby ab.

Action Manager beenden

Action Manager wird mit dem Script `stopAM.bat (sh)` gestoppt, das sich im Verzeichnis `tdi-installationsverzeichnis/bin/amc` befindet. Dieses Script verwendet die Prozess-ID der gestarteten Action Manager-Instanz, um diese zu beenden. Die Prozess-ID wird durch das Script "startAM" abgerufen und in einer Datei gespeichert, die dann durch das Script "stopAM" gelesen wird.

Geben Sie zum Stoppen der Derby-Datenbank `stopNetworkServer` ein. Hierdurch wird der im Netzmodus ausgeführte Derby-Datenbankserver gestoppt. Dieser Vorgang sollte nicht vor, sondern nach dem Stoppen von Action Manager erfolgen.

AMC-Protokolle

Die AMC-Protokolle werden im ISC-Protokoll in der Umgebung gespeichert, in der AMC ausgeführt wird. Mithilfe der hier angegebenen Informationen können Sie die Protokolle konfigurieren.

- Bei ISC SE wird die Protokolldatei unter "`${LWI_HOME}/logs`" erstellt.
- Bei IBM Dashboard Application Services Hub werden die Protokolle an der Position "`${WAS_HOME}/profiles/${profileName}/logs/${serverInstance}/SystemOut.log`" abgelegt.

Zur Konfiguration der AMC-Protokolle kann die Datei `WEB-INF/classes/logging.properties` modifiziert werden. Die AMC-Protokollierung befolgt den Java-Protokollierungsstandard (`java.util.logging`).

Protokolle für Fertigungslinien können Sie im Fenster **Protokolle der Fertigungslinie** anzeigen und löschen. Das Fenster **Protokolle der Fertigungslinie** erreichen Sie über das Fenster **Status überwachen** durch Auswahl von **Status überwachen > Details der Lösungsansicht > Protokolle anzeigen**. Wählen Sie im Fenster **Details der Lösungsansicht** die Fertigungslinie aus, deren Protokolle Sie anzeigen wollen. Wählen Sie im Fenster **Protokolle der Fertigungslinie** alle Protokolle aus, die Sie

löschen wollen, und wählen Sie anschließend **Löschen** aus. Sie können eines oder mehrere Protokolle löschen. Zum Anzeigen eines Protokolls klicken Sie auf dessen Hyperlink.

Das Fenster **Details der Lösungsansicht** enthält außerdem die Tabelle **Action Manager-Protokolle**. Sie können Protokolle in der Tabelle **Action Manager-Protokolle** auswählen und löschen.

Zur Verwaltung aller Protokolle können Sie das Fenster **Protokollverwaltung** verwenden. Sie können Kriterien für das Anzeigen von Protokollen angeben und Protokolle für alle Fertigungslinien oder für eine einzige Fertigungslinie löschen. Sie können alle Protokolle für eine oder mehrere Fertigungslinien für einen bestimmten Datumbereich oder auch die letzten n Protokolle löschen (hierbei geben Sie die Anzahl für die letzten Protokolle ein).

AMC in ISC

Nachstehend erhalten Sie Informationen zum Anzeigen der in AMC integrierten Änderungsliste sowie zum Hinzufügen und Entfernen von Benutzern und verschiedenen Arten von Rollen, die zugewiesen werden können.

ISC (Integrated Solutions Console) wurde konzipiert, um eine einheitliche Konsole für die Organisation von Verwaltungskonsolenfunktionen bereitzustellen, die dem Branchenstandard entsprechende Technologien verwendet. Ab IBM Security Directory Integrator 7.0 wurde die Integration von AMC in ISC in einigen Punkten geändert. Die primären Navigationslinks für AMC sind folgende:

- **Administration and Monitoring Console**
 - Server
 - Lösungsansichten
 - Status überwachen
 - Action Manager
- **Erweitert**
 - Protokollverwaltung
 - Konsoleigenschaften
 - Bevorzugte Lösungsansichten
 - Eigenschaftsspeicher

Konsolbenutzerberechtigung

Mit der Funktion für die Konsolbenutzerberechtigung von ISC können Sie Benutzer zu AMC hinzufügen bzw. daraus entfernen. Bei AMC für IBM Security Directory Integrator Version 7.0 und höher gibt es die folgenden Rollen:

Tabelle 28. AMC-Rollen

Benutzerrolle	Beschreibung
administrator	Benutzer, denen diese Rolle zugeordnet ist, können die Rollen konfigurieren, die anderen Benutzern zugeordnet sind.
iscadmins	Benutzer, denen diese Rolle zugeordnet ist, können die Einstellungen der eigentlichen ISC-Konsole steuern.
Security Directory Integrator AMC Admin	Diese Rolle wird durch die IBM Security Directory Integrator-AMC-Anwendung berücksichtigt, die in der ISC-Konsole implementiert ist. Benutzer, denen diese Rolle zugeordnet ist, können Server, Rollen für Lösungsansichten und Konsoleigenschaften verwalten. (Diese Rolle entspricht der Rolle "superadmin", die in AMC vor IBM Security Directory Integrator 7.0 verwendet wurde.)
Security Directory Integrator AMC User	Diese Rolle wird durch die IBM Security Directory Integrator-AMC-Anwendung berücksichtigt, die in der ISC-Konsole implementiert ist. Benutzer, denen diese Rolle zugeordnet ist, können die IBM Security Directory Integrator-Ressourcen verwenden, die durch "SDI AMC Admin" bereitgestellt werden.

Innerhalb der Rolle *SDI AMC user* sind die Benutzerberechtigungen die zugeordneten Rollen, die in der Lösungsansicht angegeben sind:

- Administrator
- Konfigurationsadministrator
- Ausführen
- Lesen

Die Rollen steuern den Zugriff auf Funktionen in der Konsole. Es werden für einen Benutzer jeweils nur diejenigen Funktionen angezeigt, für die ihm Rollen zugeordnet sind. Benutzer mit der Rolle "Security Directory Integrator AMC Admin" besitzen beispielsweise automatisch Administratorrechte für alle Lösungsansichten. Administratoren können die für das Webverwaltungstool erforderlichen Eigenschaften konfigurieren (hierzu werden Eigenschaften der Datei `amc.properties` modifiziert, die im linken Navigationsfenster unter "Konsoleigenschaften" verfügbar sind). Ein Benutzer, der in AMC die Rolle "Security Directory Integrator AMC User" besitzt, entspricht dem aktuellen AMC-Benutzer ohne Administratorrechte. Benutzer mit der Rolle "Security Directory Integrator AMC User" besitzen keinen Zugriff auf die Verwaltungsfenster (z. B. "IBM Security Directory Integrator-Server" und "Konsoleigenschaften").

ISC bietet die Rolle "AMC Admin Group" oder "iscadmins". Ein Benutzer der Gruppe "iscadmins" hat dieselben Berechtigungen wie der Administrator.

Administrator und Gruppe "iscadmins"

Der Administrator, definiert als der Benutzer, der die Anwendung installiert hat, kann AMC-Benutzer verwalten. Er kann Benutzer zur lokalen Betriebssystemregistry für die AMC-Anwendung hinzufügen bzw. aus ihr entfernen und Rollen zuordnen oder bearbeiten. IBM Security Directory Integrator Version 7.0 und höher stellt eine neue Gruppe namens **SDI AMC Admin** bereit. Nach IBM Security Directory Integrator Version 6.1.1 ist die Rolle "superadmin" nicht mehr verfügbar.

Action Manager

Action Manager ist eine eigenständige Java-Anwendung, mit deren Hilfe Sie die Ausführung von mehreren IBM Security Directory Integrator-Servern und Fertigungslinien unter Verwendung von benutzerdefinierten Regeln, Auslöserbedingungen und in AMC definierten Aktionen überwachen können. Hier können Sie die Liste der verschiedenen Auslösertypen, Aktionen und Threads anzeigen.

AMC besitzt ein Fenster "Action Manager", in dem Benutzer verschiedene Action Manager-Regeln konfigurieren können.

Eine Regel ist eine Kombination aus einem Auslösertyp und einer Gruppe zugehöriger Aktionen. Eine Regel gibt an, dass bei Erkennung einer Auslöserbedingung die zugehörige Gruppe der Aktionen ausgeführt werden muss. Die verschiedenen in AMC verfügbaren Auslösertypen sind nachfolgend beschrieben:

Tabelle 29. Auslöser in Action Manager

Auslösertyp	Auslöserdetails	Benutzereingabe für Auslöser
Kein Auslöser	Eine Regel mit diesem Auslösertyp besitzt keine Auslöserbedingung und wird infolgedessen nie durch sich selbst ausgelöst. Diese Regel kann nur durch eine andere Regel ausgeführt werden.	Keine Details erforderlich.

Table 29. Triggers in Action Manager (Forts.)

Auslösertyp	Auslöserdetails	Benutzereingabe für Auslöser
Beim Starten der Fertigungslinie	Eine Regel mit diesem Auslösertyp wird ausgelöst, wenn Action Manager für die angegebene Fertigungslinie ein Startereignis empfängt.	Name der Fertigungslinie
Bei Beendigung der Fertigungslinie	Eine Regel mit diesem Auslösertyp wird ausgelöst, wenn Action Manager für die angegebene Fertigungslinie ein Beendigungsereignis empfängt.	Name der Fertigungslinie
Beim Laden der Konfiguration	Eine Regel mit diesem Auslösertyp wird ausgelöst, wenn eine Konfiguration geladen wird.	Name der zu ladenden Konfiguration
Beim Entladen der Konfiguration	Eine Regel mit diesem Auslösertyp wird ausgelöst, wenn eine Konfiguration entladen wird. Die Konfiguration muss geladen worden sein, damit sie entladen werden kann.	Name der zu entladenden Konfiguration

Tabelle 29. Auslöser in Action Manager (Forts.)

Auslösertyp	Auslöserdetails	Benutzereingabe für Auslöser
<p>Bei einer FL-Ergebnisabfrage</p>	<p>Anmerkung: Eine Regel mit diesem Auslösertyp sollte nicht für eine Fertigungslinie (FL) mit kurzer Ausführungsdauer verwendet werden. Dies liegt daran, dass Action Manager die Kennung des Fertigungslinienobjekts beim Empfang des Startereignisses für die Fertigungslinie speichert. Später verwendet Action Manager beim Empfang des Beendigungsereignisses für die Fertigungslinie diese Kennung, um die Attribute des letzten Work-Eintrags abzufragen. Falls die Fertigungslinie beendet wird, bevor Action Manager die Kennung speichern kann, ist Action Manager nicht in der Lage, die Attribute des Work-Eintrags abzufragen. Normalerweise ist eine Ausführungszeit von 10 Sekunden ausreichend (erreicht wird dies durch die Angabe von "system.sleep(10)" vor Beendigung der Fertigungslinie, beispielsweise im Epiloghook).</p> <p>Beim Ausführen der Regel Bei einer FL-Ergebnisabfrage fragt Action Manager die Fertigungslinie fortlaufend im angegebenen Abfrageintervall ab. Der Auslöser überprüft zuerst den Attributwert, wobei die Fertigungslinie nach dem angegebenen Abfrageintervall gestartet wird. Anschließend überprüft der Auslöser den Ergebniseintrag der Fertigungslinie.</p> <p>Die Regel "Bei einer FL-Ergebnisabfrage" wird ausgelöst, wenn der letzte Work-Eintrag der angegebenen Fertigungslinie das angegebene Attribut mit einer Übereinstimmung für die Bedingung und den Wert enthält. Diese Bedingung wird nur dann überprüft, wenn Action Manager ein Stoppereignis für die Fertigungslinie empfängt. Der Benutzer kann ein Zeitintervall angeben. Die angegebene Fertigungslinie wird abhängig vom angegebenen Zeitintervall periodisch ausgeführt. Eine Regel mit diesem Auslösertyp wird ausgelöst, wenn der letzte Work-Eintrag der angegebenen Fertigungslinie das angegebene Attribut mit einer Übereinstimmung für die Bedingung und den Wert enthält. Diese Bedingung wird nur dann überprüft, wenn Action Manager ein Stoppereignis für die Fertigungslinie empfängt.</p> <p>Um den Auslöser Bei einer FL-Ergebnisabfrage zu konfigurieren, geben Sie in die folgenden Felder Werte ein:</p> <ul style="list-style-type: none"> • Name der Fertigungslinie • Attribut • Bedingung • Wert • Abfrageintervall • Abfrageeinheit 	<p>Name der Fertigungslinie, Attribut, Bedingung, Wert, Abfrageintervall und Abfrageeinheit</p>

Tabelle 29. Auslöser in Action Manager (Forts.)

Auslösertyp	Auslöserdetails	Benutzereingabe für Auslöser
Bei einem Server-API-Fehler	Eine Regel mit diesem Auslösertyp wird ausgelöst, wenn Action Manager über die Server-API keine Verbindung zum fernen Server herstellen kann. Sie können für jede Fertigungslinie abhängig von der Ausführungszeit der Fertigungslinie unterschiedliche Abfragezeitintervalle konfigurieren.	Abfrageintervall und Abfrageeinheit
Beim Empfang eines Ereignisses	Eine Regel mit diesem Auslösertyp wird ausgelöst, wenn Action Manager ein Ereignis empfängt, das die angegebenen Kriterien erfüllt. Anmerkung: Falls eines der Kriterien ignoriert werden soll, geben Sie einfach keinen Wert für das Kriterium an.	Ereignistyp, Ereignisquelle, Ereignisdaten. Die Angabe der Ereignisdaten ist optional. Entweder der Ereignistyp oder die Ereignisquelle muss angegeben sein.
Bei einer Eigenschaft	Eine Regel mit diesem Auslösertyp wird ausgelöst, wenn die angegebene Eigenschaft die angegebene Bedingung erfüllt. Action Manager überprüft diese Eigenschaft in regelmäßigen Abständen. Bei der Konfiguration dieses Auslösers können Sie ein Abfrageintervall und eine Abfrageeinheit konfigurieren. Anmerkung: Diese Regel wird nur ein einziges Mal ausgelöst und erst dann in den Bereitstatus zurückgesetzt, wenn Action Manager feststellt, dass diese Eigenschaft die angegebenen Kriterien nicht mehr erfüllt. Die einmalige Auslösung erfolgt, damit die Regel nicht wiederholt ausgelöst wird, obwohl die Auslöserbedingung nur ein einziges Mal auftritt.	Abfrageintervall, Abfrageeinheit, Eigenschaftsname, Bedingung und Wert
Bei lokaler Variable	Eine Regel mit diesem Auslösertyp wird ausgelöst, wenn die angegebenen Variablen die angegebene Bedingung erfüllen. Action Manager überprüft diese Eigenschaft in regelmäßigen Abständen. Anmerkung: Diese Regel wird nur ein einziges Mal ausgelöst und erst dann in den Bereitstatus zurückgesetzt, wenn Action Manager feststellt, dass diese Variablen die angegebenen Kriterien nicht mehr erfüllen. Die einmalige Auslösung erfolgt, damit die Regel nicht wiederholt ausgelöst wird, obwohl die Auslöserbedingung nur ein einziges Mal auftritt.	Lokale Variable, Bedingung, Wert

Tabelle 29. Auslöser in Action Manager (Forts.)

Auslösertyp	Auslöserdetails	Benutzereingabe für Auslöser
Exit-Code der Fertigungslinie überprüfen	<p>Eine Regel mit diesem Auslösertyp wird ausgelöst, wenn eine Fertigungslinie mit einem Fehler beendet wird. Die Regel Exit-Code der Fertigungslinie überprüfen sucht außerdem für jede abnormale Beendigung einer Fertigungslinie nach einer Fehlerobjektzeichenfolge. Beim Auslöser Exit-Code der Fertigungslinie überprüfen können Sie unter "Auslöser konfigurieren" die Einstellung Fehlerobjekt überprüfen aktivieren. Geben Sie im Feld "Wert" die gewünschte Zeichenfolge für das Fehlerobjekt ein. Bitte beachten Sie, dass bei einem leeren Feld "Wert" die Regel für jede abnormale Beendigung einer Fertigungslinie ausgelöst wird. Falls die Einstellung "Fehlerobjekt überprüfen" nicht ausgewählt ist, wartet der Auslöser auf die Beendigung der Fertigungslinie und überprüft den Exit-Code auf einen (vom Benutzer eingegebenen) Attributwert. Geben Sie Werte sowohl für den Attributnamen als auch für den Attributwert ein.</p> <p>Beim Auslöser "Exit-Code der Fertigungslinie überprüfen" startet Action Manager die Fertigungslinie nicht mehr und es findet keine Abfrage statt. Der Auslöser überprüft das Ergebnis der Fertigungslinie nach ihrer Ausführung nur ein einziges Mal.</p>	Name der Fertigungslinie. Falls die Einstellung "Fehlerobjekt überprüfen" aktiviert ist, müssen Sie nur im Feld "Wert" eine Angabe bereitstellen. Falls die Einstellung "Fehlerobjekt überprüfen" inaktiviert ist, müssen Sie Werte für die Felder "Attribut", "Bedingung" und "Wert" angeben.
Bei Ablauf des Zeitraums seit der letzten Ausführung	<p>Eine Regel mit diesem Auslösertyp wird ausgelöst, wenn Action Manager feststellt, dass die angegebene Fertigungslinie nicht im angegebenen Zeitraum ausgeführt wurde. Hinweis: Diese Regel wird nur ein einziges Mal ausgelöst. Anschließend wartet Action Manager auf den Empfang eines Startereignisses für die Fertigungslinie, bevor er die Regel in den Bereitmodus zurücksetzt. Die einmalige Auslösung erfolgt, damit die Regel nicht wiederholt ausgelöst wird, obwohl die Auslöserbedingung nur ein einziges Mal auftritt.</p>	Name der Fertigungslinie, Nicht ausgeführt seit, Einheit
Zeitgeber	<p>Eine Regel mit diesem Auslösertyp wird in einem bestimmten Intervall fortlaufend ausgelöst. Dieses Intervall wird in Einheiten von Sekunden, Minuten, Stunden und Tagen definiert.</p>	Intervall und Einheit

Wenn eine Regel ausgelöst wird, führt Action Manager nacheinander die Aktionen aus, die der Regel zugeordnet sind. Die folgenden verschiedenen Aktionstypen sind in AMC verfügbar:

Tabelle 30. Action Manager-Aktionen

Aktion	Aktionsdetails	Benutzereingabe für Aktion
Fertigungslinie starten	Diese Aktion startet die angegebene Fertigungslinie der angegebenen Konfigurationsdatei auf dem angegebenen IBM Security Directory Integrator-Server. Das Feld "Konfiguration" sollte den vollständigen Pfad der Konfiguration auf dem fernen Server enthalten. Das Feld "Konfigurationskennwort" ist optional und nur dann erforderlich, wenn die ferne Konfiguration durch ein Kennwort geschützt ist.	Fertigungslinie, Konfiguration, Server, Konfigurationskennwort
Fertigungslinie stoppen	Diese Aktion stoppt die angegebene Fertigungslinie der angegebenen Konfiguration auf dem angegebenen IBM Security Directory Integrator-Server. Das Feld "Konfiguration" sollte den vollständigen Pfad der Konfiguration auf dem fernen Server enthalten.	Fertigungslinie, Konfiguration, Server
Regel aktivieren/ inaktivieren	Diese Aktion aktiviert bzw. inaktiviert die ausgewählte Regel.	Regelname, Status
Regel ausführen	Diese Aktion bewirkt die Ausführung der angegebenen Regel, was wiederum die Ausführung aller in dieser Regel angegebenen Aktionen einschließt.	Regelname
Benachrichtigungsereignis	Diese Aktion weist Action Manager an, ein Ereignis mit den angegebenen Details an den Server auszugeben, der der aktuellen Lösungsansicht zugeordnet ist. Details enthalten die Angaben der API "Session.sendCustomNotification()".	Ereignistyp, Quelle, Daten
Eigenschaft modifizieren	Diese Aktion bewirkt, dass Action Manager die ausgewählte Eigenschaft gemäß der angegebenen Operation modifiziert.	Eigenschaft, Operation, Wert
Eigenschaftswert kopieren	Diese Aktion bewirkt, dass Action Manager den Wert der Quelleneigenschaft in die Zieleigenschaft kopiert.	Von Eigenschaft, In Eigenschaft
In Protokoll schreiben	Diese Aktion bewirkt, dass ein Protokoll der angegebenen Bewertung/Nachricht/Beschreibung in den Action Manager-Protokollen und der AMC-Datenbank protokolliert wird. Dasselbe Protokoll wird angezeigt, wenn der Benutzer nach Auswahl der Optionen "Status überwachen -> Details der Lösungsansicht" die Ergebnistabelle für Action Manager anzeigt. Jede Regel sollte nach Möglichkeit mindestens eine Protokollaktion (inklusive beschreibendem Text) enthalten.	Bewertung, Nachricht, Beschreibung

Tabelle 30. Action Manager-Aktionen (Forts.)

Aktion	Aktionsdetails	Benutzereingabe für Aktion
E-Mail senden	Diese Aktion bewirkt, dass an den von Ihnen bestimmten Empfänger eine E-Mail gesendet wird. Der Inhalt der E-Mail wird von Ihnen bereitgestellt. Zusammen mit dem Inhalt stellt Action Manager vor dem Senden der E-Mail weitere Details bereit. Im Eingabebereich für den Inhalt sowie in der Betreffzeile können Sie den Variablenwert %EVENT_DATA% eingeben. Bei Angabe von %EventData% wird der tatsächliche Wert der Variablen "Eventdata" eingefügt, wenn die E-Mail gesendet wird. In entsprechender Weise kann auch %Action_Error% hier eingesetzt werden. Falls die Einstellung "Action Manager-Protokoll anhängen" aktiviert ist, werden die Action Manager-Protokolle (wie in der Datei "am_logging.properties" angegeben) als E-Mail-Anhang gesendet.	An, Von, Betreff, Einstellung für Action Manager-Protokoll anhängen" (ausgewählt oder abgewählt), Inhalt
Lokale Variablen modifizieren	Diese Aktion bewirkt, dass Action Manager den Wert der angegebenen Variablen erhöht, verringert oder auf den angegebenen Wert festlegt.	Variable, Operation, Wert
Befehl ausführen	Diese Aktion bewirkt, dass der angegebene Befehl auf dem Zielcomputer ausgeführt wird. Hierbei kann es sich um einen allgemeinen Befehl oder um einen speziellen IBM Security Directory Integrator-Befehl handeln.	Zielmaschine, Port, Benutzername, Kennwort, Schlüsselspeicher, Schlüsselspeicherkenwort, Protokoll, Befehl

Für Lösungsansichten in AMC konfigurierte Regeln werden in der Derby-Datenbank von AMC gespeichert. Bei seiner Ausführung stellt Action Manager eine Verbindung zur AMC-Datenbank im Netzmodus her, liest die mit Action Manager zusammenhängenden Tabellen und erstellt für jede angegebene Regel Threads im Hauptspeicher. Jeder dieser Threads ist für seine entsprechenden Auslöserbedingungen empfangsbereit bzw. fragt diese ab. In dem Moment, in dem ein Thread feststellt, dass seine Auslöserbedingung eingetreten ist, fragt er in der Datenbank die Gruppe der Aktionen ab, die der Regel zugeordnet sind, und führt diese Aktionen nacheinander aus.

Action Manager führt neben den Regelthreads, die für die Auslöserbedingungen empfangsbereit sind, die folgenden Threads aus:

1. HealthAssemblyLine - Der Thread "HealthAssemblyLine" löst in regelmäßigen Abständen die Fertigungslinien für den ordnungsgemäßen Betrieb aus, um den Status der Lösungen abzufragen und diesen Status wieder in der AMC-Datenbank zu protokollieren. Die Fertigungslinien für den ordnungsgemäßen Betrieb müssen den Status in den Attributen "healthAL.result" und "healthAL.status" ihres letzten Work-Eintrags speichern.
2. ServerStatusListener - Der Thread "ServerStatusListener" wird für jeden Server erstellt, der bei AMC registriert ist. Dieser Thread überprüft, ob der Server zugänglich ist. Falls nicht mehr auf den Server zugegriffen werden kann, werden alle für den Server erstellten Regelthreads beendet (mit Ausnahme von Regelthreads des Typs "Bei einem Server-API-Fehler"). Falls der Server zugänglich wird, werden analog für alle Regeln, die diesem Server zugeordnet sind, Regelthreads erstellt.
3. ConfigLoadReloadListener - Der Thread "ConfigLoadReloadListener" wird für jeden aktiven Server erstellt, der bei AMC registriert ist. Er wird auf dem fer-

nen Server für alle Ereignisse zum Laden/Entladen von Konfigurationen registriert. Regelthreads werden abhängig vom Konfigurationsereignis entsprechend beendet, erstellt oder aktualisiert.

4. ServerModificationListener - Der Thread "ServerModificationListener" überprüft, ob Aktualisierungen für die bei AMC registrierte Gruppe der Server vorliegen. Abhängig vom Typ der Änderung (Hinzufügen, Entfernen usw.) werden Regelthreads beendet, erstellt oder aktualisiert.
5. DatabaseModificationListener - Dieser Datenbank-Listener-Thread überwacht fortlaufend, ob Regeln hinzugefügt, modifiziert oder gelöscht werden. Wenn Änderungen an den Regeln festgestellt werden, werden die Action Manager-Threads zur Laufzeit entsprechend hinzugefügt/erneut erstellt.

Action Manager aktualisiert die AMC-Datenbank auch mit Details über seine Ausführung. Immer dann, wenn eine Action Manager-Regel ausgelöst wird, protokolliert Action Manager einen Eintrag in der AMC-Datenbank, der den Namen der ausgelösten Regel und den Zeitpunkt der Auslösung registriert. Falls für die Regel eine Protokollaktion konfiguriert wird, wird dies ebenfalls in der AMC-Datenbank protokolliert. Die entsprechenden Datenbankeinträge werden verwendet, um den jeweiligen Status in den Überwachungsfenstern von AMC anzuzeigen.

Action Manager aktivieren

Mit den hier aufgeführten Anweisungen können Sie Action Manager aktivieren.

Action Manager ist im Ordner *tdi-installationsverzeichnis/bin/amc/Action Manager* installiert. Dieser Ordner enthält die folgenden Dateien:

- *am_logging.properties*: Diese Datei steuert die Eigenschaften für die Action Manager-Protokollierung. Wie bei AMC wird auch hier der Protokollierungsstandard von "java.util.logging" befolgt.
- *am_config.properties*: Dies ist die Konfigurationsdatei für Action Manager.
- *testadmin.jks*: Dies ist die Truststore- und Schlüsselspeicherdatei von Action Manager.

Anmerkung: Diese Datei ist ein Beispiel für eine Truststore- und Schlüsselspeicherdatei. Zur Verbesserung der Sicherheit sollten Sie eine eigene Datei generieren.

Action Manager verwendet den Treiber für den Netzmodus, um die Verbindung zur Derby-Datenbank von AMC herzustellen.

Die folgenden Eigenschaften (in der Datei *am_config.properties*) müssen auf die AMC-Datenbank zeigen:

```
com.ibm.di.amc.am.jdbc.database=jdbc:derby://localhost:1528/C:/Program Files/IBM/AppSrv
/profiles/amcprofile/tdiamcdb;create=false
com.ibm.di.amc.am.jdbc.driver=org.apache.derby.jdbc.ClientDriver
com.ibm.di.amc.am.jdbc.urlprefix=jdbc:derby:
com.ibm.di.amc.am.jdbc.user=APP
{protect}-com.ibm.di.amc.am.jdbc.password=APP
com.ibm.di.amc.am.jdbc.start.mode=automatic
com.ibm.di.amc.am.jdbc.sysibm=true
com.ibm.di.amc.am.jdbc.networkserver.host=localhost
com.ibm.di.amc.am.jdbc.networkserver.port=1528
```

Anmerkung: Sowohl AMC als auch Action Manager unterstützen alternative Datenbanken wie MS SQL, Oracle usw. Damit AMC und Action Manager zu einer dieser alternativen Datenbanken eine Verbindung herstellen können, müssen die Konfigurationsanweisungen in den Dateien *amc.properties* und *am_config.properties* ganz anders lauten.

Beim Start versucht Action Manager, eine Verbindung zur AMC-Datenbank herzustellen. Falls hierbei Erstkonfigurationstasks fehlschlagen, wird Action Manager mit einer Ausnahmebedingungs-nachricht beendet. Ermitteln Sie in der Datei `am_config.properties`, ob auf die richtige Datenbank verwiesen wird. Falls die Datenbank-einstellungen richtig zu sein scheinen, vergewissern Sie sich, dass die Datenbank, zu der Action Manager eine Verbindung herzustellen versucht, im Netzmodus ausgeführt wird und dass AMC zu derselben Datenbank eine Verbindung herstellen kann. Zum Starten der Derby-Datenbank im Netzmodus können Sie das Script `startNetworkServer.bat(sh)` verwenden.

Die SSL-Einstellungen und Verschlüsselungseigenschaften für Action Manager werden durch die folgenden Eigenschaften konfiguriert:

```
# Action Manager SSL properties
javax.net.ssl.trustStore=tdi-installationsverzeichnis/serverapi/testadmin.jks
{protect}-javax.net.ssl.trustStorePassword=administrator
javax.net.ssl.trustStoreType=jks
javax.net.ssl.keyStore=tdi-installationsverzeichnis/serverapi/testadmin.jks
{protect}-javax.net.ssl.keyStorePassword=administrator
javax.net.ssl.keyStoreType=jks
# Action Manager encryption properties
com.ibm.di.amc.am.encryption.keystore = tdi-installationsverzeichnis/testserver.jks
com.ibm.di.amc.am.encryption.key.alias = server
com.ibm.di.amc.am.encryption.keystoretype = jks
com.ibm.di.amc.am.encryption.transformation = RSA
com.ibm.di.amc.am.stash.file = tdi-installationsverzeichnis/idisrv.sth
```

Diese Eigenschaften haben Ähnlichkeit mit den Verschlüsselungseigenschaften, die durch den Server verwendet werden. Zur Vereinfachung wurde die Position der Stashdatei als Eigenschaft `com.ibm.di.amc.am.stash.file` hinzugefügt. Action Manager verwendet standardmäßig die Schlüsselspeicher- und Stashdatei des Servers wieder, um geschützte Action Manager-Eigenschaften zu verschlüsseln und zu entschlüsseln.

Die weitere Konfiguration von Laufzeitregeln, Auslösern und Aktionen ist im Abschnitt „Action Manager“ auf Seite 309 beschrieben.

Action Manager-Status in Echtzeit

Mit den hier aufgeführten Anweisungen können Sie den Status von Action Manager in einem Fenster anzeigen. Außerdem können Sie den Inhalt des Fensters anzeigen.

Wenn Sie sich bei AMC anmelden, wird in der Anzeige **Willkommen bei Administration and Monitoring Console** in einer Zeile der Status von Action Manager angezeigt. In der Anzeige "Willkommen bei Administration and Monitoring Console" wird der Status von Action Manager in einem Link angezeigt (z. B. "Action Manager wird ausgeführt" oder "Action Manager wird nicht ausgeführt"). Klicken Sie auf den Hyperlink, um das Fenster **Action Manager-Status** aufzurufen. Im Fenster "Action Manager-Status" wird der Status von Action Manager in Echtzeit angezeigt; darüber hinaus enthält es Details zu Threads und Auslösern. Die Statusinformationen werden in diesem Fenster in Echtzeit angezeigt. Das Fenster enthält die folgenden Angaben:

- Action Manager-Status, z. B. die Bootzeit
- Details zu Action Manager-Threads
- Details zu Action Manager-Auslösern

AMC fragt Action Manager direkt über die von Action Manager zugänglich gemachten APIs ab. Wenn AMC keine Sitzung mit Action Manager aufbauen kann, wird dies so interpretiert, dass Action Manager nicht verfügbar ist, weil Action

Manager gestoppt wurde. Neben dem Action Manager-Status werden in AMC Details zu Threadinformationen und zu Auslösern angezeigt.

Action Manager erstellt eine Reihe von Threads. Einige Action Manager-Threads (z. B. "DatabaseModificationListener" und "ServerStatusListener") überwachen die Grundfunktionalität von Action Manager. Darüber hinaus erstellt Action Manager aus diesen Threads heraus die Threads für die einzelnen Auslöserregeln, die in AMC konfiguriert sind. Mithilfe der RMI-Schicht kann AMC den Status der einzelnen auslöserbezogenen Threads abfragen. Dank der RMI-basierten Abfrage hat AMC Kenntnis über den Status dieser Threads, über ihre Priorität usw. AMC kann außerdem die Auslöser abfragen, die während eines bestimmten Zeitraums ausgeführt wurden.

Mit der Anforderung, den Action Manager-Status in Echtzeit abzufragen, stehen zwei neue Eigenschaften in Zusammenhang. Die Eigenschaften, die es AMC ermöglichen, den Action Manager-Status in Echtzeit anzuzeigen, heißen `am.api.host` und `am.api.port`. Der Action Manager-Status verwendet eine Action Manager umgebende RMI-Schicht, in der eine API zugänglich ist, die AMC zum Abfragen des Action Manager-Status verwendet.

Auslöser für eine bestimmte Regel über AMC erzwingen

Mit dem Befehl **Auslöser erzwingen** können Sie die für die ausgewählte Regel konfigurierten Aktionen ausführen.

AMC ermöglicht es Benutzern, einen Auslöser für eine bestimmte Regel zu erzwingen. Durch das Erzwingen eines Auslösers erhält der Benutzer eine Vorstellung davon, welche Aktionen Action Manager ausführt, wenn die Regel ausgelöst wird. Wenn Sie **Regel inaktivieren** auswählen, wird die ausgewählte Regel inaktiviert.

Action Manager kann eine Reihe von Aktionen, die für einen bestimmten Auslöser (Regel) konfiguriert sind, explizit ausführen. Der AMC-Benutzer muss hierdurch nicht warten, bis die Auslöserbedingung einer Regel erfüllt wird, damit die konfigurierten Aktionen ausgeführt werden. Benutzer können Aktionen definieren, die ausgeführt werden sollen, um diese Aktionen zu testen. Benutzer können mit dem Befehl **Auslöser erzwingen** alle unterstützten Aktionen ausführen. Die Aktion **Umkehren** bzw. **Zurücksetzen** ist jedoch nur bei einem Teil der unterstützten Aktionen möglich.

- Eigenschaft modifizieren
- Eigenschaft kopieren
- In Protokoll schreiben
- Regel aktivieren oder inaktivieren

Sicherheit bei AMC und Action Manager

AMC ist eine webbasierte Anwendung für die Überwachung und Verwaltung ferner IBM Security Directory Integrator-Lösungen. Nachstehend erhalten Sie Informationen zu ihren Funktionen und ihren verschiedenen Sicherheitskombinationen.

Einführung

Die folgenden AMC-Funktionen wurden verbessert:

- Kennwörter, die in der Datei "amc.properties" gespeichert sind, werden verschlüsselt oder verdeckt.

- Zum Speichern von Schlüsselspeicherkeywörtern werden Stashdateien verwendet.
- Das Authentifizierungsschema BUILTIN wurde in der Derby-Datenbank aktiviert.

AMC verwendet die ferne Server-API, um mit IBM Security Directory Integrator zu kommunizieren. Daher können alle Sicherheitsbedingungen und Konfigurationseinstellungen, die für Clients der fernen Server-API von IBM Security Directory Integrator anwendbar sind, auch bei AMC eingesetzt werden.

Action Manager wird zusammen mit AMC installiert. Action Manager konfiguriert sich selbst. Sein Verhalten basiert auf Regeln, die von AMC-Benutzern in der AMC-Datenbank festgelegt werden. Zur Überwachung von fernen Fertigungslinien und für Aktionen aufgrund von konfigurierten Regeln verwendet Action Manager (wie AMC) die ferne Server-API von IBM Security Directory Integrator, um mit den IBM Security Directory Integrator-Servern zu kommunizieren.

Anmerkung: Die Kommunikation zwischen AMC und Action Manager über RMI wird nicht geschützt.

AMC und SSL

Mit den hier aufgeführten Anweisungen können Sie Administration and Monitoring Console im SSL-Modus ausführen.

Bei AMC können mehrere IBM Security Directory Integrator-Server registriert werden. Jeder IBM Security Directory Integrator-Server kann anders konfiguriert sein; beispielsweise kann ein IBM Security Directory Integrator-Server mit inaktiviertem SSL ausgeführt werden, ein anderer Server mit aktiviertem SSL und ein weiterer Server mit der angepassten Authentifizierung und aktiviertem SSL - dies ist nur eine der vielen unterschiedlichen Kombinationen. AMC kann verwendet werden, um zu diesen Servern gleichzeitig Verbindungen herzustellen und die Server zu überwachen. Wie bereits zuvor erläutert, sollte zum Konfigurieren von IBM Security Directory Integrator für die Ausführung im SSL-Modus die Eigenschaft **api.remote.ssl.on** in der Datei `global.properties` (oder `solution.properties`) auf **true** gesetzt werden.

Da es sich bei AMC um eine Webanwendung handelt, die in einem Web-Container ausgeführt wird, übernimmt AMC automatisch einige Eigenschaften und Sicherheitsbedingungen vom Web-Container. Falls für den Web-Container beispielsweise ein SSL-Schlüsselspeicher oder ein SSL-Truststore konfiguriert ist, wird dies automatisch auf AMC angewendet. AMC kann jedoch diese Einstellungen überschreiben und einen eigenen Schlüsselspeicher sowie Truststore angeben.

Damit AMC mit der fernen Server-API von IBM Security Directory Integrator bei Ausführung von SSL kommunizieren kann, muss für AMC ein Schlüsselspeicher konfiguriert sein, der das Zertifikat enthält, das durch die ferne Server-API von IBM Security Directory Integrator anerkannt ist (das also im Abschnitt mit den gesicherten Zertifikaten des IBM Security Directory Integrator-Truststores vorhanden ist). Außerdem muss für AMC ein Truststore konfiguriert sein, in dem das Zertifikat enthalten ist, das von der fernen Server-API von IBM Security Directory Integrator gesendet wird. Anders ausgedrückt, muss das im Schlüsselspeicher des IBM Security Directory Integrator-Servers vorhandene Zertifikat im AMC-Truststore enthalten sein und das im IBM Security Directory Integrator-Truststore enthaltene Zertifikat muss im AMC-Schlüsselspeicher enthalten sein.

Die Standardinstallation von IBM Security Directory Integrator wird beispielsweise mit bestimmten Speichern (Dateien ".jks") ausgeliefert. Bei einer Ausführung von IBM Security Directory Integrator im SSL-Modus müssen zum Herstellen einer Verbindung zu AMC der Schlüsselspeicher und der Truststore auf denselben Wert gesetzt sein, nämlich *tdi-installationsverzeichnis/serverapi/testadmin.jks*. Außerdem muss das Kennwort "administrator" verwendet werden. Da die Datei "testadmin.jks" sowohl gesicherte Zertifikate als auch Unterzeichnerzertifikate enthält, wird eine Verbindung aufgebaut. Es wird empfohlen, dass Sie eigene SSL-Schlüsselspeicher und -Truststores einrichten.

In AMC können Sie den Pfad des Truststores und des Schlüsselspeichers festlegen, indem Sie sich bei AMC mit der Rolle "SDI AMC Admin" (Konsolenadministrator) anmelden und zum folgenden Fenster navigieren: **Erweitert -> Konsoleigenschaften -> SSL-Einstellungen**. Die Einstellungen für den Truststore und den Schlüsselspeicher werden in die Datei **amc.properties** geschrieben, die sich im Ordner "tdiamc" des Web-Containers befindet. Alternativ können Sie die Datei **amc.properties** auch direkt bearbeiten. Bei IBM Security Directory Integrator 7.0 kann AMC in Integrated Solutions Console Standard Edition (ISC SE) oder in Integrated Solutions Console Advanced Edition (ISC AE) implementiert werden. Die Position der Datei testadmin.jks variiert abhängig von der ISC-Laufzeit. Ist AMC beispielsweise in ISC SE implementiert, lautet die Position *isc-laufzeitinstallationsverzeichnis/runtime/isc/eclipse/plugin-ins/AMC_7.0.0*. Ist AMC hingegen in IBM Security Directory Integrator implementiert, lautet die Position *isc-laufzeitinstallationsverzeichnis/systemApps/isc-lite.ear/tdiamc.war*. Das Kennwort für den Schlüsselspeicher und den Truststore ist standardmäßig auf "administrator" gesetzt. Um eine SSL-basierte Verbindung mit einem fernen IBM Security Directory Integrator-Server aufzubauen, müssen Sie den Server im Modus mit aktiviertem SSL starten. Bei einer Verbindung ohne SSL-Basis starten Sie den Server im Modus mit inaktiviertem SSL.

Achtung: Für SSL werden Standardeinstellungen bereitgestellt. Bei Verwendung der Standardzertifikate ist die Sicherheit jedoch nicht größer als bei einer normalen Verbindung. Daher sollten Sie nach der Installation die SSL-Standardzertifikate ersetzen und die Schlüsselspeicher sowie die Truststores entsprechend aktualisieren, damit die Sicherheit erhöht wird.

Für jeden über SSL ausgeführten IBM Security Directory Integrator-Server, der bei AMC registriert werden soll, müssen Sie das benötigte Zertifikat in den AMC-Truststore und das benötigte AMC-Schlüsselzertifikat in den IBM Security Directory Integrator-Truststore importieren. AMC muss also IBM Security Directory Integrator und IBM Security Directory Integrator muss AMC anerkennen, damit eine gesicherte bidirektionale SSL-Verbindung hergestellt werden kann.

Da AMC in einem Web-Container ausgeführt wird, lautet die URL für AMC `http://hostname:port/ibm/console`.

Action Manager überwacht aktive Konfigurationen und Fertigungslinien auf fernen IBM Security Directory Integrator-Servern anhand von Regeln, die in AMC konfiguriert sind. Zusammen mit Action Manager werden der Schlüsselspeicher und der Truststore ausgeliefert, die zum Herstellen einer Verbindung mit einem fernen IBM Security Directory Integrator-Server erforderlich sind. Die SSL-Eigenschaften sind in der Datei *am_config.properties* definiert. Die in den vorherigen Abschnitten aufgeführten Details über das Konfigurieren von AMC für SSL gelten ebenfalls für Action Manager.

AMC und ferne IBM Security Directory Integrator-Server

AMC kann über Fernzugriff Verbindungen zu mehreren IBM Security Directory Integrator-Servern herstellen. Sie können die einzelnen Server auf verschiedene Weise konfigurieren.

Mögliche Konfigurationen:

- Konfiguration ohne SSL
- Konfiguration mit SSL
- Angepasste Authentifizierung ohne SSL
- Angepasste Authentifizierung mit SSL

Diese Fälle werden nachfolgend jeweils im Detail erläutert.

Wenn ein ferner IBM Security Directory Integrator-Server ohne SSL (also mit der Eigenschaft `api.remote.ssl.on=false`) konfiguriert ist, kommt der AMC-Schlüsselspeicher oder -Truststore auch dann nicht zum Einsatz, wenn er korrekt konfiguriert ist, da nicht versucht wird, eine SSL-Verbindung aufzubauen. In diesem Fall muss die IP-Adresse des Computers mit dem AMC-Server beim IBM Security Directory Integrator-Server registriert sein. Hierzu wird die Datei `global.properties` (oder `solution.properties`) bearbeitet. Die zu aktualisierende Eigenschaft heißt **`api.remote.nonssl.hosts`**. Sobald die IP-Adresse des AMC-Computers in der Datei "global.properties" des fernen IBM Security Directory Integrator-Servers eingegeben wurde, kann AMC eine Verbindung zu diesem Server herstellen. Diesem Umstand liegt die Voraussetzung zugrunde, dass ferne Serververbindungen (AMC-Verbindungen) nur von solchen Computern akzeptiert werden, deren IP-Adressen in der Eigenschaft **`api.remote.nonssl.hosts`** angegeben sind.

Anmerkung: Falls der IBM Security Directory Integrator-Server auf demselben Computer wie AMC ausgeführt wird, ist eine Bearbeitung dieser Eigenschaft nicht erforderlich.

Wenn ein ferner IBM Security Directory Integrator-Server für SSL (also mit der Eigenschaft **`api.remote.ssl.on=true`**) konfiguriert ist, müssen der SSL-Schlüsselspeicher und -Truststore für AMC entsprechend konfiguriert sein.

Details zu diesem Aspekt enthält der vorherige Abschnitt über AMC und SSL. Ein ferner IBM Security Directory Integrator-Server kann nicht nur mit oder ohne SSL konfiguriert sein, sondern auch eine angepasste Authentifizierung erfordern. Hierbei müssen ein Benutzername und ein Kennwort übergeben werden, wenn die Verbindung zum fernen IBM Security Directory Integrator-Server hergestellt wird. Der ferne IBM Security Directory Integrator-Server wertet diesen Benutzernamen und dieses Kennwort anhand eines Drittanbieterrepositorys (z. B. LDAP, Datei, Datenbank, Script usw.) aus und entscheidet dann, ob der Client der Server-API eine Verbindung herstellen darf oder nicht. In diesen Fällen wählen Sie zum Registrieren eines Servers bei AMC (**Server -> Server modifizieren**) im Fenster "Authentifizierungsmodus" die Option **Angepasste oder LDAP-Authentifizierung** aus und geben Sie die Werte für den Benutzernamen und das Kennwort ein, die AMC bei jedem Versuch übergeben muss, eine Verbindung zum angegebenen fernen IBM Security Directory Integrator-Server herzustellen.

Anmerkung: Falls der Benutzername oder das Kennwort (bei der angepassten Authentifizierung) bzw. die SSL-Schlüsselspeicher oder -Truststores (bei Verwendung von SSL) nicht ordnungsgemäß konfiguriert sind, kann AMC keine Verbindung zum fernen IBM Security Directory Integrator-Server herstellen und für den Server wird der Status "Gestoppt" oder "Nicht aktiv" angezeigt.

AMC und Rollenverwaltung

Jedem Benutzer (bzw. jeder Gruppe) kann in AMC eine Rolle für eine bestimmte Lösungsansicht zugeordnet werden. Nachstehend erhalten Sie Informationen zu den verfügbaren Rollen und deren Bedeutung.

Diese Rollenzuordnung erfolgt im Fenster **Lösungsansichten**. Wählen Sie dort eine bestimmte Lösungsansicht und dann die Option **ACLs konfigurieren** aus. Daraufhin wird das Fenster **ACLs konfigurieren** angezeigt. Wählen Sie den Namen des Benutzers aus, den Sie konfigurieren wollen, und klicken Sie in der Funktionsleiste auf **Benutzer konfigurieren**. Das Fenster **Benutzer konfigurieren** wird aufgerufen. Wählen Sie die **Benutzer-ID** und dann eine der verfügbaren Rollen aus:

- Lesen
- Ausführen
- Administrator
- Konfigurationsadministrator

Anmerkung: Lösungsansichten, die mit der Option für die automatische Aktualisierung erstellt wurden, müssen erneut geladen werden. Verwenden Sie die Option **Neuanzeige der Lösungsansicht** im Fenster **Lösungsansichten**. Bei Lösungsansichten, die für die automatische Aktualisierung gekennzeichnet sind, müssen Sie die Konfigurationsdatei erneut laden und die Lösungsansicht durch Klicken auf **Neuanzeige der Lösungsansicht** aktualisieren. Falls ein Benutzer für eine Lösungsansicht, die mit der Option **Einfach** erstellt und für die automatische Aktualisierung gekennzeichnet wurde, keine Neuanzeige aufruft, verursacht die Lösungsansicht möglicherweise Inkonsistenzen in der AMC-Datenbank. Inkonsistenzen in Lösungsansichten, die nicht aktualisiert werden, können zu einem Fehlverhalten von Action Manager führen.

Die Rollen sind in der nachfolgenden Auflistung mit aufsteigender Berechtigung angegeben, die Rolle "Konfigurationsadministrator" besitzt demnach die höchste Berechtigung, die Rolle "Lesen" die geringste. Die gesamte Funktionalität, die für einen Benutzer mit der Rolle "Lesen" für eine Lösungsansicht zur Verfügung steht, ist definitiv für einen Benutzer verfügbar, der die Berechtigung "Ausführen" für diese Lösungsansicht besitzt. Die gesamte Funktionalität, die für einen Benutzer mit der Rolle "Ausführen" für eine Lösungsansicht zur Verfügung steht, ist für einen Benutzer verfügbar, der die Berechtigung "Administrator" besitzt, usw.

Die einzelnen Rollen haben die folgende Bedeutung:

Lesen Diese Rolle bedeutet, dass der Benutzer lediglich die Details der Lösungsansicht lesen kann, beispielsweise die in dieser Ansicht enthaltenen Fertigungslinien oder Eigenschaften, den Status der Fertigungslinien usw. Der Benutzer ist nicht in der Lage, ein Detail dieser Lösungsansicht zu modifizieren, zu starten, zu stoppen oder zu ändern.

Ausführen

Diese Rolle ist im Grunde genommen mit der Rolle "Lesen" identisch, stellt jedoch eine zusätzliche Berechtigung bereit, nämlich die Fähigkeit, Fertigungslinien zu starten und zu stoppen.

Administrator

Ein Benutzer mit dieser Rolle kann die Lösungsansicht verwalten, jedoch nicht modifizieren. Er kann alle Funktionen ausführen, die einem Benutzer mit der Rolle "Ausführen" zur Verfügung stehen. Zusätzlich ist er in der Lage, für diese Lösungsansicht Eigenschaften zu modifizieren, Protokolle zu löschen, Regeln zu konfigurieren usw.

Konfigurationsadministrator

Ein Benutzer mit dieser Rolle kann praktisch jede beliebige Operation für die Lösungsansicht ausführen. Hierzu gehört auch das Modifizieren der Ansicht selbst, das Modifizieren der Berechtigungen anderer Benutzer für diese Ansicht usw. Dies ist die höchste Berechtigung, die einem Benutzer für eine bestimmte Lösungsansicht erteilt werden kann.

Die obigen Rollen können auch Gruppen zugeordnet werden. Wenn beispielsweise die Benutzer "test" und "tdi" zur Gruppe "DBAdmin" gehören und der Gruppe "DBAdmin" die Berechtigung "Konfigurationsadministrator" für die Lösungsansicht "SynchDatabase" erteilt wird, erhalten daher beide Benutzer ("test" und "tdi") automatisch die Berechtigung "Konfigurationsadministrator" für die Lösungsansicht "SynchDatabase".

Anmerkung:

1. Wird dem Benutzer "test" explizit die Berechtigung "Lesen" für dieselbe Lösungsansicht erteilt, hat die Berechtigung "Lesen" Vorrang vor der Berechtigung, die der Benutzer als Mitglied der Gruppe "DBAdmin" erhält. Dies geschieht, damit eine spezifische Rollenzuordnung Priorität vor einer Rollenzuordnung über Gruppen hat. Auf diese Weise kann Einzelpersonen ein eingeschränkterer oder umfassenderer Zugriff gewährt werden, ohne dass hierbei ein übernommener Zugriff aus bestimmten Gruppen berücksichtigt werden muss.
2. Falls der Benutzer "test" zu zwei Gruppen gehört, von denen die Gruppe 1 die Berechtigung "Lesen" und die Gruppe 2 die Berechtigung "Administrator" für dieselbe Lösungsansicht besitzt, erhält der Benutzer "test" in dieser Situation die höhere der beiden Berechtigungen (hier "Administrator"), sofern ihm für dieselbe Lösungsansicht keine spezifische Rolle zugeordnet ist (in diesem Fall hat - wie oben unter Punkt 1 erläutert - die spezifische Rolle des Benutzers "test" Vorrang).

AMC und Kennwörter

Nachstehend erhalten Sie Informationen zum Speichern von Kennwörtern.

Alle Kennwortfelder, die in der Datei `amc.properties` gespeichert sind (z. B. Kennwort für die LDAP-Bindung, Schlüsselspeicherkennwort usw.) werden verschlüsselt, bevor sie in die Datei `amc.properties` geschrieben werden. Zudem zeigt AMC Kennwortfelder oder geschützte Felder nie in der Konsole an. Alle derartigen Felder werden maskiert.

AMC und verschlüsselte Konfigurationen

Nachstehend erhalten Sie Informationen zu kennwortgeschützten Konfigurationen.

AMC ermöglicht es Benutzern, kennwortgeschützte Konfigurationen zu laden und Verbindungen zu ihnen herzustellen. Im AMC-Fenster "Konfigurationen laden/erneut laden" gibt es ein Kennworttextfeld. In diesem Feld müssen Benutzer das Kennwort der Konfiguration eingeben, die sie starten wollen, bevor sie auf die Schaltfläche **Starten** klicken. In der Action Manager-Anzeige wird für die Aktion "Fertigungslinie starten" entsprechend ein Kennwortfeld bereitgestellt, in dem der Benutzer das Kennwort der Konfiguration eingeben kann. Action Manager übergibt dieses Kennwort bei dem Versuch, die Konfiguration zu starten.

Anmerkung: AMC kann nicht feststellen, ob die gestartete ferne Konfiguration durch ein Kennwort geschützt ist. Aus diesem Grund wird bei fehlender oder fal-

scher Angabe des Kennworts für den Benutzer lediglich die Fehlermeldung "Die Konfiguration konnte nicht gestartet werden" ausgegeben. Der Benutzer kann jedoch die IBM Security Directory Integrator-Serverprotokolle anzeigen, die eine genaue Nachricht enthalten.

AMC-Benutzerschnittstelle

Nachstehend erhalten Sie detaillierte Informationen zur Benutzerschnittstelle von AMC.

Bei der Konsole an- und abmelden

Mit den hier aufgeführten Anweisungen können Sie sich an der Konsole anmelden und davon abmelden.

Öffnen Sie einen Web-Browser und geben Sie die folgende Adresse ein:

```
http://hostname:port/ibm/console
```

Hierbei steht *port* für den Port, an dem Ihr Web-Server ausgeführt wird. Bei einer Implementierung in dem Web-Container, der im Produktpaket enthalten ist, wird standardmäßig Port 13100 für die HTTP-Kommunikation und Port 13101 für die HTTPS-Kommunikation verwendet.

Die Anmeldeseite kann auch unter Verwendung der Datei `launchAMC.html` gestartet werden, die sich im Ordner `tdi-installationsverzeichnis/bin/amc` befindet.

Das Fenster mit der Anmeldeseite für die IBM Security Directory Integrator-Komponente "Administration and Monitoring Console" wird angezeigt.

Bei der Konsole als Konsolenadministrator anmelden

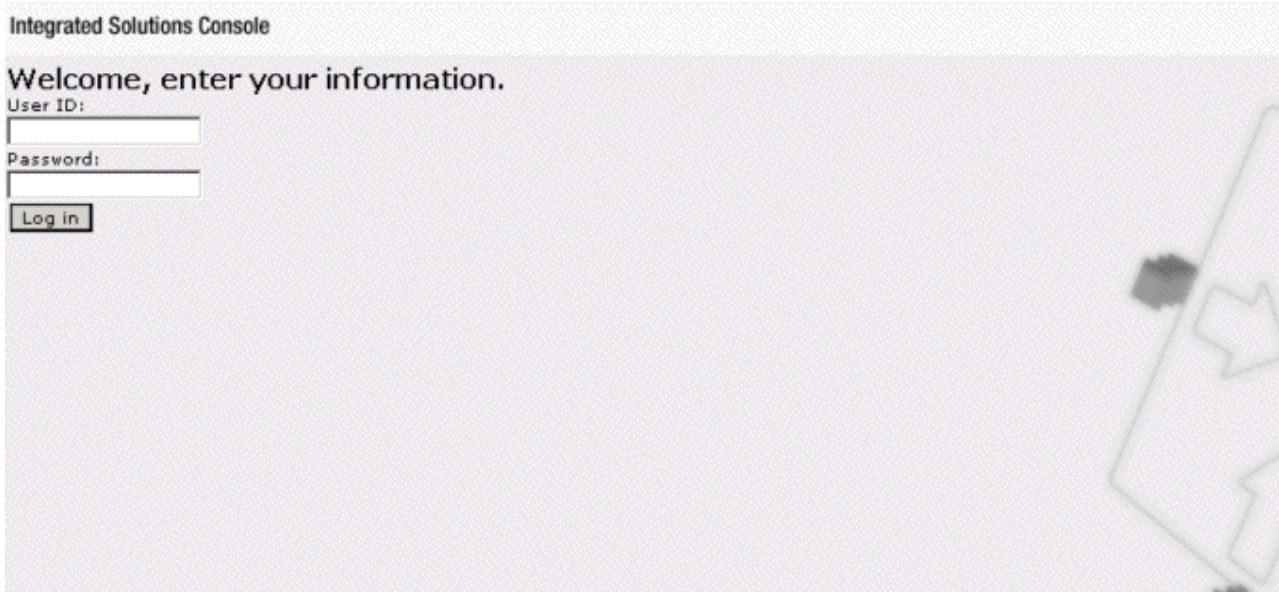
Der Konsolenadministrator ist ein Benutzer, der Folgendes ausführen kann:

- Erforderliche Eigenschaften für AMC konfigurieren
- Authentifizierungsmechanismus für AMC-Anmeldungen festlegen
- Neue Benutzer hinzufügen und Benutzerrollen konfigurieren

Verwenden Sie bei Ihrer erstmaligen Anmeldung den Systembenutzernamen und das Systemkennwort, mit denen Sie IBM Security Directory Integrator, AMC und die integrierte Webplattform installiert haben. Falls Sie AMC in IBM WebSphere Application Server/IBM Dashboard Application Services Hub implementiert haben, müssen Sie sich als ein Benutzer anmelden, dem die Rollen *iscadmins* und *administrator* zugeordnet sind.

Anmerkung: Die integrierte Webplattform verwendet auf UNIX- und Linux-Systemen den PAM-Authentifizierungsmechanismus, um die bei der Anmeldung angegebenen Werte für den Systembenutzernamen und das Systemkennwort auszuwerten. Aus diesem Grund müssen Sie auf AIX-Maschinen den Parameter `auth_type` in der Datei `/etc/security/login.cfg` auf den Wert `PAM_AUTH` setzen.

Geben Sie zur Anmeldung bei ISC Ihren Benutzernamen und Ihr Kennwort in die entsprechenden Felder des Anmeldefensters ein und klicken Sie auf die Schaltfläche **Anmelden**.



Die Schaltfläche **Abmelden** befindet sich neben der Schaltfläche **Hilfe** in der rechten oberen Ecke der Konsole. Wenn Sie auf **Abmelden** klicken, werden Sie zur Anmeldeseite zurückgeführt.

Layout der AMC-Konsole

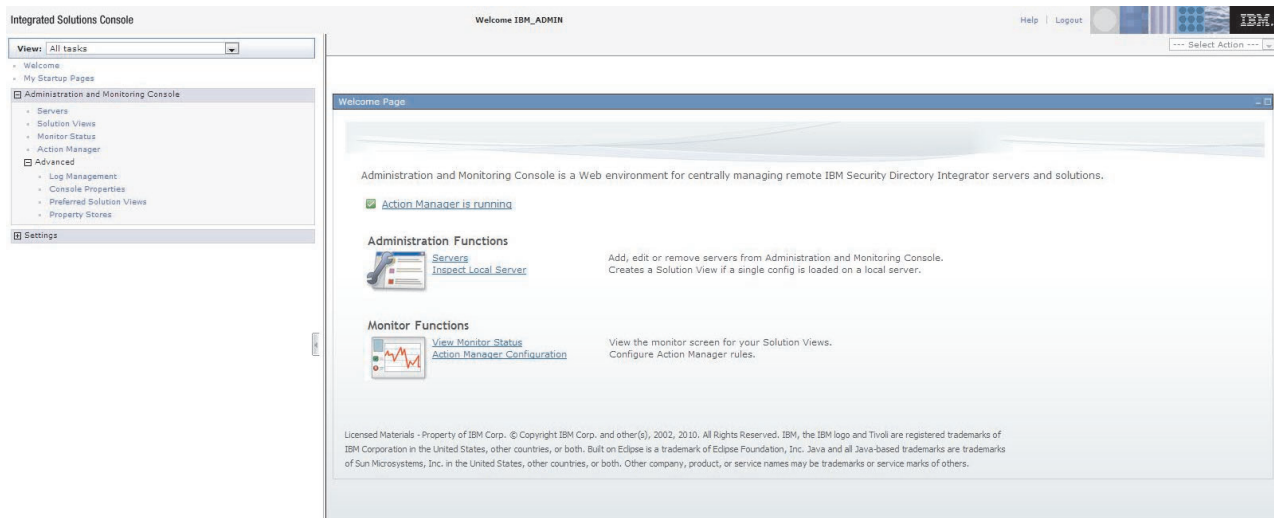
Sie können die aufgeführten Komponenten von IBM Security Directory Integrator Administration and Monitoring Console verwenden.

Navigationbereich

Der Navigationbereich enthält eine Baumstrukturansicht, mit deren Hilfe die Benutzer in den Tasks navigieren können, die für sie in der Konsole verfügbar sind. Im Navigationbereich können Sie Ordner öffnen und schließen sowie Tasks (dies sind keine Ordner) auswählen, die im Arbeitsbereich des Konsolenframeworks gestartet werden sollen.

Arbeitsbereich

Der Arbeitsbereich enthält die Informationen und Eingabefelder, die für die Ausführung der gegenwärtig bearbeiteten Task benötigt werden.



Bei der Konsole abmelden

Um sich bei der Konsole abzumelden, klicken Sie im Navigationsbereich auf **Abmelden**.

AMC-Tabellen verwenden

Mithilfe der Informationen in AMC-Tabellen können Sie nach diesen Tabelleneinträgen suchen, die Einträge verwalten und Aktionen für sie ausführen.

Die IBM Security Directory Integrator-Komponente "Administration and Monitoring Console" zeigt bestimmte Informationen (z. B. Listen mit Attributen und Einträgen) in Form von Tabellen an.

Die Tabellen der IBM Security Directory Integrator-Komponente "Administration and Monitoring Console" bieten Symbole, die Ihnen beim Verwalten und Auffinden von Informationen in der Tabelle helfen. Einige Symbole werden nur in bestimmten Tabellen angezeigt. Dies ist von der aktuellen Task abhängig. Die folgende Liste enthält alle Symbole, die möglicherweise angezeigt werden:

- Wenn Sie auf das Symbol **Filterzeile anzeigen** klicken, werden für jede Spalte in der Tabelle Filterzeilen angezeigt. Weitere Informationen zur Filterung enthält der Abschnitt „Filtern“ auf Seite 293.
- Wenn Sie auf das Symbol **Filterzeile ausblenden** klicken, werden für jede Spalte in der Tabelle Filterzeilen ausgeblendet. Weitere Informationen finden Sie im Abschnitt „Filtern“ auf Seite 293.
- Wenn Sie auf das Symbol **Alle Filter löschen** klicken, werden alle für die Tabelle definierten Filter gelöscht. Weitere Informationen finden Sie im Abschnitt „Filtern“ auf Seite 293.
- Wenn Sie auf das Symbol **Sortierung bearbeiten** klicken, werden die Informationen in der Tabelle sortiert. Weitere Informationen finden Sie im Abschnitt „Sortieren“ auf Seite 292.
- Wenn Sie auf das Symbol **Alle Sortierungen löschen** klicken, werden alle für die Tabelle definierten Sortierungen gelöscht. Weitere Informationen finden Sie im Abschnitt „Sortieren“ auf Seite 292.
- Wenn Sie auf das Symbol **Tabelle komprimieren** klicken, werden die Tabellendaten ausgeblendet.

- Wenn Sie auf das Symbol **Tabelle erweitern** klicken, werden die Tabellendaten eingeblendet.
- Wenn Sie auf das Symbol **Alles auswählen** klicken, werden alle Tabelleneinträge ausgewählt.
- Wenn Sie auf das Symbol **Alles abwählen** klicken, werden alle ausgewählten Tabelleneinträge abgewählt.
- Wenn Sie auf das Symbol **Exportieren** klicken, werden die Tabellendaten exportiert.

Dropdown-Menü "Aktion auswählen"

Über das Dropdown-Menü **Aktion auswählen** können Sie eine vollständige Liste aller Optionen anzeigen, die für eine ausgewählte Tabelle verfügbar sind, und Operationen für den Tabelleninhalt durchführen.

Informationen zu diesem Vorgang

Beispielsweise können Sie das Dropdown-Menü **Aktion auswählen** anstelle der Symbole zum Anzeigen und Ausblenden von Sortierungen und Filtern verwenden. Über das Dropdown-Menü **Aktion auswählen** können Sie auch Operationen für den Tabelleninhalt ausführen. Im Fenster **Attribute verwalten** werden beispielsweise Aktionen wie **Anzeigen**, **Hinzufügen**, **Bearbeiten**, **Kopieren** und **Löschen** nicht nur als Schaltflächen in der Funktionsleiste, sondern auch als Optionen im Dropdown-Menü **Aktion auswählen** angezeigt. Falls dies von der Tabelle unterstützt wird, können Sie die Option zum Anzeigen der Suchfunktionsleiste ebenfalls über das Dropdown-Menü **Aktion auswählen** ein- oder ausblenden. Weitere Informationen zum Suchen nach Tabelleneinträgen finden Sie im Abschnitt "Suchen".

So führen Sie eine Aktion über das Menü "Aktion ausführen" aus:

1. Wählen Sie erforderlichenfalls einen Eintrag in der Tabelle aus.
2. Klicken Sie auf das Dropdown-Menü **Aktion auswählen**.
3. Wählen Sie die Aktion aus, die ausgeführt werden soll, beispielsweise **Server beenden**.
4. Klicken Sie auf **Los**.

Blättern

Zum Anzeigen der verschiedenen Tabellenseiten verwenden Sie die Navigationssteuerelemente am unteren Rand der Tabelle.

Sie können im Navigationsfeld eine bestimmte Seitenzahl eingeben und auf **Los** klicken, um eine bestimmte Seite anzuzeigen. Mit den Pfeilschaltflächen **Weiter** und **Zurück** können Sie außerdem von Seite zu Seite blättern.

Sortieren

Sie können die Art der Sortierung für die Einträge in einer Tabelle ändern.

Informationen zu diesem Vorgang

1. Führen Sie eine der folgenden Aktionen aus:
 - Klicken Sie auf das Symbol **Sortierung bearbeiten** für die Tabelle.
 - Klicken Sie auf das Dropdown-Menü **Aktion auswählen**, wählen Sie die Option **Sortierung bearbeiten** aus und klicken Sie auf **Los**.

Für jede Spalte in der Tabelle wird ein Dropdown-Menü zur Sortierung angezeigt.

2. Wählen Sie im ersten Dropdown-Menü zur Sortierung die Spalte aus, nach der die Sortierung vorgenommen werden soll. Verfahren Sie genauso für alle anderen sortierbaren Spalten, die Sie sortieren möchten.
3. Wählen Sie über die Optionen **Aufsteigend/Absteigend** im Dropdown-Menü aus, ob die Sortierung in aufsteigender oder absteigender Reihenfolge vorgenommen werden soll. Die Standardsortierreihenfolge ist "Aufsteigend". Zur Sortierung können Sie auch die Spaltenüberschriften verwenden. Für jede Spalte wird ein kleiner Pfeil angezeigt. Ein Aufwärtspfeil bedeutet, dass die Spalte aufsteigend sortiert ist. Ein Abwärtspfeil bedeutet, dass die Spalte in absteigender Reihenfolge sortiert ist. Um die Sortierreihenfolge zu ändern, klicken Sie einfach auf die Spaltenüberschrift.
4. Wenn die Sortierung vorgenommen werden soll, klicken Sie auf **Sortieren**.

Um alle Sortierungen zu löschen, klicken Sie auf das Symbol **Alle Sortierungen löschen**.

Suchen

Sie können in einer Tabelle nach einem oder mehreren bestimmten Einträgen suchen.

Informationen zu diesem Vorgang

Anmerkung: Die Option zum Anzeigen der Suchfunktionsleiste ist nur bei bestimmten Tabellen verfügbar. Dies richtet sich jeweils nach der aktuellen Task.

1. Wählen Sie im Dropdown-Menü **Aktion auswählen** die Option für das Anzeigen der Suchfunktionsleiste aus und klicken Sie auf **Los**.
2. Geben Sie Ihre Suchbedingungen im Feld **Suchen nach** ein.
3. Bei Bedarf können Sie für die Suche im Dropdown-Menü **Bedingungen** eine Bedingung auswählen. Dieses Menü bietet die folgenden Optionen:
 - **Enthält**
 - **Beginnt mit**
 - **Endet mit**
 - **Exakte Übereinstimmung**
4. Wählen Sie im Dropdown-Menü **Spalte** die Spalte aus, auf der die Suche basieren soll.
5. Wählen Sie im Dropdown-Menü **Richtung** aus, ob die Ergebnisse in absteigender oder aufsteigender Reihenfolge angezeigt werden sollen. Wählen Sie **Nach unten** aus, um die Ergebnisse in absteigender Reihenfolge anzuzeigen. Wählen Sie **Nach oben** aus, um die Ergebnisse in aufsteigender Reihenfolge anzuzeigen.
6. Wählen Sie die Option für die Beachtung der Groß-/Kleinschreibung aus, wenn die Suchergebnisse mit den im Feld **Suchen nach** angegebenen Bedingungen für Groß- und Kleinbuchstaben übereinstimmen sollen.
7. Klicken Sie nach Eingabe der gewünschten Bedingungen auf **Suchen**, um nach den Attributen zu suchen.

Filtern

Sie können Elemente in einer Tabelle filtern.

Informationen zu diesem Vorgang

1. Führen Sie eine der folgenden Aktionen aus:

- Klicken Sie auf das Symbol **Filterzeile anzeigen**. Klicken Sie auf das Dropdown-Menü **Aktion auswählen**, wählen Sie die Option **Filterzeile anzeigen** aus und klicken Sie auf **Los**.
2. Über jeder Spalte werden Filterschaltflächen angezeigt. Klicken Sie über der Spalte, die Sie filtern wollen, auf die Schaltfläche **Filtern**.
 3. Wählen Sie im Dropdown-Menü **Bedingungen** eine oder mehrere der folgenden Bedingungen aus:
 - Enthält
 - Beginnt mit
 - Endet mit
 4. Geben Sie im Feld den Text ein, nach dem die Filterung vorgenommen werden soll. Falls Sie beispielsweise die Bedingung **Beginnt mit** ausgewählt haben, könnten Sie **C** eingeben.
 5. Wenn die Groß- und Kleinschreibung (Text in Großbuchstaben oder Text in Kleinbuchstaben) abgeglichen werden soll, wählen Sie die Option für die Beachtung der Groß-/Kleinschreibung aus.
 6. Klicken Sie auf **OK**, wenn Sie zum Filtern der Attribute bereit sind.
 7. Wiederholen Sie die obigen Schritte 2 bis 6 für jede Spalte, die gefiltert werden soll.

Um alle Filter zu löschen, klicken Sie auf das Symbol **Alle Filter löschen**.

Um die Filterzeilen auszublenden, klicken Sie erneut auf das Symbol **Filterzeile anzeigen**.

Server

In diesem Fenster können Sie den registrierten Server anzeigen. Darüber hinaus kann ein Konsolenadministrator in diesem Fenster IBM Security Directory Integrator-Server hinzufügen, bearbeiten, löschen und beenden sowie das Fenster "Konfigurationsdateien" starten.

Beim Starten von AMC wird automatisch ein lokaler IBM Security Directory Integrator-Server verwendet, der an Port 1099 registriert ist. Daher ist im Fenster **Server** unter "Lokaler Server" bereits ein Eintrag vorhanden, dessen Status je nach Zustand als "Aktiv" oder "Nicht verfügbar" angegeben ist.

Wählen Sie zum Laden oder erneuten Laden einer Konfiguration **Server** aus und klicken Sie in der Funktionsleiste des Fensters "Server" auf **Konfigurationsdateien**. Daraufhin wird das Fenster **Konfigurationsdateien** aufgerufen.

Sie können die Operationen, die Sie ausführen wollen, in der Funktionsleiste über der Tabelle oder aber im Dropdown-Menü "Aktion auswählen" auswählen. Hierzu zählen beispielsweise die folgenden Aktionen:

Hinzufügen

Klicken Sie in der Funktionsleiste auf "Hinzufügen".

Delete Wählen Sie das Optionsfeld neben dem Server aus, den Sie löschen wollen, und klicken Sie in der Funktionsleiste auf **Löschen**.

Modifizieren

Wählen Sie den Server aus, dessen Informationen Sie modifizieren wollen, und klicken Sie in der Funktionsleiste auf **Modifizieren**.

Konfigurationsdateien

Wählen Sie den Server aus, dessen Konfigurationsdateien Sie auflisten wol-

len. Sobald Sie im Fenster "Lösungsansichten" auf den Link **Konfigurationsdateien anzeigen** klicken, wird das Fenster "Konfigurationsdateien" aufgerufen. Jede Konfigurationsdatei ist mit der Bezeichnung "Geladen" oder "Nicht geladen" versehen. Die Funktionsleiste bietet eine Reihe von Optionen für das Laden, Entladen und erneute Laden.

Server beenden

Wählen Sie den Server aus, den Sie beenden wollen, und klicken Sie in der Funktionsleiste auf "Server beenden".

Ordnungsgemäß beenden

Diese Option beendet einen Server ordnungsgemäß (hierzu werden neue Threads erstellt, die das Stoppen der Fertigungslinien abwarten).

Anmerkung: Die ordnungsgemäße Beendigung wird bei IBM Security Directory Integrator-Servern aus älteren Versionen als Version 7.1 nicht unterstützt.

Server hinzufügen

Sie können einen IBM Security Directory Integrator-Server zu AMC hinzufügen.

Informationen zu diesem Vorgang

Nachdem Sie einen IBM Security Directory Integrator-Server zu AMC hinzugefügt haben, können Sie die Funktionen in anderen AMC-Fenstern verwenden, um Lösungsansichten zum IBM Security Directory Integrator-Server hinzuzufügen und um Ansichten für die Lösungsansichten zu erstellen und zu definieren, die dem IBM Security Directory Integrator-Server zugeordnet sind.

So fügen Sie einen neuen IBM Security Directory Integrator-Server hinzu:

1. Geben Sie im Feld **Name** einen Namen für den IBM Security Directory Integrator-Server ein.
2. Geben Sie im Feld **Hostname** den Hostnamen oder die IP-Adresse des Computers ein, auf dem IBM Security Directory Integrator ausgeführt wird.
3. Geben Sie die Portnummer ein, an der der IBM Security Directory Integrator-Server konfigurationsgemäß ausgeführt wird.
4. Wählen Sie den gewünschten Authentifizierungsmodus aus. Wenn Sie die LDAP-Authentifizierungsmethode oder die angepasste Authentifizierungsmethode auswählen, geben Sie den Benutzernamen und das Kennwort ein, die für die Authentifizierung verwendet werden sollen.
5. Klicken Sie auf **OK**.

Server modifizieren

Sie können die Informationen für einen vorhandenen IBM Security Directory Integrator-Server bearbeiten.

Informationen zu diesem Vorgang

So bearbeiten Sie einen vorhandenen Server:

1. Achten Sie zunächst auf die angezeigte Server-ID. Falls Sie die Server-ID ändern wollen, klicken Sie auf **Server-ID ändern**.
2. Geben Sie den Namen für den Server ein.
3. Geben Sie im Feld **Hostname** den Hostnamen oder die IP-Adresse des Computers ein, auf dem der IBM Security Directory Integrator-Server ausgeführt wird.

4. Geben Sie die Portnummer ein, an der der IBM Security Directory Integrator-Server konfigurationsgemäß ausgeführt wird.
5. Wählen Sie den gewünschten Authentifizierungsmodus aus. Wenn Sie die LDAP-Authentifizierungsmethode oder die angepasste Authentifizierungsmethode auswählen, geben Sie den Benutzernamen und das Kennwort ein, die für die Authentifizierung verwendet werden sollen.
6. Klicken Sie auf "Abbrechen", wenn Sie das Fenster verlassen wollen, ohne Änderungen vorzunehmen, oder klicken Sie auf **OK**, um die Änderungen zu speichern.
7. Klicken Sie auf **Verbindung testen**, um zu überprüfen, ob die Verbindung zum Server mit den aktuellen Einstellungen hergestellt werden kann.

Konsoleigenschaften

Verwenden Sie das Fenster "Konsoleigenschaften" von AMC, um die Konfigurationen wie z. B. die Datenbankkonfiguration von AMC, SSL-Einstellungen, die Rotationshäufigkeit von Action Manager-Protokollen etc. zu verwalten.

Allgemein

Verwenden Sie das Fenster "Allgemein" von AMC, um die Rotationshäufigkeit von Action Manager-Protokollen in Tagen festzulegen.

Konfiguration mit SSL

Verwenden Sie das Fenster "SSL", um die SSL-Einstellungen für AMC zu konfigurieren. Die SSL-Einstellungen gelten für SSL-Verbindungen, die von AMC zum fernen IBM Security Directory Integrator-Server hergestellt werden. Zugänglich gemacht werden lediglich die SSL-Eigenschaften für den Schlüsselspeicher und den Truststore von AMC. Falls SSL auf dem fernen Server aktiviert ist, muss ein Administrator sicherstellen, dass das erforderliche Zertifikat in dessen Speicher importiert wurde, damit die Verbindung ordnungsgemäß ausgeführt wird. Ein Administrator sollte für jeden fernen Server, zu dem er eine Verbindung herstellen will, das Zertifikat des fernen Servers in seinen Speicher importieren.

JDBC-Eigenschaften

JDBC-Eigenschaften werden dazu verwendet, die Verbindungseinstellungen für die Derby-Datenbank bzw. für andere mit AMC kompatible Datenbanken wie Oracle und MS SQL Server zu definieren. In der AMC-Datenbank werden die Konfigurationen, Verbindungsdetails, Action Manager-Regeln und Ergebnisse für AMC gespeichert.

IBM Security Directory Integrator AMC unterstützt neben Derby auch andere Datenbanken. Im Produktpaket von AMC ist die Derby-Datenbank enthalten. AMC kommuniziert mit der eigenen Datenbank über das JDBC-Protokoll. JDBC ist ein generisches Protokoll, das ohne großen Aufwand auf andere Datenbanken erweitert werden kann. Dank der AMC-Unterstützung für alternative Datenbanken können Sie AMC installieren und mit einer vorhandenen Datenbank kommunizieren lassen. In der Datenbank werden Action Manager-Protokolle, Ergebnisse usw. gespeichert. Im ISC-Abschnitt **Erweitert** -> **Konsoleigenschaften** sind die **JDBC-Eigenschaften** für Derby oder eine andere Datenbank zusammengefasst. Bei Derby können Sie die Datenbank für die Ausführung im integrierten Modus oder im Netzmodus konfigurieren. Der Standarddatenbank ist die Derby-Datenbank, der Standardmodus ist der Netzmodus.

In diesem Fenster können Sie die folgenden Aktionen ausführen:

- Wählen Sie im Feld **Datenbanktyp** eine Datenbank aus. Zur Auswahl stehen Derby, MS SQL Server, Oracle und DB2.
- Geben Sie im Feld **JDBC-URL** einen Wert für die JDBC-URL ein.
- Geben Sie im Feld **Benutzername** den Benutzernamen für die Datenbank ein.
- Geben Sie im Feld **Kennwort** das Kennwort für die Datenbank ein.
- Geben Sie im Feld **JDBC-Treiber** den Namen des JDBC-Treibers ein.

Die folgende Tabelle enthält einige Richtlinien für die Parameter "JDBC-URL" und "JDBC-Treiber":

Tabelle 31. Treiberparameter

Datenbank	JDBC-URL	JDBC-Treiber	JAR-Datei des Treibers
Derby	jdbc:derby://host:port/database [create=true create=false]	org.apache.derby.jdbc.ClientDriver	derby.jar
MS SQL Server (2005)	jdbc:sqlserver://host:port; databasename=database	com.microsoft.sqlserver.jdbc.SQLServerDriver	sqljdbc.jar
Oracle	jdbc:oracle:thin:@host:port:database	oracle.jdbc.driver.OracleDriver	ojdbc14.jar
DB2	jdbc:db2://host:port/database	com.ibm.db2.jcc.DB2Driver	db2jcc.jar

Anmerkung:

1. Abhängig von der ausgewählten Datenbank muss die JAR-Datei des entsprechenden Treibers in das Verzeichnis *tdi-installationsverzeichnis/1wi/1libs* kopiert werden.
2. Die Konfiguration von Action Manager ist ebenfalls erforderlich, damit die neue Datenbank angegeben ist, die für die Ausführung von Action Manager verwendet wird. Dieselbe JAR-Datei muss zum Verzeichnis *tdi-installationsverzeichnis/bin/amc/ActionManager/jars* hinzugefügt und die Datei *am_config.properties* muss entsprechend angepasst werden.
3. Falls Sie anstelle von Derby eine der Alternativen verwenden wollen, müssen Sie daran denken, dass die in der JDBC-URL angegebene Datenbank vor dem Starten von AMC bereits vorhanden sein muss (andernfalls ist AMC nicht in der Lage, eine Datenbank zu erstellen und zu füllen). Falls Sie Derby verwenden, ist dies nicht erforderlich, weil diese Datenbank die Option "create=true" in der JDBC-URL unterstützt. Diese Option bewirkt, dass AMC die Datenbank beim Start automatisch erstellt, falls die Datenbank nicht vorhanden ist.

Lösungsansichten

Im Fenster "Lösungsansichten" können Sie Lösungsansichten anzeigen, hinzufügen, modifizieren und löschen.

- Klicken Sie zum Hinzufügen einer Lösungsansicht in der Funktionsleiste auf die Schaltfläche **Hinzufügen**.
- Wählen Sie zum Modifizieren einer vorhandenen Lösungsansicht die gewünschte Lösungsansicht aus und klicken Sie auf **Modifizieren**. Führen Sie anschließend die Schritte im **Modifizierungsassistenten** aus. Klicken Sie unter **Lösungsansicht modifizieren** auf **Weiter**, um mit dem nächsten Schritt fortzufahren. Nachdem Sie alle Schritte ausgeführt haben, klicken Sie auf **Fertigstellen**.
- Um die Zugriffssteuerungslisten (Access Control Lists - ACLs) für eine Lösungsansicht zu konfigurieren, wählen Sie die Lösungsansicht aus, deren ACLs Sie konfigurieren wollen, und wählen Sie in der Funktionsleiste die Option **ACLs konfigurieren...** aus.

- Wählen Sie zum Löschen einer vorhandenen Lösungsansicht die zu löschende Lösungsansicht aus und klicken Sie in der Funktionsleiste auf die Schaltfläche **Löschen**.
- Klicken Sie zum Starten einer separaten Anzeige, in der Sie die lokalen Action Manager-Variablen für diese Lösungsansicht hinzufügen, bearbeiten oder modifizieren können, auf **Lokale Variablen...**

Anmerkung: Lösungsansichten, die mit der Option für die automatische Aktualisierung erstellt wurden, müssen erneut geladen werden.

Wenn Sie eine **Lösungsansicht modifizieren**, überprüft AMC, ob die Lösungsansicht mit **automatischer Aktualisierung** erstellt wurde. Falls die für die Modifizierung ausgewählte Lösungsansicht mit Verwendung der automatischen Aktualisierung erstellt wurde, wird die folgende Nachricht ausgegeben.

Die ausgewählte Lösungsansicht ist für die automatische Aktualisierung markiert. Stellen Sie sicher, dass die automatische Aktualisierung inaktiviert ist, wenn Sie die Lösungsansicht modifizieren möchten.

Die Lösungsansichten werden in der Tabelle "Lösungsansichten" aufgelistet. Falls eine bestimmte Lösungsansicht mit Verwendung der automatischen Aktualisierung erstellt wurde, wird ein Kurzmenü >> angezeigt, wenn Sie auf den Aufwärts- und Rechtspfeil neben dem Namen der Lösungsansicht klicken. Sie können die Optionen **Neuanzeige der Lösungsansicht** oder **Automatische Aktualisierung inaktivieren** auswählen. Bei Lösungsansichten, die für die automatische Aktualisierung gekennzeichnet sind, müssen Sie die Konfigurationsdatei erneut laden und die Lösungsansicht durch Klicken auf **Neuanzeige der Lösungsansicht** aktualisieren. Falls ein Benutzer für eine Lösungsansicht, die mit der Option **Einfach** erstellt und für die automatische Aktualisierung gekennzeichnet wurde, keine Neuanzeige aufruft, verursacht die Lösungsansicht möglicherweise Inkonsistenzen in der AMC-Datenbank. Inkonsistenzen in Lösungsansichten, die nicht aktualisiert werden, können zu einem Fehlverhalten von Action Manager führen.

ACLs konfigurieren

Sie können die Zugriffssteuerungslisten (Access Control Lists - ACLs) für einen Benutzer festlegen und diesen Benutzer einer bestimmten Lösungsansicht zuordnen.

Informationen zu diesem Vorgang

- Um einen oder mehrere Benutzer zu konfigurieren, wählen Sie die gewünschten Benutzer aus und klicken Sie in der Funktionsleiste auf **Benutzer konfigurieren**.
 1. Wählen Sie den Benutzer, dem Sie eine Rolle zuordnen wollen, im Dropdown-Menü **Benutzer-ID** aus.
 2. Wählen Sie das Optionsfeld neben der Rolle bzw. den Rollen aus, die Sie dem ausgewählten Benutzer zuordnen wollen:
 - **Lesen:** Bei dieser Rolle darf der Benutzer Details der Lösungsansicht wie Fertigungslinien, Tombstones, Protokolle, Eigenschaften der Konfiguration usw. lesen.
 - **Ausführen:** Diese Rolle erlaubt dem Benutzer das Lesen sowie Starten und Stoppen von Fertigungslinien.
 - **Administrator:** Diese Rolle erteilt dem Benutzer die Berechtigungen der Rollen "Lesen" und "Ausführen". Zusätzlich ermöglicht sie ihm das Löschen von Protokollen und Tombstones.
 - **Konfigurationsadministrator:** Bei dieser Rolle ist der Benutzer berechtigt, eine Konfiguration zu starten oder zu stoppen, die Lösungsansicht zu modifizieren sowie ACLs für andere Benutzer zuzuordnen und zu modifizieren.

3. Klicken Sie auf **Anwenden**.

- Um einen vorhandenen Benutzer zu entfernen, wählen Sie den Benutzer in der Tabelle aus und klicken Sie auf **Entfernen**.

Nachdem Sie alle gewünschten Änderungen vorgenommen haben, klicken Sie auf **Anwenden**.

Lokale Variablen

Nachstehend erhalten Sie Informationen zum Arbeiten mit lokalen Variablen.

Wählen Sie im Navigationsfenster auf der linken Seite von AMC den Eintrag "Lösungsansichten" aus. Daraufhin wird das Fenster **Lösungsansichten** angezeigt. Wählen Sie in der Funktionsleiste die Option **Lokale Variablen** aus. Im Fenster **Lokale Variablen** können Sie lokale Variablen für eine Lösungsansicht auswählen und anschließend **hinzufügen**, **modifizieren** oder **löschen**.

Action Manager-Auslöser und -Aktionen müssen lokale Variablen unterstützen, die mithilfe von Regeln und Aktionen definiert oder erhöht werden können. Lokale Variablen können als Auslöserbedingungen für andere Regeln verwendet werden. Beispiel: Eine lokale Variable kann auf den Wert 1 gesetzt und dann bei jedem Vorkommen des Ereignisses und der lokalen Variablen erhöht werden (in diesem Beispiel wird die Zahl 1 für die Erhöhung bei jedem Vorkommen des Ereignisses festgelegt). Die lokale Variable kann die Regel "Fertigungslinie beenden" auslösen. Sie können eine neue Regel konfigurieren, die ausgelöst wird, sobald die Variable den Wert 10 erreicht. Diese neue Regel könnte eine neue Fertigungslinie auf einem anderen Server starten. Diese Action Manager-spezifischen lokalen Variablen werden für eine Lösungsansicht festgelegt. Dies bedeutet, dass eine Variable, die in einer zu einer Lösungsansicht gehörenden Regel erstellt wurde, nur in den Regeln dieser Lösungsansicht verwendet werden kann und für die Regeln anderer Lösungsansichten nicht zugänglich ist.

Lösungsansicht hinzufügen

Sie können Benutzern den Zugriff auf Informationen in der Konfigurationsdatei ermöglichen, ohne ihnen die Berechtigung zur direkten Bearbeitung der Konfigurationsdatei zu erteilen.

Informationen zu diesem Vorgang

Administratoren können mithilfe einer Lösungsansicht bestimmte Informationen aus einer Konfigurationsdatei herausfiltern, damit nur einige Informationen aus der Konfigurationsdatei angezeigt werden. Sie können für jede Konfiguration mehrere Lösungsansichten erstellen, die jeweils andere in der Konfigurationsdatei enthaltene Informationen offenlegen.

Wählen Sie zum Hinzufügen einer Lösungsansicht die Option **Lösungsansicht** und anschließend in der Funktionsleiste des Fensters **Lösungsansichten** die Option **Hinzufügen** aus.

1. Geben Sie Details der Ansicht ein:
 - a. Geben Sie im Feld **Name** einen Namen für die Lösungsansicht ein.
 - b. Geben Sie im Feld **Beschreibung** eine Beschreibung der Lösungsansicht ein.
2. Wählen Sie den **Server** und die **Konfigurationen** (Konfigurationsdatei) aus, die Sie für die Erstellung einer Lösungsansicht verwenden wollen:
 - Wählen Sie im Menü **Server** den IBM Security Directory Integrator-Server aus, der die Konfigurationsdatei enthält, die Sie bei der Erstellung der Lö-

sungsansicht verwenden wollen. Falls keine IBM Security Directory Integrator-Server zu AMC hinzugefügt wurden, ist dieses Menü leer.

- Wählen Sie in der Liste **Konfigurationen** die Konfigurationsdatei aus, die Sie für die Erstellung der Lösungsansicht verwenden wollen. Dieses Menü enthält alle gegenwärtig geladenen Konfigurationen.

Anmerkung: Klicken Sie auf die Schaltfläche **Konfigurationsdateien anzeigen**, um das Fenster "Konfigurationsdateien" aufzurufen. Für die Konfigurationen in diesem Fenster können Sie Lade- oder Entladeoperationen ausführen.

3. Klicken Sie in der Funktionsleiste des Fensters **Lösungsansichten** auf **Hinzufügen**.
 - a. Geben Sie im Feld **Name** den Namen für die Lösungsansicht ein, die Sie erstellen wollen.
 - b. Geben Sie optional eine **Beschreibung** für die Lösungsansicht ein, die Sie erstellen wollen.
 - c. Wählen Sie den **Server** aus, der die Konfigurationsdatei und die Fertigungslinien enthält, die Sie für die Erstellung der Lösungsansicht verwenden wollen.
 - d. Wählen Sie in der Liste **Konfigurationen** die Konfigurationsdatei aus, die Sie verwenden wollen.
 - e. Aktivieren oder inaktivieren Sie die **automatische Aktualisierung**.

Wenn die Fertigungslinien oder Eigenschaften für eine Konfiguration geändert werden, wird die Lösungsansicht automatisch geändert, falls die Option für die **automatische Aktualisierung** aktiviert ist.

Anmerkung: Wenn Sie die automatische Aktualisierung auswählen, können Sie die mit dieser Einstellung erstellte Lösungsansicht weder bearbeiten, noch können Sie Regeln und Auslöser für Lösungsansichten erstellen, während die automatische Aktualisierung aktiviert ist. Falls Sie die Lösungsansicht bearbeiten oder Regeln und Auslöser hinzufügen wollen, müssen Sie die automatische Aktualisierung inaktivieren. Benutzer müssen die Funktion für die automatische Aktualisierung inaktivieren, damit sie Regeln und Auslöser für Lösungsansichten erstellen können, die für die automatische Aktualisierung gekennzeichnet sind. Änderungen an der Konfiguration in der Lösungsansicht können Sie prüfen, indem Sie im Fenster "Lösungsansicht" die Schaltfläche "Neuanzeige" auswählen. Diese Schaltfläche ist nur bei Konfigurationen sichtbar, deren Attribut "auto-update" (automatische Aktualisierung) auf "true" (wahr) gesetzt ist. Bei allen Konfigurationen, die mit dem Assistenten "Lösungsansicht erstellen" manuell erstellt werden, ist das Attribut für die automatische Aktualisierung auf "false" (falsch) gesetzt.

Anmerkung: Lösungsansichten, die mit der Option für die automatische Aktualisierung erstellt wurden, müssen erneut geladen werden. Verwenden Sie die Option **Neuanzeige der Lösungsansicht** im Fenster **Lösungsansichten**. Bei Lösungsansichten, die für die automatische Aktualisierung gekennzeichnet sind, müssen Sie die Konfigurationsdatei erneut laden und die Lösungsansicht durch Klicken auf **Neuanzeige der Lösungsansicht** aktualisieren. Falls ein Benutzer für eine Lösungsansicht, die mit der Option **Einfach** erstellt und für die automatische Aktualisierung gekennzeichnet wurde, keine Neuanzeige aufruft, verursacht die Lösungsansicht möglicherweise Inkonsistenzen in der AMC-Datenbank. Inkonsistenzen in Lösungsansichten, die nicht aktualisiert werden, können zu einem Fehlverhalten von Action Manager führen.

4. Bei der Erstellung einer Lösungsansicht können Sie die folgenden Optionen verwenden:

Einfach

Diese Option erstellt eine Lösungsansicht mit gängigen Standardoptionen.

Automatische Aktualisierung

Bei Lösungsansichten, die für die automatische Aktualisierung gekennzeichnet sind, müssen Sie die Konfigurationsdatei erneut laden und eine Neuanzeige der Lösungsansicht veranlassen.

Lösungsansicht aus veröffentlichter Lösung hinzufügen

Die Lösungsansicht wird auf der Basis der veröffentlichten Lösung wie im Konfigurationseditor von IBM Security Directory Integrator angegeben erstellt. Bei dieser Option muss Ihrer aktiven Konfigurationsinstanz eine veröffentlichte Lösung zugeordnet sein. Außerdem ist ein IBM Security Directory Integrator-Server der Version 7.0 erforderlich.

Lösungsansicht mit allen zugänglich gemachten Fertigungslinien hinzufügen

Es wird eine Lösungsansicht mit allen Fertigungslinien aus der zugänglich gemachten Konfigurationsinstanz, ohne Eigenschaften und ohne definierte Fertigungslinie für den ordnungsgemäßen Betrieb erstellt. Verwenden Sie diese Option für einen Schnelleinstieg (ist zu Entwicklungszwecken hilfreich). Diese Option ist nur bei IBM Security Directory Integrator-Servern der Version 6.0 und höher verfügbar.

Lösungsansicht mit allen zugänglich gemachten Fertigungslinien und Eigenschaften hinzufügen

Es wird eine Lösungsansicht mit allen Fertigungslinien aus der zugänglich gemachten Konfigurationsinstanz, allen Eigenschaften und ohne definierte Fertigungslinie für den ordnungsgemäßen Betrieb erstellt. Java-Eigenschaften werden bei dieser Option nicht zugänglich gemacht. Die Option ist nur bei IBM Security Directory Integrator-Servern der Version 6.1 und höher verfügbar. Verwenden Sie diese Option für einen Schnelleinstieg (ist zu Entwicklungszwecken hilfreich).

Lösungsansicht mit allen zugänglich gemachten Fertigungslinien und Benutzereigenschaften hinzufügen

Es wird eine Lösungsansicht mit allen Fertigungslinien aus der zugänglich gemachten Konfigurationsinstanz, allen Eigenschaften und ohne definierte Fertigungslinie für den ordnungsgemäßen Betrieb erstellt. Dies entspricht etwa einer Option für den Schnelleinstieg. Diese Option ist bei IBM Security Directory Integrator-Servern der Version 6.0 inaktiviert, weil IBM Security Directory Integrator-Eigenschaften in IBM Security Directory Integrator-Servern der Version 6.0 nicht verfügbar sind.

Anmerkung: Damit die benutzerdefinierten Eigenschaften in der Anzeige "Eigenschaftsspeicher" angezeigt werden, müssen Sie eine der folgenden Aktionen ausführen:

- Legen Sie die Datei ".properties" in dem Ordner ab, der die Konfigurationsdatei enthält.
- Geben Sie bei Erstellung des Eigenschaftsspeichers im Konfigurationseditor (**Neuer Eigenschaftsspeicher > Registerkarte "Connector" > Registerkarte "Konfiguration" > Parameter "Pfad/URL für Objektgruppe"**) einen absoluten Pfad für die Eigenschaftendatei an.

5. Klicken Sie auf **OK**, um die Erstellung der Lösungsansicht abzuschließen.

Konfigurationsdateien (zum Laden/Entladen von Konfigurationen)

Nachstehend erhalten Sie Informationen zu Konfigurationsdateien und deren Bearbeitung.

Um auf das Fenster "Konfigurationsdateien" und die Optionen für das Laden, Entladen, erneute Laden und Aktualisieren von Konfigurationsdateien zuzugreifen, wählen Sie im Navigationsbereich auf der linken Seite den Eintrag **Lösungsansichten** aus. Wählen Sie einen Server sowie eine Konfigurationsdatei aus und klicken Sie dann auf die Schaltfläche **Konfigurationsdateien anzeigen**. Daraufhin wird das Fenster "Konfigurationsdateien" aufgerufen. In diesem Fenster werden geladene Konfigurationen und die Konfigurationen im Ordner "configs" des fernen IBM Security Directory Integrator-Servers angezeigt. Wenn AMC mit einem IBM Security Directory Integrator-Server verbunden ist, enthält das Fenster "Konfigurationsdateien" eine Liste aller Dateien im fernen Konfigurationsordner (unabhängig davon, ob es sich um gültige IBM Security Directory Integrator-Konfigurationsdateien handelt oder nicht). Ladeoperationen sollten Sie nur für gültige Konfigurationsdateien von IBM Security Directory Integrator ausführen, da andernfalls in AMC eine Fehlermeldung ausgegeben wird. Der Status "Geladen" bzw. "Nicht geladen" wird in der Statusspalte durch ein grünes Symbol (= geladen) bzw. ein rotes Symbol (= nicht geladen) wiedergegeben. In der Spalte **Auswählen** der Tabelle mit den Konfigurationsdateien können Sie eine oder mehrere Konfigurationen auswählen. Nachdem Sie eine Konfiguration ausgewählt haben, können Sie mit den Schaltflächen **Laden**, **Laden als...**, **Entladen**, **Erneut laden** oder **Neuanzeige** über der Tabelle die entsprechende Aktion ausführen. Falls Sie eine Konfiguration laden wollen, die durch ein Kennwort geschützt ist, wählen Sie die Konfiguration aus und geben Sie das Kennwort im Feld "Kennwort" ein.

Nach Ausführung der Aktion (Laden, Laden als, Erneut laden, Entladen, Neuanzeige) wird eine Nachricht ausgegeben, die das Ergebnis beschreibt, also angibt, ob die Aktion erfolgreich oder nicht erfolgreich ausgeführt wurde. Bei den Aktionen "Laden", "Erneut laden" oder "Entladen" wird der neue Status der ausgewählten Konfigurationen in der Statusspalte angezeigt.

Anmerkung: Zur Ausführung dieser Aktion benötigen Sie eine Berechtigung als Superadministrator oder Konfigurationsadministrator.

- Zum Laden von Konfigurationen wählen Sie die gewünschten Konfigurationen aus und klicken Sie auf **Laden**.
- Zum Laden mehrerer Instanzen einer Konfiguration wählen Sie die zu ladende Konfiguration aus und klicken Sie auf **Laden als...** Daraufhin wird das Fenster **Angepasstes Laden** geöffnet, in dem Sie die **Konfigurationsdatei**, den **Ausführungsnamen der Konfiguration**, das **Konfigurationskennwort** und den **Eigenschaftsspeicherwert** angeben können.
- Zum Entladen von Konfigurationen wählen Sie die zu entladenden Konfigurationen aus und klicken Sie auf **Entladen**.

Anmerkung: Beim Laden eines Servers werden die der ausgewählten Konfiguration zugeordneten Fertigungslinien nicht automatisch gestartet. Nur Fertigungslinien, die für das automatische Starten gekennzeichnet sind, werden beim Laden gestartet.

- Zum Entladen von Konfigurationen wählen Sie die geladenen Konfigurationen, die Sie entladen wollen, aus und klicken Sie auf **Entladen**. Sie können nur Konfigurationen entladen, die den Status "Geladen" aufweisen.
- Zur Neuanzeige von Konfigurationen klicken Sie auf **Neuanzeige**. Daraufhin werden die Informationen für alle Konfigurationen in der Tabelle neu angezeigt.

- Klicken Sie auf **Schließen**, nachdem Sie die gewünschten Änderungen vorgenommen haben.

Anmerkung: Benutzer müssen die Datenintegrität verwalten.

- Falls beispielsweise für eine Konfiguration mit dem Namen "config1.xml" und dem Ausführungsnamen "ABC" eine Lösungsansicht und Regeln erstellt wurden, sollte eine andere Konfiguration (z. B. "config2.xml") nicht mit dem Namen "ABC" (weder als Lösungsname noch als Ausführungsname) geladen werden.
- Falls Sie Lösungsansichten wiederverwenden wollen, die Sie mit einem bestimmten Ausführungsnamen und einer speziellen Gruppe von Eigenschaftendateien erstellt haben, müssen Sie diese Konfiguration unter Verwendung desselben Ausführungsnamens und derselben Eigenschaftendateien entladen.

Angepasstes Laden:

Mit den hier aufgeführten Schritten können Sie mehrere Konfigurationsinstanzen in AMC laden.

Informationen zu diesem Vorgang

Der IBM Security Directory Integrator-Server unterstützt das Laden mehrerer Instanzen derselben Konfiguration mit verschiedenen Ausführungsnamen. Falls Sie Konfigurationsinstanzen mit der Option **Laden als...** laden, können Sie diese Konfigurationen für die Erstellung von Lösungsansichten und Regeln verwenden.

1. Wählen Sie auf der Seite **Willkommen** die Optionen **Server -> Konfigurationsdateien** aus.
2. Klicken Sie auf **Laden als...**
Daraufhin wird das Fenster **Angepasstes Laden** aufgerufen.
 - a. Wählen Sie die **Konfigurationsdatei**, aus der Sie mehrere Instanzen erstellen wollen, aus und klicken Sie auf **Los**.
 - b. Geben Sie den **Ausführungsnamen der Konfiguration** ein.
 - c. Geben Sie das **Konfigurationskennwort** ein.
 - d. Geben Sie für jeden Eigenschaftsspeichernamen den **Eigenschaftsspeicherwert** ein.
3. Klicken Sie auf **OK**, um die eingegebenen Werte zur Erstellung einer Instanz der Konfiguration mit dem angegebenen Ausführungsnamen zu erstellen. Nachdem die Instanz der Konfiguration erstellt wurde, werden Sie zum Fenster **Laden/Erneut laden** zurückgeführt.
4. Klicken Sie auf **Abbrechen**, wenn die Konfiguration mit den im Fenster **Angepasstes Laden** angegebenen Werten nicht erstellt werden soll.

Statusüberwachung und Action Manager

Nachstehend erhalten Sie Informationen zu allen Aktionen, die Sie mit der Funktion zur Statusüberwachung ausführen können.

Falls die Kategorie **Status überwachen** im Hauptnavigationsbereich von AMC noch nicht erweitert wurde, führen Sie dies nun aus.

Führen Sie eine der folgenden Aktionen aus:

- In der Tabelle "Status überwachen" können Sie Informationen zu jeder Lösungsansicht ermitteln. Es werden Angaben über die Lösungsansichten wie der Action Manager-Status, das Ergebnis der Überwachung des ordnungsgemäßen Betriebs

und der Status der Überwachung des ordnungsgemäßen Betriebs angezeigt. Außerdem können Sie **Details der Lösungsansicht**, **Serverinformationen** und **bevorzugte Ansichten** anzeigen.

- Sie können Action Manager-Regeln hinzuzufügen, bearbeiten oder löschen, indem Sie auf „**Action Manager**“ auf **Seite 309** klicken.

Status überwachen

Sie können übergeordnete Informationen zu jeder bevorzugten Lösungsansicht anzeigen.

In diesem Fenster werden die Ansichten angezeigt, die im Fenster "Bevorzugte Ansichten" ausgewählt wurden (auf Letzteres greifen Sie durch Auswahl der Optionen **Erweitert** -> **Bevorzugte Lösungsansichten** zu). Übergeordnete Informationen, beispielsweise:

Action Manager-Status

Hier wird der Status der Action Manager-Regeln für die ausgewählte Lösungsansicht angezeigt. Ein blaues Ausrufezeichen gibt an, dass kürzlich keine Action Manager-Regeln ausgelöst wurden. Ein Ausrufezeichen in einem gelben Dreieck gibt an, dass kürzlich eine Action Manager-Regel ausgelöst wurde.

Ergebnis der Überwachung des ordnungsgemäßen Betriebs

Zeigt das Ergebnis der Überwachung des ordnungsgemäßen Betriebs an, das aus dem Attribut "HealthAL.result" des letzten Work-Eintrags in der Fertigungslinie für den ordnungsgemäßen Betrieb der Lösungsansicht abgerufen wurde. Dieser Wert wird als Text angezeigt.

Status der Überwachung des ordnungsgemäßen Betriebs

Zeigt den Status der Überwachung des ordnungsgemäßen Betriebs an, das aus dem Attribut "HealthAL.status" in der Fertigungslinie für den ordnungsgemäßen Betrieb der Lösungsansicht abgerufen wurde.

Falls Sie eine Datei ".gif" mit demselben Namen wie der zurückgegebene Statuswert im Verzeichnis resources/amc_images/healthAL von AMC festgelegt haben, wird zusätzlich das Bild aus dieser Datei ".gif" in dieser Spalte angezeigt. Beispiel: Wenn das Attribut "healthAL.result" mit dem Wert "Error" zurückgegeben wird und Sie im oben genannten Verzeichnis eine Datei "Error.gif" erstellt haben, wird das Bild aus der Datei "Error.gif" in der Tabellenspalte angezeigt.

In diesem Fenster können Sie die folgenden Aktionen ausführen:

- **Details der Lösungsansicht anzeigen:** Um die Details einer bestimmten Lösungsansicht anzuzeigen, wählen Sie die gewünschte Lösungsansicht aus und klicken Sie auf **Details der Lösungsansicht**.
- **Informationen zum IBM Security Directory Integrator-Server anzeigen:** Um die Details des Servers anzuzeigen, zu dem die Lösungsansicht gehört, klicken Sie auf **Serverinformationen**.
- **Bevorzugte Lösungsansichten anzeigen:** Um die bevorzugten Lösungsansichten anzuzeigen, klicken Sie auf **Bevorzugte Lösungsansichten anzeigen**. Diese Schaltfläche ist nur dann sichtbar, wenn bevorzugte Lösungsansichten definiert sind. Bevorzugte Lösungsansichten können Sie im Fenster "Bevorzugte Lösungsansichten" der **Benutzereinstellungen** definieren.

Details der Lösungsansicht:

Sie können die speziellen Details der Lösungsansicht anzeigen.

Die Anzeige "Details der Lösungsansicht" bietet genaueren Aufschluss über die speziellen Details einer Lösungsansicht, die ein Administrator anzeigen und für die er Aktionen ausführen kann.

Dieses Fenster enthält zwei Tabellen. In der oberen Tabelle werden die Fertigungslinien angezeigt, die der ausgewählten Lösungsansicht zugeordnet sind, und der Status der jeweiligen Lösungsansicht ist angegeben. In der unteren Tabelle werden Protokollinformationen zu kürzlich ausgelösten Action Manager-Regeln angezeigt.

Klicken Sie auf **Schließen**, nachdem Sie alle gewünschten Änderungen vorgenommen haben.

Tabelle "Details der Lösungsansicht":

Die Tabelle **Details der Lösungsansicht** enthält die aufgeführten Spalten.

Spalten

Auswählen

Wählen Sie das Optionsfeld neben der Fertigungslinie aus, für die Sie eine Aktion ausführen wollen.

Fertigungslinien

In dieser Spalte wird der Name der Fertigungslinie angezeigt.

Status In dieser Spalte wird der Status der Fertigungslinie angezeigt, z. B. **Aktiv** oder **Gestoppt**.

Startzeit

Bei aktiver Fertigungslinie

In dieser Spalte ist die Startzeit angegeben, zu der die aktive Fertigungslinie gestartet wurde. Die Startzeit basiert auf der aktiven Fertigungslinie.

Bei gestoppter Fertigungslinie

In dieser Spalte ist der Zeitpunkt angegeben, an dem die letzte Ausführung der Fertigungslinie gestartet wurde. Die Startzeit basiert auf dem neuesten Tombstone-Eintrag für die Fertigungslinie (nur bei Servern von IBM Security Directory Integrator verfügbar).

Letzte Stoppzeit

In dieser Spalte ist der Zeitpunkt angegeben, an dem die letzte Ausführung der Fertigungslinie gestoppt wurde. Die Stoppzeit basiert auf dem neuesten Tombstone-Eintrag für die Fertigungslinie (nur bei Servern von IBM Security Directory Integrator verfügbar).

Statistics (Statistik)

In dieser Spalte wird die aktuelle Statistik der aktiven Fertigungslinie angezeigt.

Aktionen

Zur Auswahl der Operationen, die Sie ausführen wollen, können Sie die Funktionsleiste über der Tabelle oder das Dropdown-Menü **Aktion auswählen** verwenden. Möglich sind beispielsweise die folgenden Aktionen:

- Tombstones anzeigen: Wählen Sie die Fertigungslinie aus, die Sie anzeigen wollen, und klicken Sie auf die Schaltfläche **Tombstones anzeigen**.
- Protokolle anzeigen: Wählen Sie die Fertigungslinie aus, die Sie anzeigen wollen, und führen Sie eine der folgenden Aktionen aus:

- Klicken Sie in der Funktionsleiste auf die Schaltfläche **Protokolle anzeigen**.
- Wählen Sie im Dropdown-Menü **Aktion auswählen** die Option **Protokolle anzeigen** aus und klicken Sie auf **Los**.
- Eigenschaften verwalten: Wählen Sie das Optionsfeld neben der Fertigungslinie aus, deren Eigenschaften Sie verwalten wollen, und klicken Sie in der Funktionsleiste auf die Schaltfläche **Eigenschaften verwalten**.
- Fertigungslinie starten:
 1. Wählen Sie die Fertigungslinie aus, die Sie starten wollen.
 2. Klicken Sie auf die Schaltfläche **Dialogfenster anzeigen**.
 3. Klicken Sie auf **Fertigungslinie starten**.
- Fertigungslinie stoppen: Wählen Sie die Fertigungslinie aus, die Sie stoppen wollen, und führen Sie eine der folgenden Aktionen aus:
 1. Wählen Sie die Fertigungslinie aus, die Sie stoppen wollen.
 2. Klicken Sie auf die Schaltfläche **Dialogfenster anzeigen**.
 3. Klicken Sie auf **Fertigungslinie stoppen**.

Anmerkung: Ab IBM Security Directory Integrator Version 7.1 ist eine neue Option namens "Fertigungslinie ordnungsgemäß stoppen" verfügbar. Bei Auswahl dieser Option wird die Fertigungslinie in einem neuen Thread gestoppt. Das ordnungsgemäße Stoppen von Fertigungslinien ist bei IBM Security Directory Integrator-Servern mit einer früheren Version als Version 7.1 nicht verfügbar.

- Details der Lösungsansicht: Klicken Sie auf die Schaltfläche **Details der Lösungsansicht**. Wählen Sie die Komponente aus, die Sie anzeigen wollen, z. B. Fertigungslinien.

Fertigungslinie starten

Wählen Sie diese Option aus, um die Fertigungslinie auszuführen.

Fertigungslinie synchron starten

AMC wartet die Beendigung der Fertigungslinie ab und zeigt den Status der aktiven Fertigungslinie in regelmäßigen Abständen an. Die Ausgabeschemaattribute der Fertigungslinie nach ihrer Beendigung können bei synchronen Ausführungen von Fertigungslinien angezeigt werden.

Fertigungslinie im Simulationsmodus starten

Die Fertigungslinie führt alle Komponenten mit Ausnahme der Connectors für die Modi "Add", "Update" und "Delete" aus. Im Wesentlichen werden die Methoden "putEntry", "modEntry" und "deleteEntry" von Connectors im Simulationsmodus nicht aufgerufen. Infolgedessen führt eine im Simulationsmodus ausgeführte Fertigungslinie in Drittanbieterrepositorys keine Aktionen zum Hinzufügen, Modifizieren oder Löschen aus. Weitere Informationen zum Simulationsmodus enthält der entsprechende Abschnitt in der Veröffentlichung *Directory Integrator - Konfiguration*.

Tombstones anzeigen

Falls auf dem fernen IBM Security Directory Integrator-Server Tombstones aktiviert sind, kann AMC die Tombstone-Einträge für beendete Fertigungslinien anzeigen. Dieses Fenster enthält nützliche Informationen zu Tombstone-Einträgen. So ist beispielsweise angegeben, wann der Eintrag in den Tombstonestatus geändert wurde.

Tombstones löschen

Wählen Sie im Fenster **Status überwachen** eine Fertigungslinie aus. Wählen Sie den Pfeil rechts neben der Fertigungslinie aus und wählen Sie im Menü die Option **Tombstones löschen** aus. Hierdurch wird das Fenster **Tombstones löschen** aufgerufen. Im Abschnitt dieses Fensters mit den Komponentendetails sind die Lösungsansicht und die Fertigungslinie angegeben, mit denen Sie arbeiten. Wählen Sie im Abschnitt "Löschkriterien auswählen" eine der Optionen aus, um anzugeben, welche Tombstones gelöscht werden sollen:

- Wählen Sie die Option **Alle Tombstones** aus, um alle Tombstones der ausgewählten Fertigungslinie zu löschen.
- Verwenden Sie die Optionen **Startdatum** und **Enddatum**, um den Datumsbereich anzugeben, dessen Tombstones gelöscht werden sollen. AMC berechnet die Anzahl der Tage vom ausgewählten Datum bis zum aktuellen Datum. Anschließend löscht AMC die Tombstones, die für die berechnete Anzahl von Tagen erstellt wurden.
- Verwenden Sie die Option **Anzahl der zurückzugebenden Einträge** unter Angabe einer ganzen Zahl, um festzulegen, wie viele der neuesten Tombstones gelöscht werden sollen. Sobald Sie auf **Löschen** klicken, wird eine Bestätigungsnachricht ausgegeben. Nachdem Sie den Vorgang bestätigt haben, führt AMC den Löschbefehl aus.

Protokolle anzeigen

Die Protokolle für eine bestimmte Fertigungslinie werden im Fenster "Protokolle anzeigen" angezeigt. Wählen Sie die Optionen **Status überwachen** -> **Details der Lösungsansicht** -> **Protokolle anzeigen** aus, um die Liste mit den Protokolldateien für die ausgewählte Fertigungslinie anzuzeigen. Klicken Sie auf das Optionsfeld neben dem gewünschten Protokoll und klicken Sie dann auf **Protokolle anzeigen**.

Anmerkung: Damit ein Fertigungslinienprotokoll in AMC angezeigt werden kann, muss die Fertigungslinie zur Protokollierung die Protokollfunktion für das Systemprotokoll verwenden.

Action Manager-Ergebnistabelle:

Nachstehend erhalten Sie Informationen zu den Spalten der **Action Manager-Ergebnistabelle** sowie zur Durchführung von Operationen für Action Manager-Ergebnisse.

Wenn eine in Action Manager festgelegte Regel ausgelöst wird, werden Informationen zum Verstoß protokolliert, beispielsweise die Quelle des Verstoßes, eine Beschreibung des Fehlers sowie der Zeitpunkt, an dem der Verstoß stattfand. Diese Details werden in der **Action Manager-Ergebnistabelle** angezeigt.

Spalten

Die **Action Manager-Ergebnistabelle** enthält die folgenden Spalten:

Auswählen

Wählen Sie das Optionsfeld neben der Nachricht aus, für die Sie eine Aktion ausführen wollen.

Quelle

In dieser Spalte wird der Name der ausgelösten Action Manager-Regel angezeigt.

Bewertung

In dieser Spalte ist die Bewertung der Nachricht angegeben.

Nachricht

In dieser Spalte wird die Nachricht angezeigt, die der Action Manager-Aktion zugeordnet ist.

Beschreibung

Diese Spalte enthält zusätzliche Informationen zur Nachricht.

Zeitmarke

In dieser Spalte wird der Zeitpunkt angezeigt, an dem die Action Manager-Regel ausgelöst und die Nachricht generiert wurde.

Aktionen

Wählen Sie eines oder mehrere zu löschende Ergebnisse aus und klicken Sie auf **Löschen**.

Komponenten anzeigen:

Mit der Operation "Komponenten anzeigen" können Sie die verschiedenen Connectors, Funktionskomponenten usw. anzeigen, die in der ausgewählten Fertigungslinie konfiguriert sind.

Anmerkung: Verzweigungskomponenten (IF, SWITCH, usw.) und Scriptkomponenten werden nicht angezeigt. Dies ist beabsichtigt, da der Schwerpunkt dieser Anzeige auf den Schlüsselementen, also Connectors und Funktionskomponenten liegen soll.

Bevorzugte Lösungsansichten anzeigen:

Bevorzugte Lösungsansichten sind die Standardlösungsansichten, die im Fenster **Status überwachen** angezeigt werden.

Details der Lösungsansicht in AMC aktualisieren

Mit den hier aufgeführten Anweisungen können Sie das Aktualisierungsintervall ändern.

Das Fenster "Details der Lösungsansicht" wird nach einem festgelegten Zeitintervall zum Anzeigen des aktuellen Fertigungslinienstatus aktualisiert. Standardmäßig ist die Aktualisierungsrate auf 600 Sekunden gesetzt. Der Integrated Solutions Console-Administrator verfügt über die Berechtigung zum Ändern des Aktualisierungsintervalls.

Gehen Sie wie folgt vor, um das Aktualisierungsintervall zu ändern:

1. Rufen Sie die Anmeldeseite von AMC auf.
2. Geben Sie Ihren Benutzernamen und das entsprechende Kennwort ein und klicken Sie auf **Anmelden**. Daraufhin wird die Begrüßungsseite der Integrated Solutions Console angezeigt.
3. Klicken Sie in der linken Navigationsstruktur auf **Einstellungen -> Globale Aktualisierung verwalten**.
4. Klicken Sie im Fenster "Globale Aktualisierung verwalten" auf den Link **Status überwachen**.
5. Ändern Sie die Aktualisierungskonfigurationseinstellungen und klicken Sie auf **OK**.

Action Manager

Hier können Sie Regeln, Auslöser und Aktionen hinzufügen, löschen oder modifizieren, die infolge der Regelausführungs- und Auslöserbedingungen auszuführen sind.

Konfigurationsregeln hinzufügen/bearbeiten:

In diesem Fenster können Sie Konfigurationsregeln hinzufügen oder bearbeiten.

Mit den Einstellungen in diesem Fenster können Sie eine Action Manager-Regel für die aktuelle Lösungsansicht erstellen bzw. eine vorhandene Regel modifizieren (siehe hierzu auch „Action Manager“ auf Seite 274).

Eine Regel besteht aus zwei Teilen:

- Die Bedingung, unter der die Regel aufgerufen werden soll, also der so genannte "Auslöser".
Einige Beispiele für Auslöser sind Fehler der Server-API, das Fehlschlagen einer Fertigungslinie oder das Fehlschlagen einer Fertigungslinie, die in angegebenen Intervallen ausgeführt werden soll.
- Die Gruppe der alternativen Aktionen, die beim Auftreten des Auslösers auszuführen sind.

Einstellungen für Konfigurationsregeln:

Dieses Fenster betrifft den ersten Teil der Regel, denn hier wird der Auslöser definiert. Sie können einen Namen, eine Beschreibung und einen Auslösertyp auswählen.

Name

Geben Sie einen Namen für die Regel ein. Falls Sie eine Regel hinzufügen, ist eine Angabe in diesem Feld erforderlich.

Beschreibung

Geben Sie optional eine Beschreibung für die Regel ein.

Auslösertyp

Der Auslösertyp definiert die Bedingungen, unter denen eine Regel aufgerufen wird. Wählen Sie im Dropdown-Menü einen Auslösertyp aus:

Kein Auslöser

Es gibt keine Auslösebedingung für die Regel.

Bei Beendigung der Fertigungslinie

Die Regel wird ausgelöst, wenn die angegebene Fertigungslinie beendet wird.

Beim Laden der Konfiguration

Die Regel wird ausgelöst, wenn Action Manager für die angegebene Konfiguration ein Ereignis für das Laden der Konfiguration empfängt.

Beim Entladen der Konfiguration

Die Regel wird ausgelöst, wenn Action Manager für die angegebene Konfiguration ein Ereignis für das Entladen der Konfiguration empfängt.

Bei einer FL-Ergebnisabfrage

Die Regel wird ausgelöst, wenn der letzte Work-Eintrag der ange-

gebenen Fertigungslinie ein Attribut enthält, das mit einer angegebenen Bedingung und einem angegebenen Wert übereinstimmt.

Bei einem Server-API-Fehler

Die Regel wird ausgelöst, wenn Action Manager keine Verbindung zum fernen Server über die Server-API herstellen kann. Diese Regel wird nur ein einziges Mal ausgelöst. Die Regel wird zurückgesetzt, wenn die Verbindung zum Server über die Server-API wieder möglich ist.

Beim Empfang eines Ereignisses

Die Regel wird ausgelöst, wenn Action Manager ein Ereignis empfängt, das die Bedingungen in den Feldern für den Ereignistyp, die Ereignisquelle und die Ereignisdaten erfüllt.

Bei einer Eigenschaft

Die Regel wird ausgelöst, wenn die angegebene Eigenschaft die Angaben für "Eigenschaftsname", "Bedingung" und "Wert" erfüllt.

Bei lokaler Variable

Die Regel wird ausgelöst, wenn die angegebenen Variablen die angegebene Bedingung erfüllen. Action Manager überprüft diese Eigenschaft in regelmäßigen Abständen.

Anmerkung: Diese Regel wird nur ein einziges Mal ausgelöst und erst dann in den Bereitstatus zurückgesetzt, wenn Action Manager feststellt, dass diese Variablen die angegebenen Kriterien nicht mehr erfüllen. Durch die erneute Überprüfung wird sichergestellt, dass die Regel nicht wiederholt ausgelöst wird, obwohl die Auslöserbedingung nur ein einziges Mal auftritt.

Exit-Code der Fertigungslinie überprüfen

Die Regel wird ausgelöst, wenn eine Fertigungslinie abnormal beendet wird. Sie können ein Fehlerobjekt definieren, nach dem Action Manager im Exit-Code der Fertigungslinie sucht.

Bei Ablauf des Zeitraums seit der letzten Ausführung

Die Regel wird ausgelöst, wenn die angegebene Fertigungslinie während eines festgelegten Zeitraums nicht ausgeführt wurde.

Zeitgeber

Die Regel wird im angegebenen Intervall kontinuierlich ausgelöst.

Auslöser konfigurieren:

Für jeden Auslösertyp gibt es verschiedene Einstellungsoptionen. Falls einige der nachfolgend aufgelisteten Felder in einem Fenster nicht angezeigt werden, liegt dies daran, dass sie von dem Auslösertyp, den Sie gegenwärtig ausgewählt haben, nicht unterstützt werden.

Quelle

Geben Sie die Quelle ein, die Sie überwachen wollen.

Daten Geben Sie die Daten ein, die Sie überwachen wollen.

Eigenschaftsname

Wählen Sie im Dropdown-Menü den Namen der Eigenschaft aus, die Sie überwachen wollen.

Bedingung

Wählen Sie die Bedingung für den Vergleich von Eigenschaft und Wert aus. Folgende Optionen sind möglich:

- gleich
- ungleich
- größer als
- kleiner als

Wert Geben Sie den Wert ein, den Sie überwachen wollen.

Konfigurierte Aktionen:

In dieser Tabelle können Sie Aktionen hinzufügen, modifizieren oder löschen. Sie können auch Aktionen in der Tabelle nach oben oder nach unten verschieben.

Für jede Aktion in der Tabelle "Konfigurierte Aktionen", die Sie auswählen können, gibt es eine Spalte, in der Sie den besonderen Auslöser **Ausführung bei Fehler** aktivieren können. Der Auslöser **Ausführung bei Fehler** bewirkt, dass die ausgewählte Aktion ausgeführt wird, wenn eine Fehlerbedingung auftritt.

- Wählen Sie zur Auswahl der Aktion, die Sie verwalten wollen, das Optionsfeld vor der aufgelisteten Aktion aus.
- Zum Hinzufügen einer Aktion klicken Sie auf **Hinzufügen**.
- Zum Löschen einer Aktion wählen Sie die zu löschende Aktion aus und klicken Sie auf **Löschen**.
- Zum Modifizieren einer Aktion wählen Sie die zu modifizierende Aktion aus und klicken Sie auf **Modifizieren**.
- Um eine Aktion in der Tabelle um eine Position aufwärts zu verschieben, wählen Sie die gewünschte Aktion aus und klicken Sie auf **Nach oben**.
- Um eine Aktion in der Tabelle um eine Position abwärts zu verschieben, wählen Sie die gewünschte Aktion aus und klicken Sie auf **Nach unten**.

Die Auswahl des Auslösers **Ausführung bei Fehler** bewirkt, dass Aktionen nur dann ausgeführt werden, wenn bei der Ausführung der vorherigen Aktionen ein Fehler auftrat. Solche Aktionen können für Fehlerbehebungsmaßnahmen eingesetzt werden, mit denen Fehler bearbeitet werden können, die möglicherweise bei der Ausführung von vorhergehenden Aktionen aufgetreten sind. Aktionsfehlervariablen: AMC und Action Manager bieten die Möglichkeit, den Aktionsfehler in den verschiedenen Aktionen verfügbar zu machen. Wenn ein Fehler bei der Ausführung einer der konfigurierten Aktionen auftritt, kann Ihnen dieser Fehler zu einem beliebigen Zeitpunkt in Form von speziellen reservierten Variablen verfügbar gemacht werden. Diese reservierten Variablen können Sie dann in anderen konfigurierten Aktionen verwenden. Bei Ausführung der folgenden Aktionen ersetzt Action Manager die Zeichenfolge %Action_Error% durch den tatsächlichen Fehler, der während der Ausführung der vorhergehenden Aktionen auftrat. Wenn kein Fehler auftritt, wird die Variable %Action_Error% nicht ersetzt und bleibt unverändert.

- E-Mail senden
- Befehl ausführen
- Aktion zum Senden eines Ereignisses
- Aktion zum Schreiben in das Protokoll

Aktion hinzufügen/modifizieren:

Wenn eine Regel ausgelöst wird, führt Action Manager die Aktionen aus, die der Regel zugeordnet sind. Sie können die Aktionen angeben oder modifizieren, die Action Manager ausführen soll, wenn die Regel ausgelöst wird.

Wählen Sie im Dropdown-Menü einen Aktionstyp aus und konfigurieren Sie ihn. Klicken Sie auf **OK**, wenn Sie die Konfiguration fertiggestellt haben.

Fertigungslinie starten

Diese Aktion startet eine Fertigungslinie. Wenn Sie diese Aktion auswählen, müssen Sie den Namen der Fertigungslinie, die Sie starten möchten, und die zugehörige Konfiguration (möglicherweise auch das Konfigurationskennwort) angeben.

Server Diese Dropdown-Liste enthält die konfigurierten Server. Der Eintrag "Lokaler Server" bezeichnet den Server auf dem Computer, auf dem Action Manager ausgeführt wird.

In fernem Konfigurationsordner auswählen

Wenn dieses Markierungsfeld ausgewählt ist, wird der ferne Server nach verfügbaren Konfigurationsdateien abgefragt. Es werden die Konfigurationsdateien angezeigt, die im Ordner vorhanden sind, dessen Pfad für die Eigenschaft "api.config.folder" in der Datei `global.properties` angegeben ist.

Konfigurationsname

Geben Sie die Konfiguration ein, zu der die im Feld "Fertigungslinie" angegebene Fertigungslinie gehört. Falls die Option **In fernem Konfigurationsordner auswählen** aktiviert ist, wird eine Liste der Konfigurationsdateien angezeigt, die auf dem fernen Server verfügbar sind. Ist diese Option abgewählt, müssen Sie den Namen einer lokal verfügbaren Konfigurationsdatei eingeben.

In diesem Feld ist eine Angabe erforderlich.

Konfigurationskennwort

Falls für die ausgewählte Konfigurationsdatei ein Konfigurationskennwort erforderlich ist, geben Sie es in diesem Feld ein. Dieses Feld kommt nur dann zur Anwendung, wenn die Konfiguration durch ein Kennwort geschützt ist.

Fertigungslinie

Geben Sie den Namen der Fertigungslinie ein, die gestartet werden soll.

Fertigungslinienoperation konfigurieren

Dieser Hyperlink startet das Dialogfenster "Operation auswählen". Falls die Fertigungslinie mit einer oder mehreren angepassten Operationen definiert wurde, können Sie in diesem Dialogfenster eine solche Operation auswählen. Anschließend werden Sie aufgefordert, die Initialisierungsattribute der Fertigungslinie und die Operationsattribute für diese Operation anzugeben. Dieses Feld wird nur für IBM Security Directory Integrator 6.1.X und für IBM Security Directory Integrator-Server bei entsprechender Konfiguration angezeigt und ist bei IBM Security Directory Integrator 6.0 nicht gültig.

Fertigungslinie stoppen

Diese Aktion stoppt eine Fertigungslinie. Wenn Sie diese Aktion auswählen, müssen Sie den Namen der Fertigungslinie, die Sie stoppen möchten, und die zugehörige Konfiguration angeben.

Server Diese Dropdown-Liste enthält die konfigurierten Server. Der Eintrag "Lokaler Server" bezeichnet den Server auf dem Computer, auf dem Action Manager ausgeführt wird.

In fernem Konfigurationsordner auswählen

Wenn dieses Markierungsfeld ausgewählt ist, wird der ferne Server nach verfügbaren Konfigurationsdateien abgefragt.

Konfigurationsname

Geben Sie die Konfiguration ein, zu der die im Feld "Fertigungslinie" angegebene Fertigungslinie gehört. Falls die Option **In fernem Konfigurationsordner auswählen** aktiviert ist, wird eine Liste der Konfigurationsdateien angezeigt, die auf dem fernen Server verfügbar sind. Ist diese Option abgewählt, müssen Sie den Namen einer lokal verfügbaren Konfigurationsdatei eingeben.

In diesem Feld ist eine Angabe erforderlich.

Fertigungslinie

Geben Sie den Namen der Fertigungslinie ein, die gestoppt werden soll.

Regel aktivieren/inaktivieren

Wählen Sie die Option "Regel aktivieren/inaktivieren" aus, um eine Action Manager-Regel zu aktivieren bzw. zu inaktivieren.

Regelname

Wählen Sie den Namen des Paares aus Regel und Lösungsansicht aus, das von der Aktion "Regel aktivieren/inaktivieren" ausgeführt werden soll. In Vorgängerversionen von IBM Security Directory Integrator wurde der Regelname anstelle eines Paares aus Regel und Lösungsansicht ausgewählt. Diese Funktionsweise ist in der aktuellen Version verfügbar. Diese Option ist mit der Aktion "Regel aktivieren/inaktivieren" verbunden.

Status Wählen Sie im Dropdown-Menü den gewünschten Status aus. Falls Sie die im Feld **Regelname** angegebene Regel aktivieren wollen, wählen Sie den Status **Aktiviert** aus. Wollen Sie die Regel inaktivieren, wählen Sie **Inaktivieren** aus.

Regel ausführen

Diese Aktion bewirkt, dass Action Manager die angegebene Regel ausführt. Action Manager führt dann die Aktionen aus, die der angegebenen Regel zugeordnet sind. Die Auslöserbedingung, die der angegebenen Regel zugeordnet ist, muss hierbei nicht erfüllt werden.

Regelname

Wählen Sie den Namen des Paares aus Regel und Lösungsansicht aus, das von der Aktion "Regel ausführen" ausgeführt werden soll. In Vorgängerversionen wurde der Regelname anstelle eines Paares aus Regel und Lösungsansicht ausgewählt. Diese Funktionsweise ist in der aktuellen Version verfügbar. Diese Option ist mit der Aktion "Regel ausführen" verbunden.

Befehl ausführen

Mit der Aktion "Befehl ausführen" kann der im Feld "Befehl" eingegebene Befehl auf dem Zielcomputer ausgeführt werden, der unter **Name des Zielcomputers** angegeben ist. Hierbei kann es sich um einen allgemeinen Befehl oder um einen speziellen IBM Security Directory Integrator-Befehl handeln. Die Aktion "Befehl ausführen" kann verwendet werden, wenn ein Benutzer eine Regel konfiguriert, die spezifische Befehle des Zielcomputers oder aber IBM Security Directory Integrator-Befehle ausführen soll, die durch AMC nicht zugänglich gemacht werden. So gibt es in AMC beispielsweise keine Aktionen für den Serverneustart oder für das Laden ei-

ner Konfiguration. Der Benutzer muss die Befehle für den Neustart bzw. das erneute Laden mithilfe der Fenster "IBM Security Directory Integrator-Server" oder "Konfigurationsdateien" ausführen. Wenn bei der Befehlsausführung ein Fehler auftritt, wird dieser mit der Variablen %ACTION_ERROR% erfasst, die von Action Manager weiter verwendet werden kann.

Name des Zielcomputers

Der Name oder die IP-Adresse des Zielcomputers. Action Manager stellt eine Verbindung zu dem in diesem Feld angegebenen Computer her. Wenn weder ein Hostname noch eine IP-Adresse für den Computer angegeben wird, wird der Befehl auf dem Computer ausgeführt, auf dem Action Manager aktiv ist.

Port Der Port gibt den Kanal an, über den Action Manager eine Verbindung zum Zielcomputer herstellen kann, auf dem der Befehl ausgeführt werden soll.

Benutzername

Beim Herstellen einer Verbindung zum Zielcomputer wird der Benutzername zur Authentifizierung und Autorisierung geprüft.

Kennwort

Beim Herstellen einer Verbindung zum Zielcomputer wird das Kennwort zur Authentifizierung und Autorisierung geprüft.

Schlüsselspeicher

Der Schlüsselspeicherpfad wird eingegeben und verwendet, falls beim Herstellen einer Verbindung zum Zielcomputer eine Zertifikatsauthentifizierung erforderlich ist.

Schlüsselspeicherkenwort

Ein Schlüsselspeicherkenwort ist erforderlich, wenn für die Herstellung einer Verbindung zum Zielcomputer eine Zertifikatsauthentifizierung obligatorisch ist.

Protokoll

Das Protokoll, das zum Herstellen einer Verbindung zur fernen Maschine verwendet wird. Für das Protokoll können die folgenden Werte angegeben werden: WINDOWS, RSH, SSH oder REXEC (Windows, ferne Shell, gesicherte Shell oder Remote Execution Protocol).

Befehl Der Befehl, der ausgeführt werden soll.

Benachrichtigungsereignis

Diese Aktion weist Action Manager an, ein Ereignis mit den angegebenen Details an den IBM Security Directory Integrator-Server zu senden, der der aktuellen Lösungsansicht zugeordnet ist. Wählen Sie **Benachrichtigungsereignis** aus, um diese Aktion zur Regel hinzuzufügen. Wenn Sie diese Aktion auswählen, müssen Sie auch einen Ereignistyp angeben.

Ereignistyp

Geben Sie einen Ereignistyp ein. In diesem Feld ist eine Angabe erforderlich.

Quelle

Geben Sie eine Quelle für den Ereignistyp ein.

Daten Geben Sie Daten für den Ereignistyp ein.

Eigenschaft modifizieren

Diese Aktion weist Action Manager an, eine Eigenschaft auf der Basis einer bestimmten Operation und eines bestimmten Wertes zu modifizieren. Wenn Sie diese Aktion auswählen, müssen Sie auch einen Wert auswählen.

Eigenschaftsname

Wählen Sie im Dropdown-Menü die zu modifizierende Eigenschaft aus.

Operation

Wählen Sie im Dropdown-Menü die Operation aus, die Sie zum Modifizieren der Eigenschaft verwenden möchten. Folgende Optionen sind möglich:

- Definieren
- Erhöhen
- Verringern

Wert Geben Sie den gewünschten Wert ein. In diesem Feld ist eine Angabe erforderlich.

Eigenschaftswert kopieren

Diese Aktion weist Action Manager an, den Wert der Quelleneigenschaft in die Zieleigenschaft zu kopieren.

Von Eigenschaft

Wählen Sie im Dropdown-Menü den Namen der Eigenschaft aus, deren Wert Sie kopieren wollen.

In Eigenschaft

Wählen Sie im Dropdown-Menü den Namen der Eigenschaft aus, in die der Wert kopiert werden soll.

In Protokoll schreiben

Diese Aktion erstellt gemäß der angegebenen Bewertung, Nachricht und Beschreibung ein Protokoll der aufgerufenen Action Manager-Regeln. Dieses Protokoll können Sie unter **Status überwachen** in der **Action Manager-Ergebnistabelle** des Fensters "Details der Lösungsansicht" anzeigen. Es wird empfohlen, für jede Regel mindestens eine Protokollaktion anzugeben. Wenn Sie diese Aktion auswählen, müssen Sie im Feld **Nachricht** eine Nachricht eingeben.

Bewertung

Wählen Sie im Dropdown-Menü die gewünschte Bewertung aus. Folgende Optionen sind möglich:

- Schwerwiegend
- Warnung
- Information
- OK

Nachricht

Geben Sie die gewünschte Nachricht ein.

Beschreibung

Geben Sie optional eine Beschreibung ein.

E-Mail senden

Diese Aktion bewirkt, dass an den von Ihnen bestimmten Empfänger eine E-Mail gesendet wird. Der Inhalt der E-Mail wird von Ihnen bereitgestellt. Zusammen mit dem Inhalt stellt Action Manager vor dem Senden der E-Mail weitere Details bereit. Im Eingabebereich für den Inhalt sowie in der

Betreffzeile können Sie den Variablenwert `%EVENT_DATA%` eingeben. Durch die Angabe von `%EventData%` wird der tatsächliche Wert der Variablen für die Ereignisdaten eingefügt, sobald die E-Mail gesendet wird. In entsprechender Weise kann auch `%Action_Error%` hier eingesetzt werden. Falls die Einstellung "Action Manager-Protokoll anhängen" aktiviert ist, werden die Action Manager-Protokolle (wie in der Datei "am_logging.properties" angegeben) als E-Mail-Anhang gesendet. Im Eingabebereich für den Inhalt können Sie den Variablenwert `%EVENT_DATA%` eingeben. Durch die Angabe von `%EventData%` im Inhalt wird der tatsächliche Wert der Variablen für die Ereignisdaten eingesetzt, sobald die E-Mail gesendet wird. In entsprechender Weise wird hier auch die Variable `%Action_Error%` ersetzt. Falls die Einstellung **Action Manager-Protokoll anhängen** aktiviert ist, werden die Action Manager-Protokolle (wie in der Datei "am_logging.properties" angegeben) als E-Mail-Anhang gesendet.

Variable für Ereignisdaten ersetzen:

Sie können Daten auswählen, die von bestimmten Action Manager-Auslösern ausgegeben wurden, und diese Daten in bestimmten Aktionen verwenden, die durch eine Regel ausgelöst werden.

Wählen Sie im Navigationsfenster auf der linken Seite oder in der Anzeige **Willkommen** die Option **Action Manager** aus. In Action Manager können Sie einer Lösungsansicht eine Regel hinzufügen. Sie können eine neue Regel benennen sowie eine vorhandene Regel bearbeiten oder löschen. Sie können Ereignisdaten beim Konfigurieren oder Senden dieser Daten an andere Aktionen verfügbar machen.

Mit Action Manager können Sie Ereignisdaten bereitstellen, wenn Aktionen für einen Auslöser konfiguriert werden. Eine Regel können Sie in Action Manager **hinzufügen**, **modifizieren** oder **löschen**. Wenn Sie eine Regel hinzufügen, vergeben Sie einen Namen für die Regel und wählen einen **Auslösertyp** aus. AMC und Action Manager stellen den ausgelösten Aktionen die Daten in Form einer reservierten Variablen zur Verfügung. Die Aktion verwendet dann die Daten, die in der Variablen gespeichert sind. Sie können diese reservierte Variable in allen Aktionen einsetzen, die Sie für diesen Auslöser konfiguriert haben.

Die folgenden Auslösertypen können Ereignisdaten erzeugen, die von Aktionen verwendet werden können:

- Beim Starten der Fertigungslinie: Ereignisdaten sind als Variable `%Event_Data%` verfügbar.
- Beim Beenden der Fertigungslinie: Ereignisdaten aus diesem Typ sind als Variable `%Event_Data%` verfügbar.
- Beim Empfang eines Ereignisses: Ereignisdaten aus dem empfangenen Ereignis werden als Variable `%Event_Data%` zugeordnet.
- Bei lokaler Variable: Ereignisdaten aus diesem Ereignis werden als Variable `%Event_Data%` zugeordnet.
- Beim Laden der Konfiguration: Ereignisdaten aus diesem Ereignis sind als Variable `%Event_Data%` verfügbar.
- Beim Entladen der Konfiguration: Ereignisdaten aus dem Auslöser sind als Variable `%Event_Data%` verfügbar.
- Bei einer FL-Ergebnisabfrage: Ereignisdaten sind als Variable `%attribute_name%` verfügbar. Die Variable `%attribute_name%` wird durch die Details für das tatsächliche Attribut aus dem letzten Work-Eintrag ersetzt.

- Exit-Code der Fertigungslinie überprüfen: Ereignisdaten sind als Variable %attribute_name% und %Event_Data% verfügbar.
 - Einstellung "Aktiviert" für "Fehlerobjekt überprüfen": Falls der Benutzer beim Konfigurieren des Auslösers "Exit-Code der Fertigungslinie überprüfen" die Option "Fehlerobjekt überprüfen" aktiviert (= auf "wahr" bzw. "true" setzt), wird die Variable %Event_Data% durch die tatsächlichen Fehlerdaten ersetzt. Die Variable %attribute_name% ist für Aktionen nicht verfügbar.
 - Einstellung "Inaktiviert" für "Fehlerobjekt überprüfen": Falls der Benutzer beim Konfigurieren des Auslösers "Exit-Code der Fertigungslinie überprüfen" die Option "Fehlerobjekt überprüfen" inaktiviert (= auf "falsch" bzw. "false" setzt), wird die Variable %attribute_name% durch die Details für das tatsächliche Attribut aus dem letzten Work-Eintrag ersetzt. Die Variable %Event_Data% ist für Aktionen nicht verfügbar.

Auslöser mit möglicher Erzeugung von Ereignisdaten:

Sie können mit den aufgeführten Auslösertypen Ereignisdaten erzeugen, die von Aktionen verwendet werden können.

- Beim Starten der Fertigungslinie: Ereignisdaten sind als Variable %Event_Data% verfügbar.
- Beim Beenden der Fertigungslinie: Ereignisdaten aus diesem Typ sind als Variable %Event_Data% verfügbar.
- Beim Empfang eines Ereignisses: Ereignisdaten aus dem empfangenen Ereignis werden als Variable %Event_Data% zugeordnet.
- Bei lokaler Variable: Ereignisdaten aus diesem Ereignis werden als Variable %Event_Data% zugeordnet.
- Beim Laden der Konfiguration: Ereignisdaten aus diesem Ereignis sind als Variable %Event_Data% verfügbar.
- Beim Entladen der Konfiguration: Ereignisdaten aus dem Auslöser sind als Variable %Event_Data% verfügbar.
- Bei einer FL-Ergebnisabfrage: Ereignisdaten sind als Variable %attribute_name% verfügbar. Die Variable %attribute_name% wird durch die Details für das tatsächliche Attribut aus dem letzten Work-Eintrag ersetzt.
- Exit-Code der Fertigungslinie überprüfen: Ereignisdaten sind als Variable %attribute_name% und %Event_Data% verfügbar.
 - Einstellung "Aktiviert" für "Fehlerobjekt überprüfen": Falls der Benutzer beim Konfigurieren des Auslösers "Exit-Code der Fertigungslinie überprüfen" die Option "Fehlerobjekt überprüfen" aktiviert (= auf "wahr" bzw. "true" setzt), wird die Variable %Event_Data% durch die tatsächlichen Fehlerdaten ersetzt. Die Variable %attribute_name% ist für Aktionen nicht verfügbar.
 - Einstellung "Inaktiviert" für "Fehlerobjekt überprüfen": Falls der Benutzer beim Konfigurieren des Auslösers "Exit-Code der Fertigungslinie überprüfen" die Option "Fehlerobjekt überprüfen" inaktiviert (= auf "falsch" bzw. "false" setzt), wird die Variable %attribute_name% durch die Details für das tatsächliche Attribut aus dem letzten Work-Eintrag ersetzt. Die Variable %Event_Data% ist für Aktionen nicht verfügbar.

Aktionen mit möglichem Zugriff auf Ereignisdaten:

Nachstehend erhalten Sie Informationen zu Aktionen mit möglichem Zugriff auf Ereignisdaten.

Die für die vorstehenden Auslöser ausgeführten Aktionen können unter Verwendung der Variablen %Event_Data% auf die durch die Auslöser erzeugten Ereignisdaten zugreifen. Jedes Vorkommen der Variablen %Event_Data% wird durch die tatsächlichen Ereignisdaten für diesen Auslöser ersetzt. Die folgenden Aktionstypen können die aus ihren jeweiligen Auslösern verfügbaren Ereignisdaten nutzen:

- Benachrichtigungsereignis: Benutzer können die Variable %Event_Data% nur im Textfeld "Daten" angeben.
- In Protokoll schreiben: Für Benutzer wird eine Protokollnachricht angezeigt, die in einer Datenbank protokolliert ist. Falls die Protokollnachricht nach dem Ersetzen der Variablen %Event_Data% eine Länge von 500 Zeichen überschreitet, wird sie nach den ersten 500 Zeichen abgeschnitten. Dies liegt daran, dass für die Datenbank eine Längenbegrenzung von lediglich 500 Zeichen besteht.
- E-Mail senden: Alle Ereignisdaten oder Fehlerdaten, die durch die Variable %Event_Data% bzw. %Action_Error% angegeben sind, werden in der Betreffzeile der E-Mail eingesetzt. Action Manager hängt vor dem Senden der E-Mail weitere Daten über die Ausführung an. Sie können den Variablenwert %EVENT_DATA% im Textfeld "Inhalt" angeben. Durch die Angabe von %EventData% im Inhalt wird der tatsächliche Wert der Variablen für die Ereignisdaten eingesetzt, sobald die E-Mail gesendet wird. In entsprechender Weise können Sie hier auch die Variable %Action_Error% ersetzen. Falls die Einstellung **Action Manager-Protokoll anhängen** aktiviert ist, werden die Action Manager-Protokolle (wie in der Datei "am_logging.properties" angegeben) als E-Mail-Anhang gesendet.

Für jede Aktion (z. B. "E-Mail senden", "Befehl ausführen", "Protokoll", "Fertigungsline starten" usw.), die als Reaktion auf denselben Auslöser ausgeführt wird, wird die Zeichenfolge %Event_Data% automatisch durch die vom Auslöser generierten Ereignisdaten ersetzt.

Zusammenfassung der Regeln anzeigen:

Sie können die aktuelle Action Manager-Instanz für die ausgewählte Fertigungsline anzeigen.

Klicken Sie auf **Zusammenfassung der Regeln anzeigen**. In der Tabelle sind alle definierten Regeln, Auslöser und Aktionen aufgelistet, die der Lösungsansicht zugeordnet sind. Klicken Sie auf **Schließen**, wenn Sie die Anzeige nicht mehr benötigen. In dieser Anzeige sind nur Regeln mit dem Status "Aktiviert" angegeben.

Eigenschaftsspeicher

Nachstehend erhalten Sie Informationen zu Eigenschaftsspeichern und deren Verwendung.

Falls Sie die Kategorie **Eigenschaftsspeicher** im Navigationsbereich von AMC noch nicht erweitert haben, führen Sie dies nun aus. Zum Hinzufügen oder Bearbeiten von Java-Eigenschaften, Lösungseigenschaften, globalen Eigenschaften, Systemeigenschaften, Benutzereigenschaften und Kennwortspeichereigenschaften klicken Sie auf **Erweitert > Eigenschaftsspeicher**.

Nachdem Sie die gewünschten Eigenschaftswerte eingegeben haben, klicken Sie auf **OK**, um Ihre Änderungen zu speichern.

Die Reihenfolge, in der die Eigenschaftsspeicher aufgelistet werden, ist relevant. Die Eigenschaftsspeicher werden von oben nach unten ausgewertet, wobei jedoch die letzte Definition für die jeweilige Eigenschaft verwendet wird. Das System ist

standardmäßig so konfiguriert, dass die in einer lösungsspezifischen Eigenschaftendatei namens `solution.properties` definierten Eigenschaften (Position dieser Datei ist das Lösungsverzeichnis) die korrespondierenden Eigenschaften in der systemweiten Datei `global.properties` außer Kraft setzen.

Anmerkung: Bestimmte Systemeigenschaften und Java-Eigenschaften sind schreibgeschützt. Diese schreibgeschützten Eigenschaften werden in den entsprechenden Eigenschaftsspeichern angezeigt. Ein Versuch, diese Eigenschaften zu modifizieren, bleibt wirkungslos.

Lösungsansicht auswählen

In diesem Fenster können Sie eine Lösungsansicht auswählen. Das Menü enthält nur diejenigen Lösungsansichten, für die Sie eine der Zugriffsberechtigungen *Lesen*, *Ausführen*, *Konfigurationsadministrator* oder *Administrator* besitzen. Falls Sie keinen Zugriff auf die erstellten Lösungsansichten besitzen, ist dieses Fenster leer. Klicken Sie nach Auswahl einer Ansicht auf **Definieren**.

Nachdem Sie eine Lösungsansicht ausgewählt haben, können Sie Eigenschaften verwalten, indem Sie auf die Registerkarten für die anderen Eigenschaften klicken, z. B. **Lösungseigenschaften** und **Globale Eigenschaften**.

Lösungseigenschaften

In diesem Fenster können Sie Eigenschaften in der Liste "Lösungseigenschaften" hinzufügen, bearbeiten und löschen.

Globale Eigenschaften

In diesem Fenster können Sie globale Eigenschaften hinzufügen, bearbeiten und löschen.

Java-Eigenschaften

In diesem Fenster können Sie Java-Eigenschaften hinzufügen, bearbeiten und löschen.

Systemeigenschaften

In diesem Fenster können Sie Systemeigenschaften hinzufügen, bearbeiten und löschen.

Kennwortspeicher

In diesem Fenster können Sie Eigenschaften im Kennwortspeicher hinzufügen, bearbeiten und löschen.

Eigenschaftsspeicher des Benutzers

In diesem Fenster können Sie Eigenschaften in der Liste "Eigenschaftsspeicher des Benutzers" hinzufügen, bearbeiten und löschen.

Das Dropdown-Menü "Eigenschaftsspeicher" enthält eine Liste der Eigenschaftsspeicher, die vom Benutzer konfiguriert wurden. Globale Eigenschaften, Lösungseigenschaften, Java-Eigenschaften sowie Kennwortspeichereigenschaften sind in dieser Liste nicht enthalten. Wählen Sie den Eigenschaftsspeicher aus, dessen

zugehörige Eigenschaften Sie anzeigen, hinzufügen, bearbeiten oder löschen wollen.

Protokollverwaltung

Nachstehend erhalten Sie Informationen zur Protokollverwaltung und deren Verwendung.

Falls Sie die Kategorie **Erweitert** im Navigationsbereich von AMC noch nicht erweitert haben, führen Sie dies nun aus. Um Protokolldateien für alle Fertigungslinien, für eine bestimmte Fertigungslinie oder für einen bestimmten Datumsbereich zu löschen, klicken Sie auf **Protokollverwaltung**. Wenn Sie eine neue Lösungsansicht auswählen, können Sie auf **Neuanzeige** klicken. Durch Klicken auf **Neuanzeige** werden alle Fertigungslinien aufgelistet, die zur soeben ausgewählten Lösungsansicht gehören.

In diesem Fenster können Sie den Namen einer Lösungsansicht auswählen. Die für den Löschvorgang aufgelisteten Fertigungslinien stammen aus der von Ihnen ausgewählten Lösungsansicht. Sie können Protokolldateien für alle Fertigungslinien oder für eine bestimmte Fertigungslinie löschen. Die zu löschenden Protokolle können Sie aber auch anhand des Datums angeben. So verwalten Sie das Anzeigen und Löschen von Protokollen:

1. Wählen Sie im Menü **Lösungsansicht** die Lösungsansicht mit den Fertigungslinien aus, deren Protokolle Sie bereinigen wollen.
2. Führen Sie im Abschnitt **Komponente auswählen** eine der folgenden Aktionen aus:
 - Wählen Sie das Optionsfeld **Alle Fertigungslinien** aus, wenn Sie die Protokolle aller Fertigungslinien in der ausgewählten Lösungsansicht löschen wollen.
 - Wählen Sie das Optionsfeld **Bestimmte Fertigungslinie** aus, wenn Sie nur die zu einer bestimmten Fertigungslinie gehörenden Protokolle löschen wollen.
3. Falls Sie die Option **Bestimmte Fertigungslinie** ausgewählt haben, müssen Sie im Menü die Fertigungslinie auswählen, deren Protokolle Sie löschen wollen.
4. Führen Sie im Abschnitt **Protokolldateien anzeigen** eine der folgenden Aktionen aus:
 - Wählen Sie **Alle** aus, wenn Sie alle Protokolle der ausgewählten Fertigungslinie(n) löschen wollen.
 - Verwenden Sie die Optionen **Startdatum** und **Enddatum**, wenn Sie Protokolle löschen wollen, die aus einem bestimmten Datumsbereich stammen. Dann werden die Protokolle gelöscht, die zwischen den beiden angegebenen Datumswerten erstellt wurden. Geben Sie in den Datumsfeldern die gewünschten Werte ein. Das Format der Werte ist von der Ländereinstellung abhängig. Nach Auswahl der Schaltfläche "Kalender" können Sie zur Angabe eines Datums auch eine Auswahl in einem Kalender treffen.
 - Wenn Sie bei Servern aus Vorgängerversionen von IBM Security Directory Integrator Protokolle löschen möchten, die vor einem bestimmten Datum erstellt wurden, wählen Sie die Option **Enddatum** aus. Dann werden alle Protokolle gelöscht, die älter als das angegebene Datum sind.
 - Wählen Sie das Optionsfeld **Neueste behalten** aus, wenn die aktuellsten Protokolle erhalten bleiben sollen. Geben Sie an, wie viele der neuesten Protokolle nicht gelöscht werden sollen. Mit dieser Option wird angegeben, dass die neuesten Protokolldateien erhalten bleiben und die anderen aufgelistet werden sollen. Mithilfe des Bearbeitungsfelds geben Sie die Grenze für die

neuesten Dateien an. Falls Sie den Wert 20 eingeben, wird AMC hierdurch angewiesen, die neuesten 20 Protokolldateien beizubehalten und die übrigen Dateien in der Tabelle aufzulisten, damit sie für den Löschvorgang verfügbar sind. Falls Sie den Wert 10 eingeben, werden die neuesten 10 Protokolle nicht gelöscht.

5. In der Tabelle **Protokolldateien** wird eine Liste von Protokolldateien angezeigt. Wählen Sie in der Spalte "Auswählen" alle Protokolle aus, die Sie löschen wollen, und klicken Sie auf **Löschen**. Im Menü **Aktion auswählen** können Sie Optionen für die folgenden Aktionen auswählen:

- Daten exportieren
- Alle ausgewählten ändern
- Tabelle komprimieren
- Wiederherstellen

Nachdem Sie eine dieser Optionen ausgewählt haben, klicken Sie auf **Los**.

6. Wählen Sie in der Anzeige, die infolge der ausgewählten Kriterien ausgegeben wird, die zu löschenden Protokolle aus und klicken Sie auf **Löschen**, um die angegebenen Protokolle zu entfernen. Nachdem Sie alle gewünschten Protokolle gelöscht haben, klicken Sie auf **Schließen**, um dieses Fenster zu verlassen.

Bevorzugte Lösungsansichten

In der Anzeige **Bevorzugte Lösungsansichten** können Sie die Lösungsansichten auswählen, die standardmäßig im Überwachungsfenster geladen werden sollen.

Die bevorzugten Konfigurationen werden beim Öffnen des Fensters "Status überwachen" standardmäßig angezeigt. Falls keine Ansichten definiert sind, wird in dieser Anzeige lediglich die Nachricht ausgegeben, dass keine Ansichten vorhanden sind. Sobald eine Reihe von Ansichten definiert wurde, kann der Benutzer festlegen, welche Ansichten als Standardansichten angezeigt werden sollen. Diese Anzeige kann von jedem Benutzer aufgerufen werden, dem vom Superadministrator eine Reihe von Ansichten zugeordnet wurden.

Sie können eine Lösungsansicht als bevorzugte Lösungsansicht definieren, indem Sie ihr Markierungsfeld in der Spalte **Auswählen** auswählen und dann auf die Option für die Aktivierung als bevorzugte Ansicht klicken.

Den bevorzugten Status einer Lösungsansicht können Sie hingegen inaktivieren, indem Sie das Markierungsfeld der Lösungsansicht in der Spalte **Auswählen** auswählen und dann auf die Option für die Inaktivierung als bevorzugte Ansicht klicken.

Befehlszeilendienstprogramme für AMC und Action Manager

Nachstehend erhalten Sie Informationen zu den Befehlszeilendienstprogrammen für AMC und Action Manager.

AMC und das zugehörige Produkt "Action Manager" umfassen eine Reihe von Befehlszeilendienstprogrammen. Diese Befehlszeilendienstprogramme sind hilfreich, wenn die WAR-Datei für AMC installiert, deinstalliert oder erneut installiert werden muss. Darüber hinaus gibt es Scripts für Sicherung und Wiederherstellung sowie ein Migrationsscript. Das Migrationsscript dient zur Migration auf künftige Versionen von AMC und Action Manager. Zur Migration vorheriger Versionen auf die aktuelle Version kann es nicht eingesetzt werden. Alle Scripts sind im Verzeichnis *tdi-installationsverzeichnis/bin/amc* installiert.

install Das Script "install.bat" (.sh) wird zum Implementieren des AMC-Konsolenmoduls unter ISC SE oder IBM Dashboard Application Services Hub verwendet. Das Script verwendet das Script setupCmdLine zum Festlegen der benötigten Umgebungsvariablen und das Script tdiISCHome zum Ermitteln der Position für die ISC-Laufzeit und des Typs der verwendeten Laufzeit (integrierte Webplattform oder IBM WebSphere Application Server). Dieses Script wird durch das Installationsprogramm aufgerufen.

Syntax: install

Für dieses Script können keine Parameter verwendet werden.

uninstall

Das Script "uninstall.bat" (.sh) deinstalliert das AMC-Konsolenmodul unter ISC SE oder IBM Dashboard Application Services Hub. Das Script verwendet das Script setupCmdLine zum Festlegen der benötigten Umgebungsvariablen und das Script tdiISCHome zum Ermitteln der Position für die ISC-Laufzeit und des Typs der verwendeten Laufzeit (integrierte Webplattform oder IBM WebSphere Application Server).

Syntax: uninstall

Für dieses Script können keine Parameter verwendet werden.

backupamc

Das Script "backupamc.bat" (.sh) sichert alle konfigurationsbezogenen Informationen von AMC (Konfigurationsdateien, Protokolle usw.). Innerhalb des Sicherungsverzeichnisses wird ein Ordner backup_tdiamc erstellt.

Syntax: backupamc [-d *ordner_für_sicherungserstellung*]

Falls die Option "-d" nicht angegeben wird, werden die Dateien in das Verzeichnis *tdi-installationsverzeichnis/bin/amc/ActionManager/backup_tdiamc* kopiert.

Die Sicherung umfasst die folgenden Dateien:

1. amc.properties
2. logging.properties
3. amcdbschema.xml
4. amcdbhandler.properties

restoreamc

Das Script "restoreamc.bat" (.sh) stellt die gesicherten Dateien in einer neuen AMC-Implementierung wieder her. Die gesicherten Dateien müssen zuvor mit dem Script "backupamc" erstellt worden sein, damit dieses Script verwendet werden kann.

Syntax: restoreamc

Für dieses Script können keine Parameter verwendet werden.

migrateamc

Dieses Script bietet einen einzigen Befehl für die Sicherung, Wiederherstellung, Deinstallation und Installation. Bei seiner Ausführung werden die alten AMC-Daten gesichert, das alte AMC-Plug-in-Archiv deinstalliert, die neue AMC-Version installiert und die alten AMC-Konfigurationsdaten wiederhergestellt.

Dieses Script setzt voraus, dass das neue AMC-Plug-in-Archiv in das Verzeichnis *tdi-installationsverzeichnis/amc* kopiert wird.

Syntax: `migrateamc.bat [-d sicherungsverzeichnis]`

start_tdiamc

Dieses Script ist ein komfortables Wrapperdienstprogramm zum Starten von AMC. Es führt einen internen Start der ISC-Laufzeit aus. Falls als Laufzeit die integrierte Webplattform verwendet wird, ruft das Script den Befehl `lwi start` auf. Wird hingegen IBM WebSphere Application Server als Laufzeit eingesetzt, ruft es den Befehl `startServer server1` auf. Vor dem Starten der ISC-Laufzeit ruft das Script den Befehl `startNetworkServer` auf, der das Starten der Derby-Datenbank im gesicherten Netzmodus vornimmt. Falls ein anderer Datenbanktyp als Derby verwendet wird, startet dieses Script lediglich die ISC-Laufzeit.

Auf Windows-Plattformen:

Syntax: `start_tdiamc [dienstname]`

Falls ein Dienstname übergeben wird, wird der Dienst gestartet, statt den Befehl "lwiStart" aufzurufen.

UNIX-Plattformen

Syntax: `start_tdiamc`

stop_tdiamc

Dieses Script ist ein komfortables Wrapperdienstprogramm zum Stoppen von AMC. Es führt einen internen Stopp der ISC-Laufzeit aus. Falls als Laufzeit die integrierte Webplattform verwendet wird, ruft das Script den Befehl `lwi stop` auf. Wird hingegen IBM WebSphere Application Server als Laufzeit eingesetzt, ruft es den Befehl `stopServer server1` auf. Nach der Ausführung des Befehls wird von diesem Script das Script `stopNetworkServer` aufgerufen, mit dem die Derby-Datenbank gestoppt wird. Falls ein anderer Datenbanktyp als Derby verwendet wird, stoppt dieses Script lediglich die ISC-Laufzeit.

Auf Windows-Plattformen:

Syntax: `stop_tdiamc [dienstname]`

Falls ein Dienstname übergeben wird, wird der Dienst gestoppt, statt den Befehl "lwiStop" aufzurufen.

UNIX-Plattformen

Syntax: `stop_tdiamc`

startAM

Action Manager wird mit dem Script "startAM.bat" (.sh) gestartet, das sich im Verzeichnis *tdi-installationsverzeichnis/bin/amc* befindet.

Anmerkung: Im Script ist der Klassenpfad für alle von Action Manager benötigten JAR-Dateien definiert. Es gibt zwei Variablen namens CLASSPATH und DB_CLASSPATH. Die Variable DB_CLASSPATH enthält die nach Pfaden getrennte Liste der JAR-Dateien, die für die Erzielung der JDBC-Konnektivität mit der Datenbank erforderlich sind. Wenn AMC für die Verwendung von Oracle, MS SQL Server oder DB2 konfiguriert ist, sollten die entsprechenden JAR-Dateien dieser Datenbanken für JDBC zur Variablen DB_CLASSPATH hinzugefügt werden.

Bei Windows kann für das Script optional ein Parameter für den Dienstnamen verwendet werden, der zum Starten eines bereits registrierten Dienstes verwendet werden kann:

startAM.bat [dienstname]

stopAM

Action Manager wird mit dem Script "stopAM.bat" (.sh) gestoppt, das sich im Verzeichnis *tdi-installationsverzeichnis/bin/amc* befindet. Dieses Script verwendet die Prozess-ID der gestarteten Action Manager-Instanz, um diese zu beenden. Die Prozess-ID wird durch das Script "startAM" abgerufen und in einer Datei gespeichert, die dann durch das Script "stopAM" gelesen wird.

Bei Windows kann für das Script optional ein Parameter für den Dienstnamen verwendet werden, der zum Stoppen eines bereits registrierten Dienstes verwendet werden kann:

stopAM.bat [dienstname]

startNetworkServer

Mit diesem Script wird der Derby-Datenbankserver im Netzmodus an Port 1528 gestartet. Der ausgewählte Port weicht vom Standardport für Derby ab.

Syntax: startNetworkServer

stopNetworkServer

Mit diesem Script wird der Derby-Datenbankserver im Netzmodus gestoppt.

Syntax: stopNetworkServer

setDBType

Mit diesem Script wird der Typ der verwendeten Datenbank festgelegt. Das Script definiert den Wert für die Eigenschaft namens `DB_TYPE`. Falls die Eigenschaft `DB_TYPE` auf "Derby" gesetzt ist, wird bei der Ausführung des Scripts `startNetworkServer` die Derby-Datenbank auf dem Host und an dem Port gestartet, die in der Scriptdatei `startNetworkServer` angegeben sind. Das Script "setDBType" legt außerdem den Datenbankbenutzernamen und das zugehörige Kennwort fest. Der Datenbankbenutzername und das entsprechende Kennwort werden vom Script `startNetworkServer` zum Aktivieren des Sicherheitsmechanismus `BUILTIN` und zum Hinzufügen des Benutzers zur Liste der berechtigten Benutzer benötigt.

Das Script "setDBType" wird intern durch die Scripts `startNetworkServer` und `stopNetworkServer` aufgerufen, um die Eigenschaften `DB_TYPE` sowie `DB_USER` und `DB_PASSWORD` festzulegen.

backupamcdb

Dieses Script wird während der Migration einer AMC-Datenbank eingesetzt. Es sichert die AMC-Datenbank und exportiert die Daten in einem von IBM Security Directory Integrator definierten XML-Format. Dieses Script wird durch das Installationsprogramm aufgerufen, wenn Sie den Migrationspfad auswählen.

Syntax: backupamcdb -d *ordner_mit_amc-sicherung* -p *position_der_datei_amc.properties*

restoreamcdb

Dieses Script wird verwendet, um die AMC-Datenbank während der Migration wiederherzustellen. Es wird durch das Installationsprogramm aufgerufen, wenn Sie den Migrationspfad auswählen.

Syntax: `restoreamcdb -d ordner_mit_amc-sicherung -p position_der_datei_amc.properties`

backupam

Mit diesem Script werden die Eigenschaftendateien für Action Manager gesichert. Das Script sichert die Dateien `am_config.properties` und `am_logging.properties`.

Syntax: `backupam [-d sicherungsverzeichnis]`

Die archivierten Daten werden im Sicherungsordner erstellt. Falls die Option "-d" nicht angegeben ist, werden die Dateien in das Verzeichnis `tdi-installationsverzeichnis/bin/amc/ActionManager/backup_tdiamc` kopiert.

restoream

Mit diesem Script werden die Eigenschaftendateien für Action Manager wiederhergestellt, die mit dem Script "backupam" gesichert wurden. Das Wiederherstellungsscript stellt die Dateien `am_config.properties` und `am_logging.properties` wieder her.

Syntax: `restoream [-d sicherungsverzeichnis]`

Falls die Option "-d" nicht angegeben wird, werden die Dateien aus dem Verzeichnis `tdi-installationsverzeichnis/bin/amc/ActionManager/backup_tdiamc` kopiert.

setAMCRoles

Mit diesem Script werden dem Benutzer, der IBM Security Directory Integrator AMC installiert, die Rollen des ISC-Administrators (ISC admin) und des SDI-AMC-Administrators (SDI AMC Admin) zugeordnet. Dieses Script ist erstmalig ab IBM Security Directory Integrator 7.0 verfügbar.

Nachdem der Installationsbenutzer diese Rollen erhalten hat, besitzt er die Berechtigung, neue Benutzer zu erstellen und diesen die erforderlichen Rollen zu erteilen. Der Installationsbenutzer wird zum Administrator für das AMC-Konsolenmodul.

Syntax: `setAMCRole benutzername [betriebssystemgruppe]`

Der Parameter "betriebssystemgruppe" ist ein optionaler Parameter, der bei der Implementierung von AMC unter ISC SE verwendet werden kann.

tdimigam

Dieses Script wird zur Migration der Datei `am_config.properties` verwendet.

Der Befehl hat die folgende Syntax:

`tdimigam -f eigenschaftendatei [-b sicherungsdatei] [-n neue_datei] [-v] [-?]`

Hierbei gilt Folgendes:

- f eigenschaftendatei - Name der zu migrierenden Datei
- b sicherungsdatei - Ursprüngliche Datei unter dem angegebenen Namen sichern
- n neue_datei - Name für die migrierte Datei
- v - Ausführlichen Modus aktivieren
- ? - Verwendungsanweisung ausgeben

Die Protokollierung für diesen Befehl wird durch die Datei "tdimigam-Log4J.properties" gesteuert.

tdimigamc

Dieses Script wird zur Migration der Datei `amc_config.properties` verwendet. Es besitzt ähnliche Optionen wie die Scripts "tdimigam" und "tdimig-bl", die zur Migration der Dateien `am_config.properties` bzw. `global.properties` eingesetzt werden.

Der Befehl hat die folgende Syntax:

```
tdimigamc -f eigenschaftendatei [-b sicherungsdatei] [-n neue_datei] [-v] [-?]
```

Hierbei gilt Folgendes:

- f eigenschaftendatei - Name der zu migrierenden Datei
- b sicherungsdatei - Ursprüngliche Datei unter dem angegebenen Namen sichern
- n neue_datei - Name für die migrierte Datei
- v - Ausführlichen Modus aktivieren
- ? - Verwendungsanweisung ausgeben

Die Protokollierung für diesen Befehl wird durch die Datei "tdimigamc-Log4J.properties" gesteuert.

addAMCService

Mit diesem Script wird AMC als Service oder Dienst auf einem System hinzugefügt.

Syntax: `addAMCService servicename_bzw._dienstname`

Unter Windows registriert das Script die ausführbare Datei für den generischen Windows-Dienst (*tdi-installationsverzeichnis/bin/amc/amcwin-service.exe*) aus IBM Platform Integration Toolkit. Der generische Windows-Dienst verwendet die Konfigurationsdatei *tdi-installationsverzeichnis/bin/amc/amcwin-service.ini*. Diese Datei gibt den Namen des Dienstes sowie den Start- und Stoppbefehl an. Sie wird durch das Installationsprogramm oder das Script "addAMCService" automatisch mit Inhalt gefüllt.

Standardmäßig sieht diese Datei folgendermaßen aus:

```
[Service]
ServiceName=$service_name$
WorkingDirectory="$install_dir$\bin\amc"
StartCommand="$install_dir$\bin\amc\amcservice.bat" start amc am"
StopCommand="$install_dir$\bin\amc\amcservice.bat" stop amc am"
```

Dies bedeutet, dass der AMC-Dienst standardmäßig sowohl AMC als auch Action Manager ausführt.

Nach dem Aufruf von "addAMCService" können Sie die Datei ".ini" bearbeiten, um die durch den Dienst ausgeführten Komponenten (AMC plus Action Manager, ausschließlich AMC oder ausschließlich Action Manager) anzupassen.

Damit beispielsweise lediglich AMC ausgeführt wird, geben Sie den Start- und Stoppbefehl wie folgt an:

```
StartCommand="$install_dir$\bin\amc\amcservice.bat" start amc"
StopCommand="$install_dir$\bin\amc\amcservice.bat" stop amc"
```

Zum Starten und Stoppen des Dienstes verwenden Sie das Dienstprogramm "Dienste" in der grafischen Benutzerschnittstelle (**Systemsteuerung -> Verwaltung -> Dienste**) oder das Befehlszeilentool für den Dienstcontroller:

```
sc start <dienstname>
sc stop <dienstname>
```

Bitte beachten Sie, dass der Anzeigename des registrierten Dienstes im Dienstprogramm "Dienste" wie folgt lautet:

```
IBM Tivoli Directory Integrator Administration and Monitoring Console - myamc
```

Hierbei steht "myamc" für den Dienstnamen, den Sie als Argument für das Script "addAMCService.bat" angegeben haben.

Unter UNIX hängt das Script eine Zeile ähnlich der Folgenden am Ende der Systemdatei /etc/inittab an:

```
<dienstname>::once:<installationsverzeichnis>/bin/amc/amcservice.sh" start amc am
```

Verwenden Sie zum Stoppen des Service das Script "amcservice.sh" aus dem Ordner *TDI_installationsverzeichnis/bin/amc*:

```
amcservice.sh stop amc am
```

Oder geben Sie einfach den Befehl

```
amcservice.sh stop amc
```

an, falls der Service lediglich AMC ausführt.

deleteAMCService

Dieses Script wird verwendet, um AMC als Service aus einem System zu entfernen.

Syntax: `deleteAMCService servicename`

setDerbyProps

Dieses Script legt die erforderlichen Eigenschaften der Derby-Datenbank fest, die von den Scripts `startNetworkServer` und `stopNetworkServer` verwendet werden.

Syntax: `setDerbyProps`

amcservice

Dieses Script startet/stoppt die gesamte AMC-Konfiguration. Die folgenden Konfigurationen werden unterstützt:

- AMC plus Action Manager
- Nur AMC
- Nur Action Manager

Das Script ruft intern "start_tdiadc"/"stop_tdiadc" und "startAM"/"stopAM" auf. Es ist zur Verwendung bei der Registrierung eines Betriebssystemservice gedacht.

Syntax: `amcservice [start|stop] [amc] [am]`

Beispiele:

```
amcservice start amc
amcservice stop amc am
```

Beispielablauf der Erstellung von Lösungsansicht und Regeln

Sie können die Schritte anzeigen, die für das Erstellen einer Lösungsansicht, das Konfigurieren einer Regel und das Auslösen der Regel in Action Manager erforderlich sind.

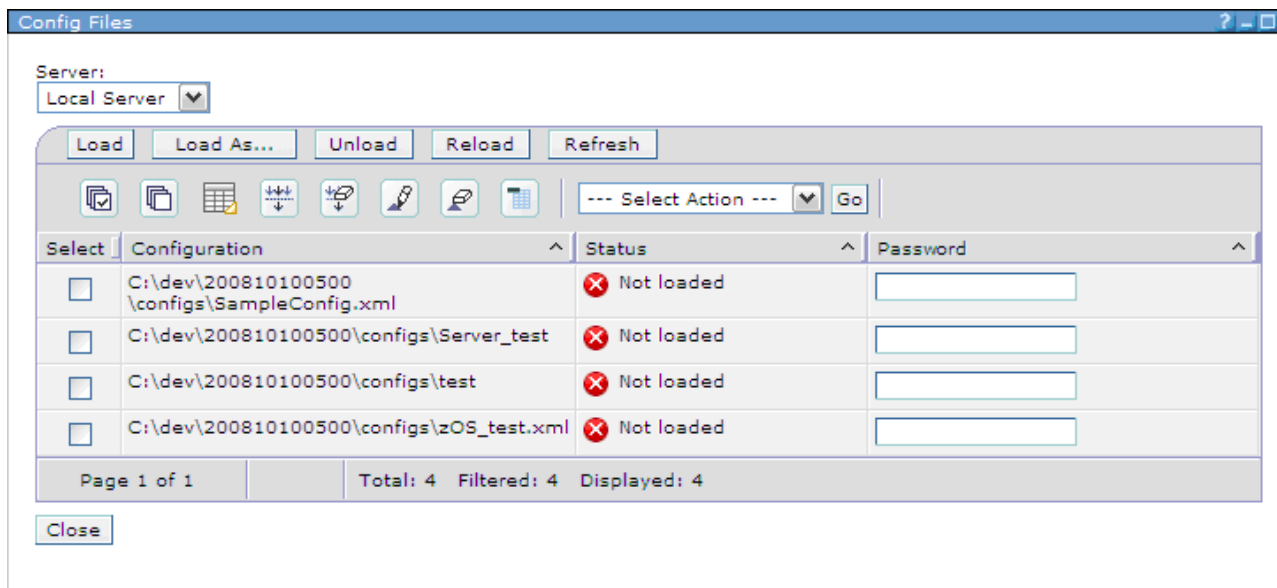
Die Anweisungen setzen voraus, dass IBM Security Directory Integrator zusammen mit AMC installiert ist. Die in diesem Beispiel verwendete Konfiguration

SampleConfig.xml ist diejenige Konfiguration, die mit dem Lernprogramm erstellt wird, das in der Veröffentlichung *Erste Schritte* unter "IBM Security Directory Integrator - Einführung" > "Erste Fertigungslinie erstellen" verfügbar ist. Diese Konfiguration sollte in den Ordner *tdi-installationsverzeichnis*/configs kopiert werden. Hierbei steht *tdi-installationsverzeichnis* für das Installationsverzeichnis von IBM Security Directory Integrator. Diese Lösung liest Daten aus der Datei *examples/Tutorial/People.csv* und schreibt Daten in die Datei *examples/Tutorial/Output.xml*.

In diesem Abschnitt werden alle Schritte aufgeführt, die für das Erstellen einer Konfigurationsansicht, das Konfigurieren von Regeln und das Auslösen dieser Regeln in Action Manager erforderlich sind. Die Anweisungen setzen voraus, dass IBM Security Directory Integrator zusammen mit AMC installiert ist. Die Beispielkonfiguration (SampleConfig.xml) ist zusammen mit den zugehörigen Dateien im Downloadabschnitt verfügbar und sollte in den Ordner *tdi-installationsverzeichnis*/configs kopiert werden. Diese Lösung liest Daten aus der Datei "sample.csv" und schreibt Daten in die Datei "sample.xml".

Schritte

1. Starten Sie den IBM Security Directory Integrator-Server im Dämonmodus.
2. Starten Sie AMC mit dem Befehl *tdi-installationsverzeichnis\bin\amc\start_tdiamc.bat*
3. Melden Sie sich unter Verwendung der URL mit der Syntax `http://hostname:port/ibm/console` sowie mit dem Standardbenutzernamen und dem Standardkennwort bei der AMC-Konsole an.
4. Wählen Sie nach der Anmeldung bei der AMC-Konsole im Navigationsfenster den Eintrag "Server" und anschließend den Server aus, den Sie verwenden wollen. Wählen Sie die Schaltfläche **Konfigurationsdateien** aus. Daraufhin wird die folgende Anzeige aufgerufen:



Wählen Sie die Datei *SampleConfig.xml* aus und klicken Sie auf die Schaltfläche **Laden**.

5. Wählen Sie im Navigationsfenster den Link "Lösungsansicht" aus, um für die geladene Konfiguration eine Lösungsansicht zu erstellen. Wählen Sie die Schalt-

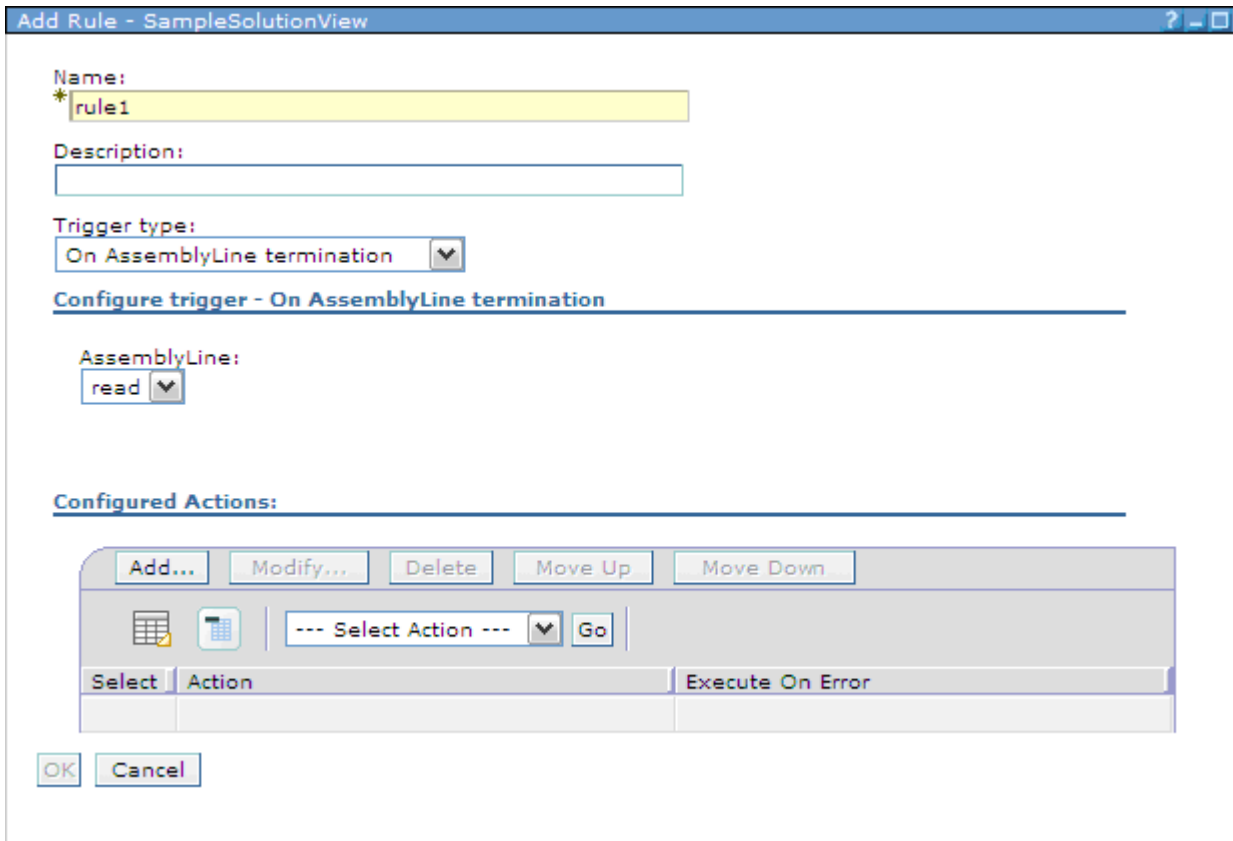
fläche "Hinzufügen" aus. Daraufhin wird die folgende Anzeige aufgerufen:

The screenshot shows a dialog box titled "Add Solution View". It contains the following elements:

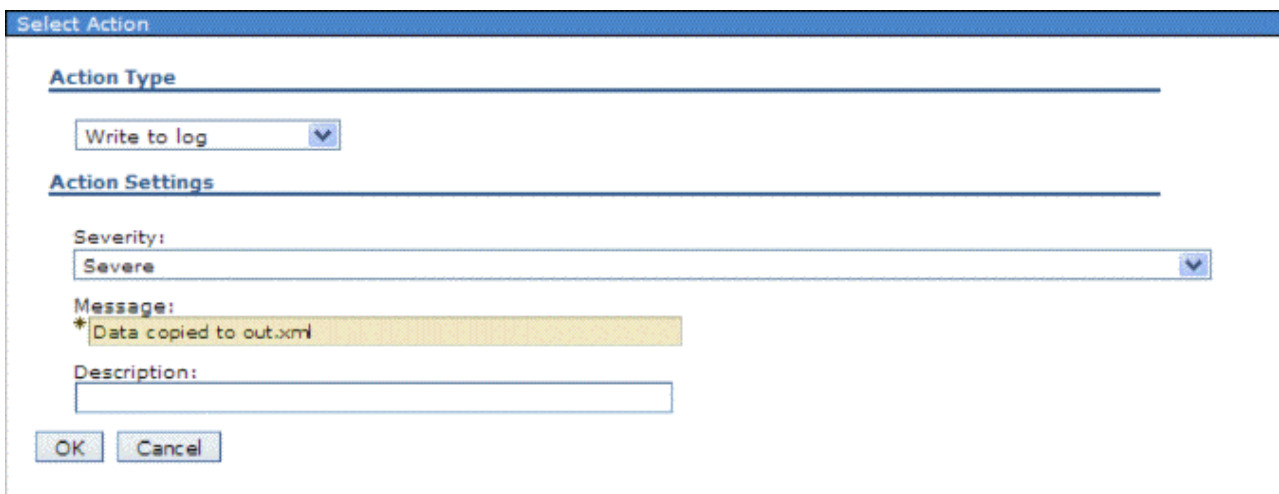
- Solution View Name:** A text box containing "SampleSolutionView".
- Description:** An empty text box.
- Server:** A dropdown menu showing "Local Server".
- Configs:** A dropdown menu showing "C__dev_200810100500_configs_SampleConfig.xml" and a "View Config Files" button.
- Simple:** A radio button is selected. Below it are four options:
 - Auto Update
 - Add Solution View from published solution.
 - Add Solution View with all AssemblyLines exposed.
 - Add Solution View with all AssemblyLines and all properties exposed.
 - Add Solution View with all AssemblyLines and all User properties exposed.
- Advanced:** A radio button is not selected, with the text "Configure the properties to create a Solution View."
- Buttons:** "OK" and "Cancel" buttons at the bottom.

Fügen Sie einen geeigneten Namen für die Konfiguration hinzu (z. B. "SampleSV"). Nach Auswahl von "OK" wird die Nachricht angezeigt, dass die Lösungsansicht "SampleSV" erfolgreich erstellt wurde.

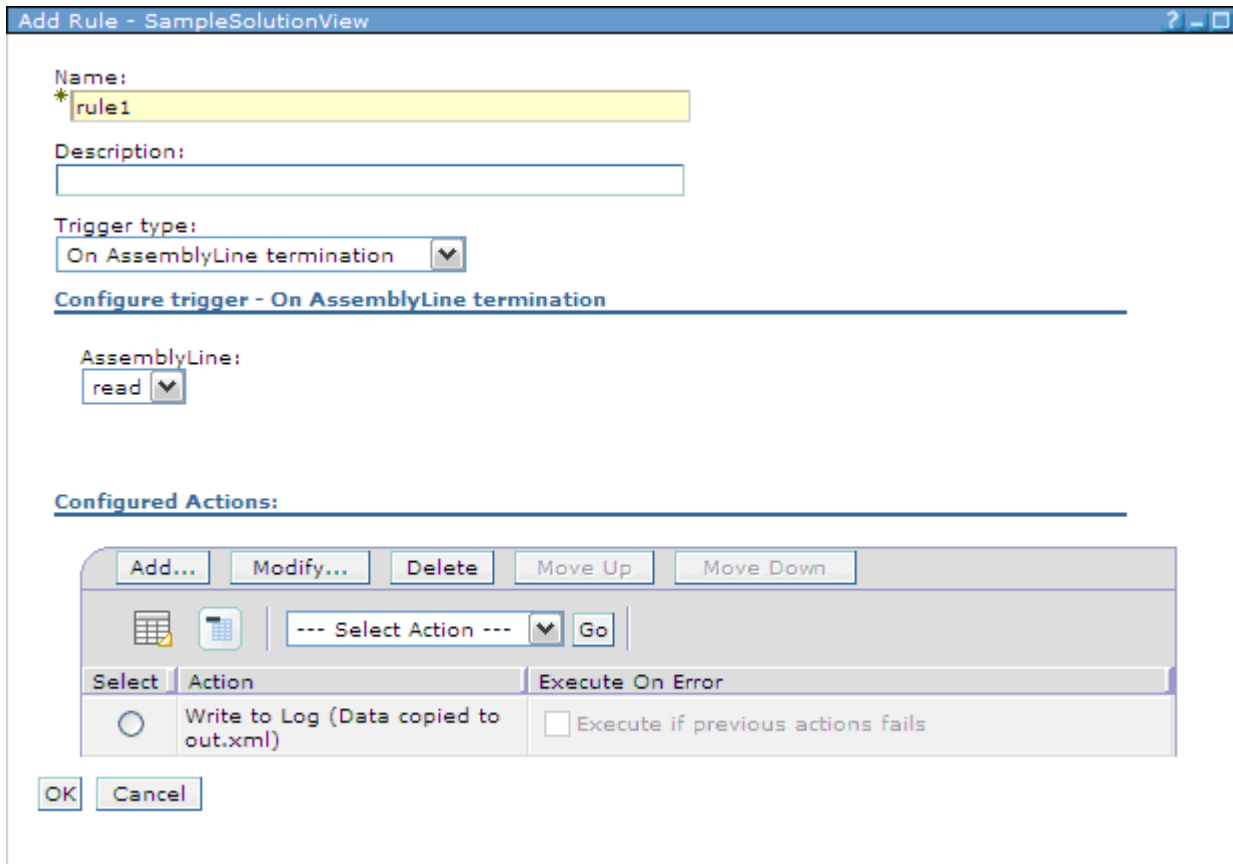
6. Wählen Sie im Navigationsfenster den Link "Action Manager" aus, um die Konfigurationsanzeige für die Action Manager-Regeln aufzurufen. Wählen Sie im Dropdown-Feld "Lösungsansichten auswählen" den Eintrag "SampleConfig-View" aus. Klicken Sie im Abschnitt "Konfigurierte Regeln" auf die Schaltfläche **Hinzufügen**. Daraufhin wird die folgende Anzeige aufgerufen:



Geben Sie einen Namen ein, z. B. "Rule1". Wählen Sie den Auslösertyp "Bei Beendigung der Fertigungslinie" aus und klicken Sie im Abschnitt "Konfigurierte Aktionen" auf die Schaltfläche **Hinzufügen**.



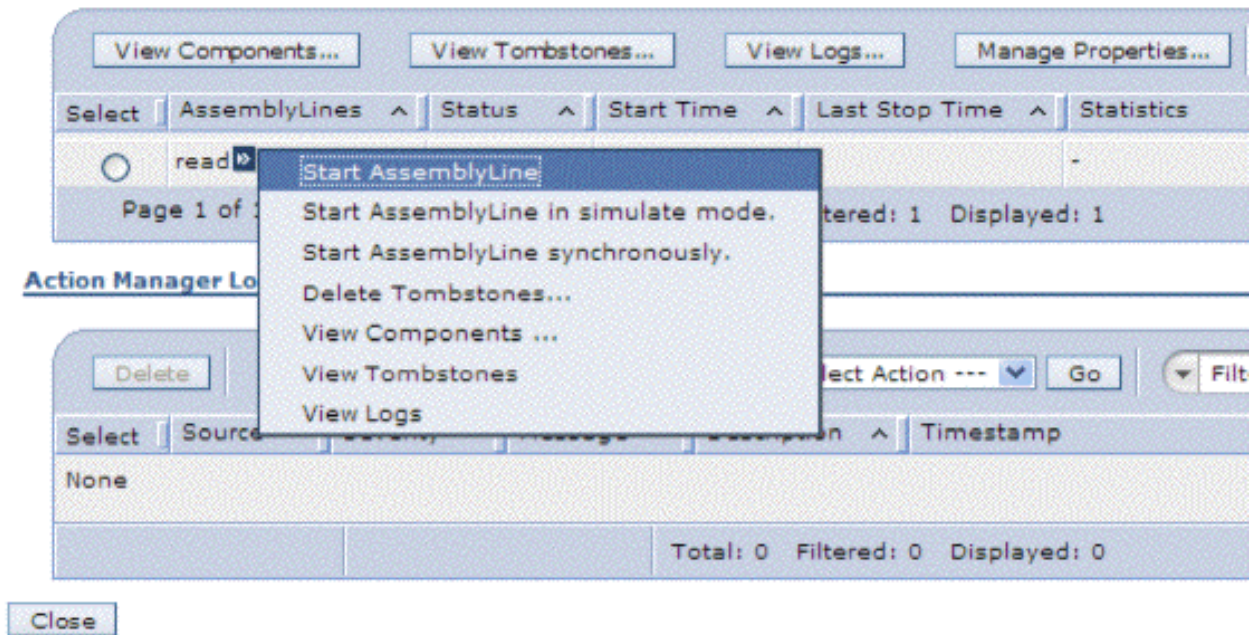
Wählen Sie im kombinierten Feld "Aktionstyp" der Anzeige "Aktion auswählen" die Option "In Protokoll schreiben" aus. Fügen Sie im Textfeld "Nachricht" den Text "data copied to out.xml" hinzu und klicken Sie auf **OK**. Die folgende Regelanzeige wird ausgegeben.



Wenn Sie auf **OK** klicken, schließen Sie hierdurch die Erstellung dieser Regel und der für dieses Beispiel erforderlichen AMC-Konfiguration ab.

7. Starten Sie Action Manager mit dem Script `tdi-installationsverzeichnis\bin\amc\startAM.bat`. Für die Regel "Rule1" wird nun ein Thread erstellt, der auf die Beendigung der angegebenen Fertigungslinie warten sollte.
8. Zum Auslösen dieser Regel muss die Fertigungslinie "read" von "SampleConfig.xml" ausgeführt werden. Wählen Sie im Navigationsfenster den Link **Status überwachen** aus. Wählen Sie in der Anzeige "Status überwachen" den Eintrag "SampleSV" aus und klicken Sie auf die Schaltfläche **Details der Lösungansicht**. Daraufhin wird die folgende Anzeige ausgegeben:

AssemblyLines



- Starten Sie die Fertigungslinie durch Auswahl der oben dargestellten Option. Die Regel wird ausgelöst und in der Action Manager-Konsole wird der folgende Status angezeigt.

```

C:\WINDOWS\system32\cmd.exe - startAM.bat
2008-10-16 17:05:01 con.ibm.di.amc.actionmanager.ServerConfigHandler startAMCServerModificationListe
nerThread
INFO: CTGDJB659I Started AMCServerModificationListener thread.
2008-10-16 17:05:01 con.ibm.di.amc.actionmanager.api.AMService initialize
INFO: AM.RMI.REGISTRY.SYSTEM.SECURITYMANAGER.NULL
2008-10-16 17:05:01 con.ibm.di.amc.actionmanager.api.AMService initialize
INFO: AM.RMI.REGISTRY.RESETTING.SYSTEM.SECURITYMANAGER.NULL
2008-10-16 17:05:01 con.ibm.di.amc.actionmanager.api.AMService initialize
INFO: CTGDJB720I The Action Manager RMI registry is started on port 13104.
2008-10-16 17:05:01 con.ibm.di.amc.actionmanager.AMHandler startHealthALManagerThread
INFO: CTGDJB512I Started Health AssemblyLine Manager.
2008-10-16 17:05:01 con.ibm.di.amc.actionmanager.AMHandler main
INFO: CTGDJB511I Action Manager initialization completed.
2008-10-16 17:05:01 con.ibm.di.amc.actionmanager.AMHandler startDatabaseModificationListener
INFO: CTGDJB532I Database modification listener started.
2008-10-16 17:06:23 con.ibm.di.amc.actionmanager.AssemblyLineTerminateListener triggerRule
INFO: CTGDJB549I Rule rule1 triggered.
2008-10-16 17:06:24 con.ibm.di.amc.actionmanager.RuleExecutionManager writeToLog
SEVERE: CTGDJB622I [Message]: Data copied to out.xml
[Description]: None

2008-10-16 17:06:24 con.ibm.di.amc.actionmanager.RuleExecutionManager execute
INFO: CTGDJB615I Action successfully executed.
[Solution View]: SampleSolutionView
[Rule]: rule1
[Action Order]: 2
[Action Type]: WRITE_LOG

```

In AMC wird die nachfolgend dargestellte Tabelle "Action Manager-Protokolle" in der Anzeige "Details der Lösungsansicht" angezeigt:

AssemblyLines

View Components...		View Tombstones...		View Logs...		Manage Properties...	
						--- Select Action ---	Go
Select	AssemblyLines	Status	Start Time	Last Stop Ti...	Statistics		
<input checked="" type="checkbox"/>	read	Stopped	-	-	-		
Page 1 of 1		Total: 1 Filtered: 1 Displayed: 1					

Action Manager Logs

Delete								
							--- Select Action ---	Go
Select	Source	Severity	Message	Description	Timestamp			
<input type="checkbox"/>	rule1	SEVERE	Data copied to out.xml	None	16.10.2008 17:06:24			
Page 1 of 1		Total: 1 Filtered: 1 Displayed: 1						

Close

Kapitel 17. Touchpoint-Server

Mit dem Touchpoint-Server können Sie Zugriff auf IBM Security Directory Integrator-Komponenten bereitstellen. Nachstehend erhalten Sie Informationen zu seiner Implementierung.

Der Touchpoint-Server ermöglicht den Zugriff auf IBM Security Directory Integrator-Komponenten (Connectors und Fertigungslinien) über das ReSTful-Kommunikationsprotokoll. Clients senden HTTP-Anforderungen an den Touchpoint-Server, um von einem IBM Security Directory Integrator-Server die Erstellung einer Touchpoint-Instanz anzufordern, mit der die Clients Daten austauschen können.

Eine Touchpoint-Instanz wird unter Verwendung von IBM Security Directory Integrator-Standardkomponenten implementiert. Sie ermöglicht HTTP-basierten Clients den Zugriff auf Drittanbietersysteme, mit denen IBM Security Directory Integrator kommunizieren kann. In diesem Kontext ist die Touchpoint-Instanz mit einer Art "Proxy" zwischen einer Clientanwendung und einem fernen Service vergleichbar, da sie Clients die Verwendung eines einheitlichen Protokolls für die Kommunikation mit einer Vielzahl von Systemen ermöglicht, die keine HTTP-basierte Schnittstelle besitzen.

Touchpoint-Konzepte

Nachstehend erhalten Sie Informationen zu den verschiedenen Touchpoint-Konzepten.

Das Touchpoint-Protokoll ist im Wesentlichen ein Bereitstellungsprotokoll, das den Zugriff auf Connectors und Fertigungslinien von IBM Security Directory Integrator über HTTP ermöglicht. Bei der Erstellung einer Touchpoint-Instanz erhalten Sie einen "Proxy", über den Sie mit einem fernen System arbeiten können. Sobald die Touchpoint-Instanz erstellt und konfiguriert wurde, senden Sie ausschließlich HTTP-Anforderungen und sind von den Spezifikationen dieses Systems völlig unabhängig.

Die nachfolgenden Abschnitte enthalten Details zu den verschiedenen Konzepten in Verbindung mit Touchpoint-Instanzen.

Touchpoint-Server

Mit dem Touchpoint-Server können Sie verschiedene Tasks ausführen, beispielsweise Informationen speichern oder Instanzen steuern. Nachstehend erhalten Sie Informationen zur Verwendung des Touchpoint-Servers.

Der Touchpoint-Server ist die Anwendung, die Informationen zu den in der jeweiligen Domäne definierten Touchpoint-Instanzen speichert. Der Touchpoint-Server wird verwendet, um die fernen Touchpoint-Instanzen zu steuern. Er ist dafür zuständig, dass diese konfiguriert, gestartet und gestoppt werden. Der Touchpoint-Server wird als Service bereitgestellt, der im IBM Security Directory Integrator-Server ausgeführt wird.

Clients des Touchpoint-Servers verwenden das Atom Publishing Protocol, um auf Details für die Touchpoint-Provider, die Touchpoint-Typen und die Touchpoint-Ins-

tanzen zuzugreifen. Weitere Einzelangaben über das definierte Schema enthält der Abschnitt „Touchpoint-Schema“ auf Seite 348.

Touchpoint-Provider

Mit dem Touchpoint-Provider können Sie die Instanzen erstellen. Nachstehend erhalten Sie Informationen zur Verwendung des Touchpoint-Providers und zu den Unterschieden zwischen einem Touchpoint-Server und einem Touchpoint-Provider.

Ein Touchpoint-Provider ist ein Server, auf dem der Touchpoint-Server die Touchpoint-Instanzen erstellt. Bei TDI kann jeder IBM Security Directory Integrator-Server der Version 7.1 oder höher als Touchpoint-Provider verwendet werden. Der Touchpoint-Server kann nur mit IBM Security Directory Integrator-Servern arbeiten, da ein anderer Touchpoint-Provider nicht unterstützt wird.

Der Touchpoint-Server wird als Add-on für den IBM Security Directory Integrator-Server ausgeliefert. Er wird in der JVM des Servers ausgeführt. Hierdurch kann er mit dem Server über die lokale Server-API kommunizieren. Aus diesem Grund wurde der IBM Security Directory Integrator-Server standardmäßig als lokaler Touchpoint-Provider registriert. Um einen Touchpoint-Provider zu registrieren, der einen fernen IBM Security Directory Integrator-Server darstellt, müssen Sie die RMI-Einstellungen des fernen Servers verwenden. Die Registrierung erfolgt sowohl für einen lokalen als auch einen fernen Server in der Standardkonfigurationsdatei des Touchpoint-Servers.

Anmerkung: Gegenwärtig enthalten weder der Konfigurationseditor noch das Webverwaltungstool (AMC) Benutzerschnittstellenanzeigen für die Konfiguration des Touchpoint-Servers oder der Touchpoint-Provider. Die gesamte Konfiguration muss über XML-Konfigurationsdateien erfolgen.

Weitere Details enthält der Abschnitt „Touchpoint-Konfiguration“ auf Seite 352.

Sobald der Touchpoint-Provider einmal konfiguriert wurde, kann er über die Atom-Schnittstellen nicht mehr geändert werden. Außerdem sind in der Atom-Schnittstelle einige der Details verdeckt, die die Verbindung zum fernen IBM Security Directory Integrator-Server angeben. Diese Details werden ausschließlich vom Touchpoint-Server für die Kommunikation mit dem Touchpoint-Provider verwendet und sollen für die Clients des Protokolls nicht erkennbar sein.

Touchpoint-Typ

Nachstehend erhalten Sie Informationen zu den Kategorien von Touchpoint-Typen.

Ein Touchpoint-Typ ist eine abstrakte Darstellung, die Metainformationen zu jeder Touchpoint-Instanz bereitstellt und deren Verhalten bestimmt. Jede Touchpoint-Instanz besitzt genau 1 Touchpoint-Typ. Die Anzahl der Touchpoint-Instanzen, die für einen bestimmten Typ erstellt werden können, ist hingegen nicht begrenzt.

Es gibt drei Kategorien von Touchpoint-Typen:

Standardtyp

Diese Kategorie entspricht den IBM Security Directory Integrator-Connectors, die durch den ausgewählten Touchpoint-Provider unterstützt werden, und beginnt mit dem Präfix `system`. Jeder Touchpoint-Provider besitzt eine eigene Gruppe von Touchpoint-Standardtypen, da er verschiedene Connectors bereitstellt. Durch die Auswahl eines dieser Typen geben Sie an, dass die Touchpoint-Basisschablone für die Struktur ihrer Touchpoint-Instanz

zugrunde gelegt wird (Details über Schablonen enthält der Abschnitt „Touchpoint-Schablone“ auf Seite 342). Darüber hinaus bestimmt der ausgewählte Typ die Vererbung des Serviceconnectors der Schablone (der Connector, der mit einem Drittanbietersystem arbeitet). Falls Sie beispielsweise Daten aus einem Managementsystem für relationale Datenbanken lesen müssen, kann der JDBC-Connector verwendet werden. Daher erstellen Sie eine Touchpoint-Instanz mit dem Typ `system:/Connectors/ibmdi.JDBC` und konfigurieren sie entsprechend. Diese Touchpoint-Typen unterstützen nur die Touchpoint-Rollen "Provider" und "Initiator".

Angepasster Typ

Diese Kategorie stellt angepasste Touchpoint-Schablonen dar, die durch Sie bereitgestellt werden und am Präfix `file` erkennbar sind. Statt die mit IBM Security Directory Integrator ausgelieferten Basisschablonen zugrunde zu legen, können Sie eigene Schablonen erstellen, die das Touchpoint-Verhalten gemäß Ihren Anforderungen optimieren. Dafür bieten diese Schablonen jedoch nicht die gesamte Flexibilität, die durch die Basisschablone bereitgestellt wird. Sobald eine angepasste Schablone erstellt wurde, kann der Typ ihrer Connectors nicht mehr geändert werden. Dies schränkt die erstellten Touchpoint-Instanzen dahingehend ein, dass immer mit demselben Typ eines fernen Systems gearbeitet werden muss. Den angepassten Touchpoint-Typ können Sie beispielsweise einsetzen, wenn Sie mit mehreren Datenquellen arbeiten müssen, z. B. Daten aus einem Managementsystem für relationale Datenbanken lesen und Informationen aus einem LDAP-Server hinzufügen wollen. Dies kann mit einer einzigen Touchpoint-Instanz, die auf der Touchpoint-Basisschablone fußt, nicht realisiert werden. Zur Lösung können Sie eine angepasste Touchpoint-Schablone erstellen und ihren entsprechenden Touchpoint-Typ (z. B. `file:/schablonendateiname.xml`) für die Erstellung einer Touchpoint-Instanz verwenden. Die einzige Einschränkung besteht darin, dass alle anschließend aus diesem Typ erstellten Touchpoint-Instanzen ebenfalls mit einem Managementsystem für relationale Datenbanken und einem LDAP-Server arbeiten (da der Typ des Service-Connectors nicht geändert werden kann). Diese Typen unterstützen alle Touchpoint-Rollen.

Virtueller Typ

Diese Kategorie besteht aus einem einzigen Touchpoint-Typ namens `virtual://Intermediary`. Anders als die vorgenannten Typen, die mit einigen realen Ressourcen verbunden sind (dies ist bei Standardtypen ein IBM Security Directory Integrator-Connector und bei angepassten Typen eine Schablonendatei), wird mit diesem Touchpoint-Typ ein Verfahren für die Erstellung einer sofort einsatzfähigen Touchpoint-Instanz mit der Rolle "Intermediary" bereitgestellt. Zu diesem Zweck fußt dieser Typ auf der mit IBM Security Directory Integrator ausgelieferten Touchpoint-Basisschablone. Dieser Touchpoint-Typ unterstützt ausschließlich die Touchpoint-Rolle "Intermediary".

Bei der Erstellung einer Touchpoint-Instanz müssen Sie die Konfiguration des Connectors angeben, der mit dem Drittanbietersystem kommuniziert. Falls ein Touchpoint-Standardtyp verwendet wird, bedeutet dies, dass Sie die Konfiguration des entsprechenden IBM Security Directory Integrator-Connectors bereitstellen müssen. Bei angepassten Typen müssen Sie die Konfiguration für den Service-Connector der angepassten Schablone bereitstellen. Bei virtuellen Typen ist keine Konfiguration erforderlich, da Touchpoint-Instanzen mit der Rolle "Intermediary" zur Ausführung ihrer eigenen Tasks nur HTTP-Komponenten verwenden.

Es stellt sich nun die Frage, welche Parameter für die Erstellung einer Touchpoint-Instanz erforderlich sind. Zu diesem Zweck unterstützt der Touchpoint-Server Eigenschaftenblattdefinitionen, also XML-Dokumente, die Informationen für das Schema einer IBM Security Directory Integrator-Komponente bereitstellen. Um die URL für die Eigenschaftenblattdefinition abzurufen, senden Sie eine HTTP-Anforderung GET an die URL des jeweiligen Touchpoint-Typs. Sie definiert die erforderlichen Connectorparameter, deren Standardwerte sowie weitere nützliche Informationen, die Sie beim Konfigurieren einer Touchpoint-Instanz benötigen.

Im Allgemeinen sind (mit Ausnahme der Parameterbeschreibungen) die Informationen der Eigenschaftenblattdefinitionen den Angaben ähnlich, die beim Konfigurieren eines Connectors im Konfigurationseditor von IBM Security Directory Integrator verfügbar sind. Weitere Details über Eigenschaftenblattdefinitionen und deren Verwendung enthält der Abschnitt „Eigenschaftenblattdefinitionen“ auf Seite 357.

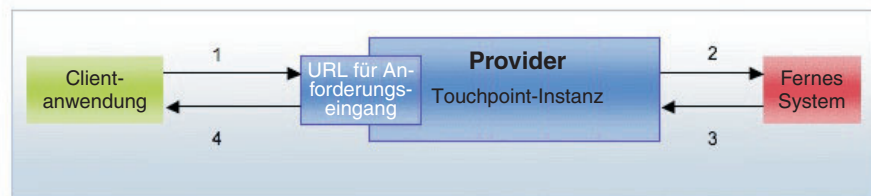
Touchpoint-Instanz

Mit einer Touchpoint-Instanz können Sie den Zugriff über das HTTP-Protokoll verwalten. Nachstehend erhalten Sie Informationen dazu.

Eine Touchpoint-Instanz ist ihrer Spezifik nach ein Proxy, der den Zugriff auf einen fernen Service über das allgemeine HTTP-Protokoll ermöglicht. Der Ablauf der Kommunikation variiert je nach der Rolle der Touchpoint-Instanz jedoch erheblich. Nachstehend sind einige Details für die unterstützten Touchpoint-Rollen aufgeführt:

Provider

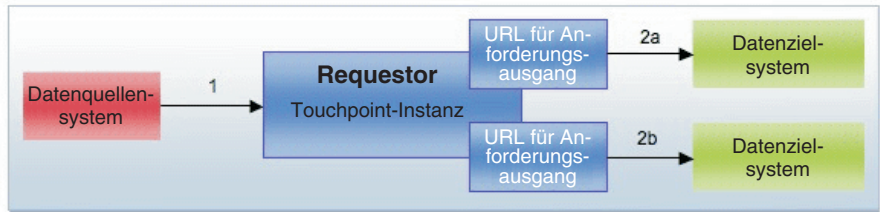
Dieser Modus ermöglicht Clients den Zugriff auf Drittanbieterservices. Ein Client sendet eine HTTP-Anforderung an die Touchpoint-Instanz (1), die ihrerseits die Anforderung in die native Sprache des fernen Systems übersetzt und an dieses System sendet (2). Das ferne System antwortet mit dem richtigen Ergebnis an die Touchpoint-Instanz (3), die das Ergebnis anschließend über HTTP zurück an den Client (4) überträgt.



Aus diesem Diagramm ist ersichtlich, dass der Provider eine **einzige Eingabeschnittstelle** besitzt. Diese ist dargestellt durch eine URL, an die Sie Ihre Anforderungen senden können (URL für Anforderungseingang). Diese URL ist intrinsisch, was bedeutet, dass sie von der Touchpoint-Instanz erstellt und Ihnen zur Verfügung gestellt wird.

Initiator

Dieser Modus ermöglicht den Transport von Informationen zwischen zwei Endpunkten. Die Touchpoint-Instanz fordert mit der systemabhängigen Sprache Daten von einem System an (1) und überträgt anschließend diese Daten über HTTP an mehrere Datenzielsysteme (2a, 2b). Hierbei handelt es sich um HTTP-Server, die Daten empfangen können (z. B. eine Touchpoint-Instanz mit der Rolle "Provider").

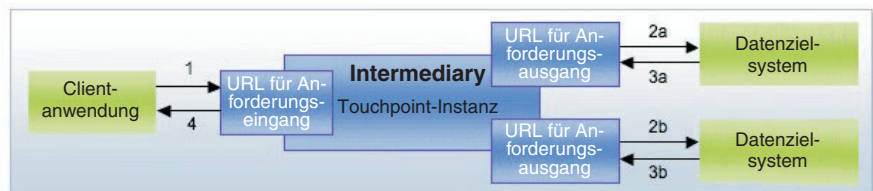


Um dies zu erreichen, kann die Touchpoint-Instanz mit der Rolle "Initiator" **mehrere Ausgabeschnittstellen** besitzen, also URLs für den Anforderungsausgang, an die die verfügbaren Daten gesendet werden. Diese Zielpunkte, die einfach auch "Ziele" genannt werden, können zur Laufzeit zur Touchpoint-Instanz mit der Rolle "Initiator" hinzugefügt oder aus ihr entfernt werden, was die Verteilung von Daten zwischen Systemen außerordentlich flexibel macht. Die Touchpoint-Instanz mit der Rolle "Initiator" beginnt mit der Verarbeitung, wenn ihre erste URL für den Anforderungsausgang hinzugefügt wird, und stoppt, nachdem alle diese URLs entfernt wurden. Die Touchpoint-Instanz mit der Rolle "Initiator" sendet standardmäßig Daten an alle ihre Ziele, ungeachtet deren Antworten. Dieses Verhalten kann modifiziert werden (um beispielsweise die Touchpoint-Instanz mit der Rolle "Initiator" zu stoppen, falls der Datenempfang bei einem der Ziele fehlschlägt). Hierzu können Sie die Basisschablone bearbeiten oder eine angepasste Schablone bereitstellen.

Intermediary

Dieser Modus stellt einen Weiterleitungsservice zur Verfügung. Die Touchpoint-Instanz mit der Rolle "Intermediary" akzeptiert Anforderungen (1) und sendet diese dann an über HTTP an mehrere Ziele (2a, 2b). Die Ziele antworten an die Touchpoint-Instanz (3a, 3b), die deren Antworten zusammenführt und das Ergebnis zurück an die aufrufende Anwendung sendet (4).

Für eine Clientanwendung hat es den Anschein, als ob die Touchpoint-Instanz mit der Rolle "Intermediary" eine Touchpoint-Instanz mit der Rolle "Provider" ist, die den Zugriff auf ein Drittanbietersystem ermöglicht. Für die Datenzielsysteme sieht diese Instanz wie eine Touchpoint-Instanz mit der Rolle "Initiator" aus, die Daten sendet.



Die Touchpoint-Instanz mit der Rolle "Intermediary" besitzt eine **einzige Eingabeschnittstelle** (URL für den Anforderungseingang) und **mehrere Ausgabeschnittstellen** (URLs für den Anforderungsausgang). Sie können die Ausgabeschnittstellen genauso wie bei der Rolle "Initiator" konfigurieren. Die Eingabeschnittstelle hat Ähnlichkeit mit der Eingabeschnittstelle der Rolle "Provider" (ihre URL wird also durch den Touchpoint-Server festgelegt). In ihrer einfachsten Form modifiziert die Touchpoint-Instanz mit der Rolle "Intermediary" die weitergeleiteten Daten nicht. Sie können jedoch entsprechende Logik angeben, indem Sie die Touchpoint-Basisschablone bearbeiten oder eine angepasste Schablone bereitstellen.

Ein weiteres spezielles Merkmal der Touchpoint-Instanz mit der Rolle "Intermediary" ist zu beobachten, wenn sie als Proxy für den Zugriff auf mehrere Touchpoint-Instanzen mit der Rolle "Provider" fungiert. Wie bereits erläutert, können Sie mit diesem komplexen System so interagieren, als ob es sich um eine einfache Touchpoint-Instanz mit der Rolle "Provider" handelte. Es ist jedoch unbedingt erforderlich, dass die von den End Providern zurückgegebenen Antworten zusammengeführt werden. Hierzu ist die folgende Standardlogik verfügbar:

- Falls eines der Ziele eine Antwort über eine erfolgreiche Ausführung zurückgibt, wird diese an die aufrufende Anwendung zurückgegeben.
- Falls mehrere Ziele eine Antwort über eine erfolgreiche Ausführung zurückgeben, werden die Daten aus den Hauptteilen der Antworten zusammengeführt und mit einem HTTP-Code 200 zurückgegeben.
- Falls alle Ziele Antworten über eine fehlgeschlagene Ausführung zurückgeben, wird an die aufrufende Anwendung eine Fehlerantwort mit dem Code 500 zurückgegeben. In ihrem HTTP-Hauptteil enthält diese Antwort für jedes Ziel die folgenden Informationen:
 - Die Angabe, dass die Anforderung an die URL des Ziels fehlgeschlagen ist.
 - Als HTTP-Status den zurückgegebenen HTTP-Fehlercode.
 - Als HTTP-Hauptteil den zurückgegebenen HTTP-Hauptteil.

Um das Verhalten beim Zusammenführen zu ändern, müssen Sie die verwendete „Touchpoint-Schablone“ auf Seite 342 bearbeiten.

Weitere Details über das zwischen dem Client und der Touchpoint-Instanz verwendete Kommunikationsprotokoll enthält der Abschnitt „Kommunikationsprotokoll der Touchpoint-Instanz“ auf Seite 353.

Neben den vorgenannten Rollen müssen Sie beim Einrichten einer Touchpoint-Instanz zwei weitere Konfigurationselemente angeben. Es handelt sich um die folgenden Elemente:

- **Eigenschaftenblatt** der Touchpoint-Instanz: Jede Touchpoint-Instanz stellt einen bestimmten Touchpoint-Typ dar, was bedeutet, dass jede Touchpoint-Instanz eine bestimmte Konfiguration besitzt. Sie richtet sich nach der durch den Touchpoint-Typ definierten Konfiguration (Eigenschaftenblattdefinition) und gibt das Schema des IBM Security Directory Integrator-Connectors wieder, der für diese Touchpoint-Instanz zur Arbeit mit dem fernen System verwendet wird (dieser Connector wird als "Service-Connector" bezeichnet).

Anmerkung: Eine Touchpoint-Instanz mit der Rolle "Intermediary" verwendet für ihre Arbeit keinen Service-Connector, sodass für sie keine derartigen Konfigurationsinformationen benötigt werden. Beim Einrichten solcher Touchpoint-Instanzen sollten Benutzer ein leeres Eigenschaftenblatt senden, da alle bereitgestellten Parameter einfach ignoriert werden.

Weitere Informationen zum Format der Eigenschaftenblätter finden Sie im Abschnitt „Instanzkonfiguration“ auf Seite 352.

- **Verwaltungsstatus** der Touchpoint-Instanz: Dieses Element wird zur Mikroverwaltung für den Status der Instanz verwendet. Es kann beispielsweise vorkommen, dass Sie eine aktive Touchpoint-Instanz mit der Rolle "Provider" für einen gewissen Zeitraum inaktivieren wollen. Statt die Touchpoint-Instanz mit der Rolle "Provider" zu löschen, können Sie ihren Verwaltungsstatus einfach auf "disab-

led" setzen. Wenn die Instanz später wieder benötigt wird, müssen Sie dadurch lediglich ihren Status wieder auf "enabled" aktualisieren, damit sie erneut aktiv ist.

Ein weiterer möglicher Einsatzbereich ergibt sich bei den Rollen "Intermediary" und "Initiator". Sobald Sie zu Touchpoint-Instanzen mit diesen Rollen das erste Ziel hinzufügen, beginnen die Instanzen mit der Arbeit (und wahrscheinlich mit dem Senden von Daten). Zusätzliche Ziele können zwar später hinzugefügt werden, aber dies kann zu einem Datenverlust führen, da einige der Daten bereits gesendet wurden. Um dieses Problem zu lösen, können Sie Touchpoint-Instanzen mit einem Verwaltungsstatus "disabled" erstellen (was ihren Start verhindert) und so viele Ziele wie benötigt hinzufügen. Nach Fertigstellung der Konfiguration können Sie den Status in "enabled" ändern. Die Touchpoint-Instanz beginnt dann mit dem Senden von Daten an alle Ziele.

Nachdem Sie die Touchpoint-Instanz erstellt haben, können Sie auf ihren aktuellen Betriebsstatus zugreifen. Für eine Touchpoint-Instanz gibt es zwei mögliche Status, deren Bedeutung je nach Rolle der Instanz variiert.

Eine Touchpoint-Instanz mit der Rolle *Provider* kann die folgenden Status aufweisen:

- **Unavailable:** Dieser Status besteht, wenn die Touchpoint-Instanz vorsätzlich über den Verwaltungsstatus inaktiviert wurde.
- **Available:** Dieser Status besteht, wenn die Touchpoint-Instanz konfiguriert ist und ihr Verwaltungsstatus "enabled" lautet.

Für den Status einer Touchpoint-Instanz mit der Rolle "Provider" gibt es einen zusätzlichen Parameter. Neben dem Betriebsstatus können Sie die URL für den Anforderungseingang der Touchpoint-Instanz abrufen, also die URL-Adresse, an die Sie Ihre Anforderungen senden, damit die Touchpoint-Instanz mit der Rolle "Provider" diese an das ferne System übertragen kann.

Eine Touchpoint-Instanz mit der Rolle *Initiator* kann die folgenden Status aufweisen:

- **Unavailable:** Die Touchpoint-Instanz weist diesen Status auf, wenn Folgendes zutrifft:
 - Sie ist nicht vollständig konfiguriert, besitzt also keine Ziele (URLs für den Anforderungsausgang).
 - Sie wurde durch das Festlegen des entsprechenden Verwaltungsstatus vorsätzlich inaktiviert.
 - Ihr Service-Connector hat das Lesen der Daten aus der Datenquelle beendet. Dieser spezielle Fall beruht auf dem Verhalten des Service-Connectors für die Touchpoint-Instanz mit der Rolle "Initiator". Falls es sich bei diesem Connector um einen Standarditerator handelt, liest er seine konfigurierte Datenquelle und wird anschließend gestoppt, was einen Stopp der gesamten Touchpoint-Instanz bewirkt. Änderungserkennungs- oder JMS-Connectors warten jedoch weiterhin auf neue Daten und die Touchpoint-Instanz bleibt für unbegrenzte Dauer aktiv (und verfügbar). In diesem Fall muss die Instanz explizit dadurch gestoppt werden, dass entweder die Instanz gelöscht oder ihr Verwaltungsstatus auf "disabled" gesetzt wird.
- **Available:** Wenn die Touchpoint-Instanz diesen Status aufweist, liest sie gegenwärtig Daten aus der Datenquelle.

Wie zuvor erläutert, stellt eine Touchpoint-Instanz mit der Rolle *Intermediary* im Grunde genommen eine Kombination aus den Touchpoint-Rollen "Provider" und "Initiator" dar. Dies spiegelt sich auch in ihrem Status wieder.

- **Unavailable:** Die Touchpoint-Instanz weist diesen Status auf, wenn Folgendes zutrifft:
 - Sie ist nicht vollständig konfiguriert, besitzt also keine Ziele (URLs für den Anforderungsausgang).
 - Sie wurde durch das Festlegen des entsprechenden Verwaltungsstatus vorsätzlich inaktiviert.
- **Verfügbar:** Wenn die Touchpoint-Instanz diesen Status aufweist, lautet ihr Verwaltungsstatus "enabled" und die Instanz leitet ihre eingehenden Anforderungen aktiv an die Ziele weiter.

Ähnlich wie eine Touchpoint-Instanz mit der Rolle "Provider" besitzt eine Instanz mit der Rolle "Intermediary" eine URL für den Anforderungseingang, die aus ihrem Statuseintrag abgerufen werden kann.

Touchpoint-Schablone

Mit der Touchpoint-Schablone können Sie die Konfiguration von IBM Security Directory Integrator starten.

Wenn eine Touchpoint-Instanz gestartet wird, erstellt der Touchpoint-Server eine IBM Security Directory Integrator-Konfiguration und startet sie auf seinem zugeordneten Touchpoint-Provider (IBM Security Directory Integrator-Server) als temporäre IBM Security Directory Integrator-Konfigurationsinstanz. Die IBM Security Directory Integrator-Konfiguration basiert auf einer mit dem Touchpoint-Server bereitgestellten Basisschablone. Der Pfad zu dieser Schablonendatei ist in der „Konfiguration“ auf Seite 360 des Touchpoint-Servers angegeben; der Standardwert lautet *tdi-installationsverzeichnis/etc/TouchpointTemplate.xml*. Die Schablone befindet sich auf der Seite des Touchpoint-Servers und ist eine allgemeine Schablone, der kein bestimmter IBM Security Directory Integrator-Server zugeordnet ist. Die Konfiguration jeder Touchpoint-Instanz wird in die Basisschablone eingefügt, bevor sie als Konfigurationsinstanz gestartet wird.

Die Standardbasisschablone enthält die folgende Struktur:

Fertigungslinien:

- **ProviderServer:** Diese Fertigungslinie ist für das Empfangen von HTTP-Anforderungen und das Starten der geeigneten Handlerfertigungslinien für diese Anforderungen zuständig. Sie fungiert als allgemeiner Einstiegspunkt für den Zugriff auf alle Touchpoint-Instanzen mit der Rolle "Provider" und "Intermediary", die durch den Touchpoint-Server verwaltet werden.

Bei diesem Verfahren wird für die Kommunikation mit allen diesen Touchpoint-Instanzen nur ein einziger HTTP-Server-Connector benötigt, was bedeutet, dass nur ein einziger TCP-Port geöffnet sein muss (standardmäßig Port 1097). Dies verhindert mögliche Firewallprobleme, die auftreten können, falls viele wahlfreie Ports geöffnet sein müssen.

Die Fertigungslinie "ProviderServer" verwendet die folgenden IBM Security Directory Integrator-Komponenten:

- *HttpServer:* Der Connector, der die Anforderungen von Clientanwendungen empfängt. Standardmäßig ist er an Port 1097 empfangsbereit. Dies kann jedoch in der „Konfiguration“ auf Seite 360 des Touchpoint-Servers geändert werden.

- *HandleRequest*: Eine IBM Security Directory Integrator-Scriptkomponente, die abhängig von der Anforderung bestimmt, welche Fertigungslinie gestartet werden soll.
- **ProviderHandler**: Diese Fertigungslinie wird durch die Fertigungslinie "ProviderServer" gestartet, wenn eine Anforderung für die Touchpoint-Instanz mit der Rolle "Provider" empfangen wird. Sie führt die eigentliche Kommunikation mit dem fernen System aus. Die folgenden Komponenten werden verwendet:
 - *ServiceConnector*: Dies ist der Connector, der mit dem fernen System kommuniziert. Sein Status ist auf "passive" gesetzt, was bedeutet, dass er nicht über den Datenfluss der Fertigungslinie, sondern über ein Script gesteuert wird. Ein wichtiges Merkmal dieses Connectors ist die Tatsache, dass er aus "GenericServiceConnector" in der Bibliothek der Basisschablone erbt (Vererbung aus /Connectors/GenericServiceConnector). Die Rolle dieses übergeordneten Connectors ist nachfolgend beschrieben.

Anmerkung: Eine Touchpoint-Fertigungslinie sollte nur einen einzigen Service-Connector besitzen (bzw. bei der Rolle "Intermediary" gar keinen). Falls mehrere Connectors angegeben sind, haben alle dieselbe Konfiguration.

- *HandleRequest*: Dieses Script verwendet eine servletähnliche Struktur, um die eingehenden Anforderungen zu verarbeiten. Es initialisiert die Komponente "ServiceConnector" und führt je nach empfangener Anforderung unterschiedliche Operationen für die Datenquelle aus. Weitere Details über die unterstützten Provideroperationen enthält der Abschnitt „Kommunikationsprotokoll der Touchpoint-Instanz“ auf Seite 353.
- **IntermediaryHandler**: Wenn die Fertigungslinie "ProviderServer" eine Anforderung für eine Touchpoint-Instanz mit der Rolle "Intermediary" empfängt, leitet sie die Anforderung an diese Fertigungslinie weiter. Da bei der Rolle "Intermediary" Daten standardmäßig lediglich an mehrere Ziele weitergeleitet werden, benötigt sie nur eine einzige Komponente:
 - *SendToDestinations*: Diese Scriptkomponente leitet jede (an die Fertigungslinie in Form eines Work-Eintrags übergebene) Anforderung mit dem folgenden Aufruf an die konfigurierten Touchpoint-Ziele weiter:


```
sendToDestinations(work, mergeResponsesCallback)
```

Diese Methode wird durch die Senderscriptkomponente bereitgestellt, die sich in der Bibliothek der Basisschablone befindet. Neben dem gesendeten Eintrag ist eine Rückruffunktion erforderlich, mit deren Hilfe die Antworten der einzelnen Ziele zusammengeführt werden. Details über das Standardverhalten der Touchpoint-Instanz mit der Rolle "Intermediary" beim Zusammenführen finden Sie unter „Touchpoint-Instanz“ auf Seite 338.

- **Initiator**: Diese Fertigungslinie wird zur Darstellung einer Touchpoint-Instanz mit der Rolle "Initiator" verwendet. Dies ist die einzige Touchpoint-Rolle, auf die nicht über den Einstiegspunkt "ProviderServer" zugegriffen wird. Diese Fertigungslinie besteht aus den folgenden Komponenten:
 - *ServiceConnector*: Dieser Connector wird verwendet, um der Fertigungslinie Daten aus dem fernen System zuzuführen. Wie bei der Fertigungslinie "ProviderHandler" erbt dieser Connector aus "GenericServiceConnector" in der Bibliothek der Basisschablone.

Anmerkung: Eine Touchpoint-Fertigungslinie sollte nur einen einzigen Service-Connector besitzen (bzw. bei der Rolle "Intermediary" gar keinen). Falls mehrere Connectors angegeben sind, haben alle dieselbe Konfiguration.

- *ConvertToHTTPContent*: Diese Scriptkomponente wandelt die aus dem Service-Connector gelesenen Daten in einen HTTP-Eintrag um.
- *SendToDestinations*: Diese Scriptkomponente leitet die empfangene Anforderung mit dem folgenden Aufruf an die Touchpoint-Ziele weiter:
`sendToDestinations(work, null)`

Derselbe Aufruf wird auch von der Touchpoint-Instanz mit der Rolle "Intermediary" verwendet. Der einzige Unterschied besteht darin, dass hier keine Rückruffunktion für das Zusammenführen der Antworten von den unterschiedlichen Zielen bereitgestellt wird. Details über das Standardverhalten der Touchpoint-Instanz mit der Rolle "Initiator" beim Zusammenführen finden Sie unter „Touchpoint-Instanz“ auf Seite 338.

Ressourcen (Bibliothek der Basisschablone):

- **Connectors** – Mehrere Connectors führen bestimmte Tasks für die verschiedenen Fertigungslinien aus:
 - *GenericServiceConnector*: Dieser Connector ist das übergeordnete Element aller Service-Connectors. Wenn eine Touchpoint-Instanz erstellt wird, konfiguriert der Touchpoint-Server diesen Connector für die Arbeit mit dem jeweiligen fernen System. Da der Service-Connector in jeder Fertigungslinie von diesem Connector erbt, haben alle Service-Connectors dieselbe Konfiguration.

Anmerkung:

1. Der Name des Service-Connectors in der Fertigungslinie ist nicht von Bedeutung, da er aus "GenericServiceConnector" erbt.
2. Es sollte nur ein einziger Service-Connector pro Fertigungslinie bereitgestellt werden.

Der Touchpoint-Server behandelt "GenericServiceConnector" je nach Touchpoint-Typ unterschiedlich:

- Beim **Touchpoint-Standardtyp** wird "GenericServiceConnector" verwendet, um sowohl die Vererbung als auch die Parameter der Service-Connectors festzulegen. Beispiel: Sie erstellen eine Touchpoint-Instanz mit der Rolle "Provider", die mit einem Managementsystem für relationale Datenbanken arbeitet. Diese Instanz erstellen Sie aus dem Typ `system:/Connectors/ibmdi.JDBC`, konfigurieren sie im Modus "Provider" und übergeben die durch den JDBC-Connector benötigten Parameter. Dies bedeutet, dass die Vererbung aus "GenericServiceConnector" mit `system://Connectors/ibmdi.JDBC` überschrieben wird und dass die bereitgestellten JDBC-Parameter festgelegt werden. Auf diese Weise erhalten beide Service-Connectors in den Fertigungslinien "ProviderHandler" und "Initiator" ebenfalls dieselbe Konfiguration (da sie aus "GenericServiceConnector" erben). Die Fertigungslinie "ProviderHandler" wird gestartet und Sie erhalten eine Touchpoint-Instanz mit der Rolle "Provider" für die Arbeit mit einem Managementsystem für relationale Datenbanken.
- Beim **angepassten Touchpoint-Typ** wird "GenericServiceConnector" verwendet, um den Typ der Service-Connectors zu ermitteln und

deren Parameter festzulegen. Zu diesem Zeitpunkt wird die Vererbung aus "GenericServiceConnector" nicht geändert. Sie wird stattdessen vom Touchpoint-Server verwendet, um den Typ des ServiceConnectors zu ermitteln, damit sein Schema bekannt ist. Wenn anschließend die Touchpoint-Instanz konfiguriert wird, werden die Parameter wieder wie bei "GenericServerConnector" festgelegt und somit an die untergeordneten Elemente weitergegeben.

- Beim **virtuellen Touchpoint-Typ** wird "GenericServiceConnector" (zumindest gegenwärtig) nicht verwendet, da zu diesem Schema bislang nur der Touchpoint-Typ "Intermediary" gehört und keine Verbindung zu Drittanbietersystemen hergestellt wird.
 - *HTTPClientConnector*: Der HTTP-Client, der von Touchpoint-Instanzen mit der Rolle "Initiator" und "Intermediary" zum Senden von Daten an Ziele verwendet wird. Er wird durch das Senderscript eingesetzt und für die Touchpoint-Instanzen über die Methode `sendToDestinations()` zur Verfügung gestellt.
 - *MemoryPropertiesConnector*: Ein Script-Connector, der vom Speicher "MemoryProperties" verwendet wird, um die URLs für den Anforderungsausgang der Ziele aufzunehmen. Er bietet die Möglichkeit, mit einer aktiven Touchpoint-Instanz zu kommunizieren und Ziele zu dieser Instanz hinzuzufügen bzw. aus ihr zu entfernen (weitere Details finden Sie in der Beschreibung des Speichers "MemoryProperties"). Dieser Connector ahmt das Verhalten des Eigenschaftsconnectors nach, allerdings mit dem wesentlichen Unterschied, dass er die bereitgestellten Daten nicht in einer Datei speichert.
- **Properties:**
 - *MemoryProperties*: Ein Eigenschaftsspeicher, der für die Kommunikation mit einer aktiven Touchpoint-Fertigungslinie verwendet wird. Zum Speichern der übergebenen Daten im Hauptspeicher stützt er sich auf den Connector "MemoryProperties". Für jede Touchpoint-Instanz wird eine separate Konfigurationsinstanz gestartet, die jeweils einen eigenen Speicher "MemoryProperties" besitzt. Hierdurch besteht keine Gefahr, dass Kommunikationsnachrichten miteinander verwoben werden.

Die Kommunikationsprozedur vollzieht sich folgendermaßen:

1. Der Touchpoint-Server verwendet die ferne Server-API, um eine bestimmte Eigenschaft – `com.ibm.di.tp.destinations` – im Speicher "MemoryProperties" abzulegen. Ihr Wert ist ein Element `java.util.List` von `java.util.Map`, das die bereitgestellten Zielparameter enthält (beispielsweise URLs für den Anforderungsausgang und Anforderungsfehler), die für die Touchpoint-Instanz konfiguriert sind.
2. Immer dann, wenn ein Ziel zu einer Touchpoint-Instanz mit der Rolle "Initiator" oder "Intermediary" hinzugefügt bzw. aus ihr entfernt wird, wird der Wert dieser Eigenschaft aktualisiert und gibt die aktuelle Liste der URLs an.
3. Bei jeder Iteration ruft die Fertigungslinie der Touchpoint-Instanz mit der Rolle "Initiator" oder "Intermediary" den aktuellen Wert dieser Eigenschaft ab und sendet ihre Daten an die gespeicherten URLs (dies wird durch die Routine `sendToDestinations()` ausgeführt).

Die Touchpoint-Basisschablone stellt einen korrekt konfigurierten Speicher "MemoryProperties" bereit. Bei jeder Erstellung einer Touch-

point-Instanz überprüft der Touchpoint-Server, ob der Speicher "MemoryProperties" fehlt. Wenn dies der Fall ist, fügt er ihn zur Konfiguration hinzu. Wenn Sie diesen Speicher in ihren angepassten Schablonen nicht konfigurieren, ist daher immer noch der Standard-Speicher verfügbar. Modifizieren Sie hingegen den Speicher "MemoryProperties", um ein geändertes Kommunikationsverhalten zu erzielen (beispielsweise Verwendung eines Eigenschaftsconnectors, damit die Ziel-URLs in einer Datei gespeichert werden), überschreibt der Touchpoint-Server dessen neue Konfiguration nicht.

- **Scripts:**

- *Sender*: Ein Script, das die Methode `sendToDestinations()` bereitstellt. Diese Routine liest den Inhalt der Eigenschaft `com.ibm.di.tp.destinations` im Speicher "MemoryProperties", um die URLs für den Anforderungsausgang und für Anforderungsfehler des Ziels abzurufen, und sendet den bereitgestellten Work-Eintrag mit dem HTTP-Client-Connector.

Anmerkung: Die URL für Anforderungsfehler wird nicht standardmäßig verwendet, sondern nur automatisch für die Touchpoint-Schablone bereitgestellt, damit Sie den Standardmechanismus für die Fehlerbehebung erweitern oder einen angepassten Mechanismus implementieren können.

- *Utils*: Dieses Script stellt eine Reihe von Dienstprogrammfunktionen bereit, die durch die Touchpoint-Fertigungslinien verwendet werden.

Mithilfe von **angepassten Schablonen** können Sie in einer Fertigungslinie komplexes/angepasstes Verhalten bereitstellen und als neuen Touchpoint-Typ verfügbar machen. Die Standardbasisschablone wird für die direkte Bereitstellung von IBM Security Directory Integrator-Komponenten und einer Touchpoint-Instanz mit der Rolle "Intermediary" verwendet. Sie müssen in diesem Fall keine Kenntnisse darüber besitzen, welche Aktionen innerhalb von IBM Security Directory Integrator stattfinden. Es kann jedoch sein, dass Sie nicht nur eine einzige Komponente, sondern weitere Logik und/oder Connectors zu einer Touchpoint-Instanz hinzufügen wollen. Zu diesem Zweck akzeptiert der Touchpoint-Server angepasste Schablonen, damit neue angepasste Touchpoint-Typen und neues Verhalten unterstützt werden.

Die Struktur einer angepassten Schablone *sollte* der Struktur der Basisschablone entsprechen. Falls Ihr angepasster Touchpoint-Typ jedoch nicht alle Touchpoint-Rollen unterstützen muss, haben Sie die Möglichkeit, nur eine Untergruppe der Fertigungslinien anzugeben. Für jede Rolle bestehen die folgenden Mindestvoraussetzungen:

- Rolle **Provider**: Zur Unterstützung dieser Rolle sollte die angepasste Schablone die Fertigungslinie *ProviderHandler* und die folgenden Bibliothekskomponenten enthalten: die Scriptkomponente *Utils* sowie *GenericServiceConnector*.

Anmerkung: Die Fertigungslinie "ProviderServer" der Basisschablone wird verwendet, um Anforderungen an alle Touchpoint-Instanzen zu delegieren. Selbst dann, wenn eine angepasste Schablone eine Fertigungslinie "ProviderServer" enthält, wird diese daher nicht durch den Touchpoint-Server verwendet. Stattdessen wird die Fertigungslinie der Basisschablone verwendet.

- Rolle **Initiator**: Diese Rolle erfordert die Fertigungslinie "Initiator" und die folgenden Bibliothekskomponenten: die Scriptkomponenten *Sender* und *Utils* sowie *MemoryPropertiesConnector*, *GenericServiceConnector* und *HttpClientConnector*.

Anmerkung: Der Speicher "MemoryProperties" ist nicht aufgeführt, denn falls er fehlt, wird er durch den Touchpoint-Server hinzugefügt.

- Rolle **Intermediary**: Diese Rolle erfordert die Fertigungslinie *IntermediaryHandler* und die folgenden Bibliothekskomponenten: die Scriptkomponenten *Sender* und *Utils* sowie *MemoryPropertiesConnector* und *HttpClientConnector*.

Falls Sie versuchen, eine Touchpoint-Instanz mit einer Rolle zu erstellen, deren Voraussetzungen durch die Schablone des angepassten Typs nicht erfüllt werden, wird eine Ausnahmebedingung ausgelöst.

Beim Bearbeiten der Touchpoint-Basisschablone oder beim Erstellen einer angepassten Schablone müssen Sie die folgenden wichtigen Punkte berücksichtigen:

1. Die Service-Connectors (die mit einem Drittanbietersystem kommunizieren) sollten aus "GenericServiceConnector" in der Bibliothek erben.
2. In einer Touchpoint-Fertigungslinie sollte nur ein einziger Server-Connector verwendet werden. Falls mehrere Connectors bereitgestellt werden, erhalten alle dieselbe Konfiguration und sind somit ohne Nutzen.
3. Falls Sie den für die Ziel-URLs der Kommunikation verwendeten Speicher ändern wollen, muss dieser den Namen "MemoryProperties" erhalten.

Ressourcenpersistenz

Nachstehend erhalten Sie Informationen zur Persistenz und ihrer Verwendung. Beachten Sie außerdem die Bedingungen beim Neustart der Touchpoint-Instanz.

Der Touchpoint-Server unterstützt die Persistenz der Atom-Dokumente, die entweder automatisch durch ihn generiert oder von einem fernen Client bereitgestellt wurden. Die durch den Server verwendete Persistenz ist in einem konfigurierten Ordner mit einer Baumstruktur gespeichert. Das Standardpersistenzverzeichnis heißt `lösungsverzeichnis/tp_state`.

Die persistenten Ressourcen werden beim Start des Touchpoint-Servers erneut gelesen. Zu diesem Zeitpunkt stellt der Touchpoint-Server die gesamte Ressourcenbaumstruktur wieder her. Der Touchpoint-Server speichert die Konfigurationen der Touchpoint-Instanzen, damit eine Touchpoint-Instanz auch bei einem Neustart des Servers erhalten bleibt. Eine Touchpoint-Instanz wird erneut gestartet, wenn alle nachfolgenden Bedingungen erfüllt sind:

- Alle erforderlichen Konfigurationen für diese Touchpoint-Instanz sind im Persistenzspeicher verfügbar.
- Der ferne Touchpoint-Provider ist betriebsbereit.
- Auf dem Touchpoint-Provider wird keine andere Touchpoint-Instanz mit dieser Konfiguration ausgeführt.

Wenn der ferne Touchpoint-Provider nicht aktiv ist, wird nach einer erneuten Aktivierung des Touchpoint-Providers die Touchpoint-Instanz nicht automatisch gestartet. Vor dem Starten des Touchpoint-Servers müssen Sie entweder sicherstellen, dass der Touchpoint-Provider aktiv ist, oder eine Anforderung GET an die Zuführungsressource für Touchpoint-Typen senden, um die Aktualisierung der Verbindung zum fernen Touchpoint-Provider durch den Touchpoint-Server zu erzwingen.

Eine Bearbeitung der Dateien im Persistenzverzeichnis ist nicht empfehlenswert. Gegenwärtig sollten diese Dateien ausschließlich durch den Touchpoint-Server bearbeitet werden.

Touchpoint-Schema

Ein Touchpoint-Schema besteht aus vielen Komponenten. Nachstehend erhalten Sie Informationen zu dem Schema sowie zur Ressourcenbaumstruktur und zu den zulässigen Operationen.

In diesem Abschnitt ist das Kommunikationsprotokoll zwischen dem Touchpoint-Server und einem Client dargestellt, der ihn zur Bereitstellung von Touchpoint-Instanzen verwendet. Die Kommunikation zwischen einem Client und der Touchpoint-Instanz selbst ist in diesem Abschnitt nicht beschrieben.

Der Touchpoint-Server ermöglicht den Zugriff auf die verschiedenen Ressourcen, die an der Definition einer Touchpoint-Instanz beteiligt sind. Die Darstellung dieser Ressourcen sieht wie ein Strukturbaum aus. Der Zugriff auf jede Ressource erfolgt unter Verwendung von Atom-Dokumenten über das HTTP/HTTPS-Protokoll.

Das folgende Schema wird vom Touchpoint-Server verwendet:

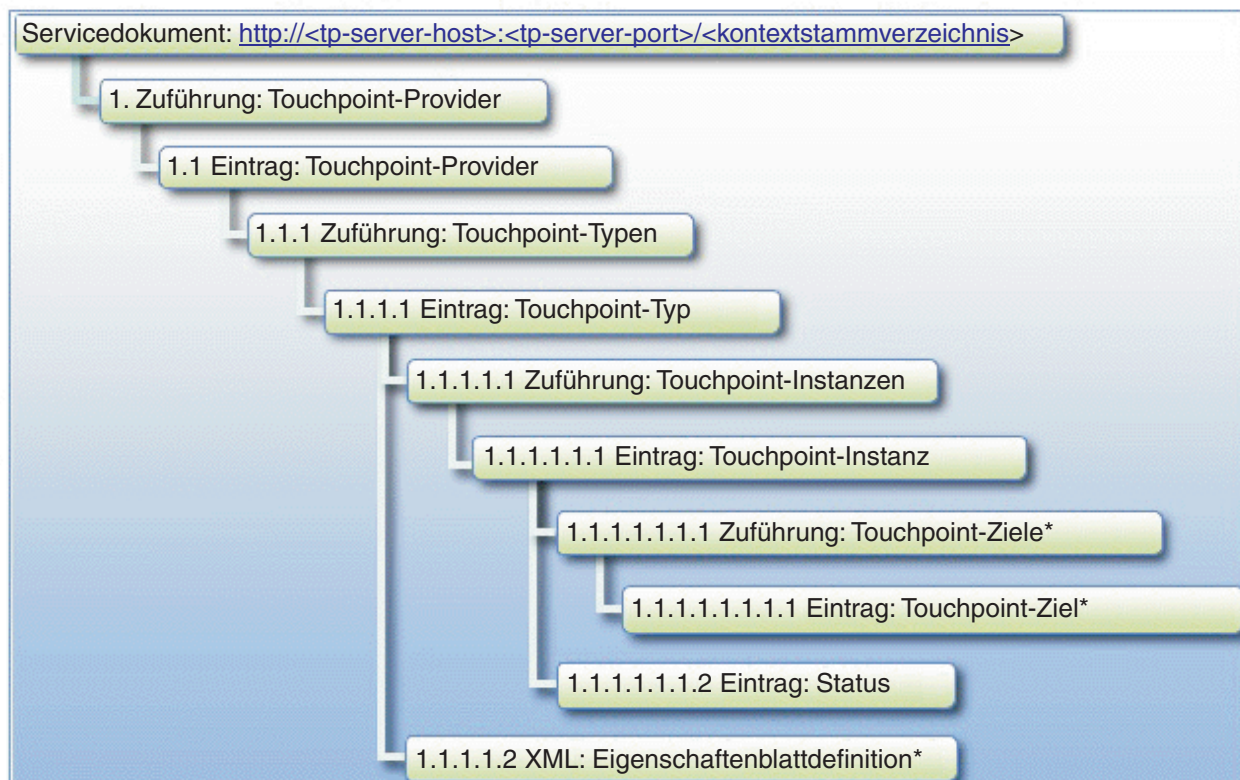


Abbildung 1. Baumstruktur des Touchpoint-Server-Schemas

Mit einem Stern (*) gekennzeichnete Baumknoten sind in bestimmten Fällen verfügbar. Details enthält die nachfolgende Tabelle.

Im obigen Schema werden die folgenden Variablen verwendet:

- **tp-server-host:** Dies ist die Hostadresse, an der der Touchpoint-Server empfangsbereit ist.
- **tp-server-port:** Dies ist der Port, an dem der Touchpoint-Server empfangsbereit ist (Standardport ist 1098).

- **kontextstammverzeichnis:** Dies ist das Kontextstammverzeichnis, unter dem die Anwendung des Touchpoint-Servers verfügbar ist (Standardwert ist "tp").

Die Navigation in der Baumstruktur erfolgt mit einem ReSTful-Verfahren. Dies bedeutet, dass Clientanwendung lediglich den Einstiegspunkt (also die URL des Servicedokuments) und den Typ der Referenzen (Atom-Links) kennen müssen, die der Touchpoint-Server für den Zugriff auf die einzelnen Knoten in der Ressourcenbaumstruktur verwendet. Diese Referenzen (URLs) werden durch den Touchpoint-Server automatisch generiert. Sobald die URLs abgerufen wurden, bleiben sie so lange gleich, bis der Touchpoint-Server auf eine neuere Version aktualisiert wird. Dies bedeutet auch, dass sich Clientanwendungen die abgerufenen URLs über die Aktualisierungen des Touchpoint-Servers hinweg zwar "merken" können, jedoch eine bestimmte Ressourcen-URL erneut ermitteln sollten, falls der Touchpoint-Server aktualisiert wird.

Je nach dem in der folgenden Tabelle angegebenen Protokoll sollten die folgenden Schritte ausgeführt werden, damit eine Clientanwendung ausgehend vom Servicedokument zur Zuführung für die Touchpoint-Instanz navigieren kann.

1. Senden Sie eine HTTP-Anforderung GET an die URL des Servicedokuments. Hierdurch wird das Servicedokument zurückgegeben, aus dem die Zuführungs-URL der Touchpoint-Provider abgerufen werden kann.
2. Senden Sie eine HTTP-Anforderung GET an die Zuführungs-URL der Touchpoint-Provider. Hierdurch wird das Zuführungsdokument zurückgegeben, aus dem die Referenz-URL für den Eintrag des Touchpoint-Providers abgerufen werden kann.
3. Senden Sie eine HTTP-Anforderung GET an die Referenz-URL für den Eintrag des Touchpoint-Providers. Hierdurch wird das Eintragsdokument zurückgegeben, das den jeweiligen Touchpoint-Provider darstellt. Dieser Eintrag enthält die Zuführungs-URL des Touchpoint-Typs.
4. Senden Sie eine HTTP-Anforderung GET an die Zuführungs-URL des Touchpoint-Typs. Hierdurch wird das Zuführungsdokument zurückgegeben, das vollständige Kopien aller Einträge für Touchpoint-Typen enthält, die für diesen Kontext gültig sind. Aus einem Eintrag für einen Touchpoint-Typ kann die Clientanwendung die Zuführungs-URL der Touchpoint-Instanz abrufen.

In der folgenden Tabelle sind die zulässigen Operationen für die einzelnen Ressourcen beschrieben. Außerdem gelten für die gesamte Ressourcenbaumstruktur die folgenden Punkte:

- Jeder Eintrag besitzt einen Link mit einem Selbstbezug ("self"), der auf das eigenständige Eintragsdokument zeigt.
- Ressourcen, die die HTTP-Methoden PUT und DELETE akzeptieren, besitzen einen Link mit einem Bearbeitungsbezug ("edit"). Alle derartigen Anforderungen sollten an diesen Link (URL) gesendet werden.
- Jede Anforderung an den Touchpoint-Server wird mit dem HTTP-ETag "response-header" wie in der HTTP-Spezifikation definiert annotiert. Der ETag-Wert kann zusammen mit den Angaben "If-Match", "If-None-Match" und "If-Range" von "request-header" verwendet werden, um dem Touchpoint-Server die Vorbedingungen der Clientanwendung mitzuteilen, bevor die Anforderung bedient wird.

Tabelle 32. Zulässige Operationen nach Ressourcen

Ressource	GET	POST	PUT	DELETE
Servicedokument	Ruft das Servicedokument ab, das eine Liste der verfügbaren Services enthält. Die Zuführungs-URL der Touchpoint-Provider wird als Attribut "href" für eine Objektgruppe festgelegt, die zur Kategorie "connectivity-provider" gehört.	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend
1. Zuführung: Touchpoint-Provider Kategorien (Begriff: Schema): connectivity-provider: http://www.ibm.com/xmlns/prod/scmp#resource	Ruft die Liste der Einträge für die Touchpoint-Provider ab. Alle Einträge sind Referenzen zu den eigentlichen Eintragsdokumenten, die die verfügbaren Touchpoint-Provider darstellen.	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend
1.1 Eintrag: Touchpoint-Provider	Ruft einen Eintrag für einen Touchpoint-Provider gemäß der Konfiguration in der Konfigurationsdatei des Touchpoint-Servers ab. Dieser Eintrag stellt im Element <{http://www.ibm.com/xmlns/prod/scmp}:data/> einige zusätzliche Details bereit. Der Link zur Zuführung für die Touchpoint-Typen wird unter Verwendung eines Links mit dem Bezug "http://www.ibm.com/xmlns/prod/scmp#touchpoint" bereitgestellt.	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend
1.1.1 Zuführung: Touchpoint-Typen Kategorien (Begriff: Schema): touchpoint: http://www.ibm.com/xmlns/prod/scmp#resource	Ruft die Liste der Einträge für Touchpoint-Typen ab. Diese Einträge sind vollständige Kopien der eigentlichen Eintragsdokumente, die die Touchpoint-Typen darstellen.	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend
1.1.1.1 Eintrag: Touchpoint-Typ Kategorien (Begriff: Schema): touchpoint: http://www.ibm.com/xmlns/prod/scmp#resource resource-type: http://www.ibm.com/xmlns/prod/scmp#aspect Begriff, der einen Touchpoint-Typ eindeutig kennzeichnet: http://www.ibm.com/xmlns/prod/scmp#touchpoint-type	Ruft einen Eintrag für einen Touchpoint-Typ ab. Die Zuführungs-URL der Touchpoint-Instanz wird als Link mit dem Bezug "http://www.ibm.com/xmlns/prod/scmp#instance-feed" bereitgestellt. Die URL zur XML mit der Eigenschaftenblattdefinition wird als Link mit dem Bezug "http://www.ibm.com/xmlns/prod/scmp#property-sheet-definition" bereitgestellt. Anmerkung: Der virtuelle Touchpoint-Typ besitzt keine Eigenschaftenblattdefinition, da für Touchpoint-Instanzen mit der Rolle "Intermediary" keine Connectors konfiguriert werden müssen.	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend
1.1.1.1.1 Zuführung: Touchpoint-Instanzen Kategorien (Begriff: Schema): touchpoint: http://www.ibm.com/xmlns/prod/scmp#resource	Ruft die Liste der Einträge für die Touchpoint-Instanzen ab. Alle Einträge sind Referenzen auf die tatsächlichen Eintragsdokumente, die die verfügbaren Touchpoint-Instanzen darstellen.	Erstellt einen neuen Eintrag für eine Touchpoint-Instanz. Der Eintrag MUSS eine „Touchpoint-Konfiguration“ auf Seite 352 enthalten, die die „Touchpoint-Konfiguration“ auf Seite 352 enthält. Der Eintrag muss eine Kategorie aus dem Schema "http://www.ibm.com/xmlns/prod/scmp#touchpoint-role" enthalten.	Nicht zutreffend	Nicht zutreffend

Tabelle 32. Zulässige Operationen nach Ressourcen (Forts.)

Ressource	GET	POST	PUT	DELETE
<p>1.1.1.1.1.1 Eintrag: Touchpoint-Instanz</p> <p>Kategorien (Begriff: Schema):</p> <p>touchpoint: http://www.ibm.com/xmlns/prod/scmp#resource</p> <p>provider-tp ODER initiator-tp ODER intermediary-tp: http://www.ibm.com/xmlns/prod/scmp#aspect</p> <p>Begriff, der einen Touchpoint-Typ eindeutig kennzeichnet: http://www.ibm.com/xmlns/prod/scmp#touchpoint-type</p>	<p>Ruft den Eintrag für die Touchpoint-Instanz ab. Dieser Eintrag enthält drei Links zu den Ressourcen, die die Touchpoint-Instanz beschreiben:</p> <ul style="list-style-type: none"> • Eine URL für das Aktualisieren/ Löschen dieser Touchpoint-Instanz wird unter Verwendung eines Links mit dem Bezug "edit" bereitgestellt. • Eine URL für die Zielzuführung wird unter Verwendung eines Links mit dem Bezug "http://www.ibm.com/xmlns/prod/scmp#tp-destination" bereitgestellt. <p>Anmerkung: Touchpoint-Instanzen mit der Rolle "Provider" unterstützen keine Ziele, daher fehlt bei ihnen dieser Link.</p> <ul style="list-style-type: none"> • Eine URL zum Statureintrag wird unter Verwendung eines Links mit dem Bezug "http://www.ibm.com/xmlns/prod/scmp#status" bereitgestellt. • Eine URL zum Eintrag für den Touchpoint-Typ, zu dem diese Touchpoint-Instanz gehört. Der Link hat den Bezug "http://www.ibm.com/xmlns/prod/scmp#resource-type". 	Nicht zutreffend	Aktualisiert den Eintrag für die Touchpoint-Instanz. Der bereitgestellte Eintrag muss die vollständige Kopie des Eintrags und nicht nur die Änderungen enthalten. Die Rolle der Touchpoint-Instanz kann mit dieser Operation nicht geändert werden. Die Operation startet eine aktive Touchpoint-Instanz erneut, um sie zu rekonfigurieren.	Löscht den Eintrag für die Touchpoint-Instanz.
<p>1.1.1.1.1.1.1 Zuführung: Touchpoint-Ziele</p>	<p>Ruft die Zielzuführung für die Touchpoint-Instanz ab. Diese Zuführung kann mehrere Zieleinträge für die Touchpoint-Instanz enthalten.</p>	Erstellt einen neuen Zieleintrag für die Touchpoint-Instanz. Der Eintrag MUSS ein Datenelement enthalten, das die URL für den Anforderungsausgang des Ziels konfiguriert.	Nicht zutreffend	Nicht zutreffend
<p>1.1.1.1.1.1.1 Eintrag: Touchpoint-Ziel</p> <p>Kategorien (Begriff: Schema):</p> <p>tp-destination: http://www.ibm.com/xmlns/prod/scmp#resource</p>	<p>Ruft den Eintrag für das Touchpoint-Ziel ab, der die URL für den Anforderungsausgang an den fernen HTTP-Service im Element <{http://www.ibm.com/xmlns/prod/scmp}:data/> enthält. Dieser Eintrag stellt einen Link mit dem Bezug "edit" bereit, um Aktualisierungs-/ Löschvorgänge für diese Ressource zu ermöglichen.</p>	Nicht zutreffend	Aktualisiert den Eintrag für die Touchpoint-Instanz. Der bereitgestellte Eintrag muss die vollständige Kopie des Eintrags und nicht nur die Änderungen enthalten. Die Rolle der Touchpoint-Instanz kann mit dieser Operation nicht geändert werden. Diese Operation startet eine aktive Touchpoint-Instanz NICHT erneut, um die URL für den Anforderungsausgang zu ändern.	Löscht den Eintrag für das Touchpoint-Ziel.
<p>1.1.1.1.1.2 Eintrag: Status</p> <p>Kategorien (Begriff: Schema):</p> <p>touchpoint: http://www.ibm.com/xmlns/prod/scmp#resource</p> <p>status: http://www.ibm.com/xmlns/prod/scmp#aspect</p>	<p>Ruft den Eintrag für den Status der Touchpoint-Instanz ab, der den Betriebsstatus der betreffenden Touchpoint-Instanz beschreibt. Er ist im Element <{http://www.ibm.com/xmlns/prod/scmp}:data/> enthalten.</p> <p>Anmerkung: Bei Touchpoint-Instanzen mit der Rolle "Provider" und "Intermediary" enthält der Status auch 1 URL für den Anforderungseingang, die von Clients beim Abfragen der Touchpoint-Instanz verwendet werden sollte.</p>	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend

Tabelle 32. Zulässige Operationen nach Ressourcen (Forts.)

Ressource	GET	POST	PUT	DELETE
1.1.1.1.2 XML: Eigenschaftenblattdefinition	Ruft die „Eigenschaftenblattdefinitionen“ auf Seite 357 für den ausgewählten Touchpoint-Typ ab. Sie enthält das Schema eines IBM Security Directory Integrator-Connectors in Form eines XML-Dokuments.	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend

Touchpoint-Konfiguration

Sie können das Schema der Konfigurationsdaten bereitstellen, um eine Touchpoint-Instanz zu konfigurieren und zu starten.

Jedes Element der Konfigurationsdaten ist im Atom-Dokument innerhalb eines Elements `<data>` platziert. Dieses Element "data" muss zum folgenden Namensbereich gehören: `http://www.ibm.com/xmlns/prod/scmp`.

Instanzkonfiguration

Nachstehend erhalten Sie Informationen zu Konfigurationsdaten.

Diese Konfigurationsdaten geben Folgendes an:

- Die Rolle, in der die Touchpoint-Instanz ausgeführt wird. Die Touchpoint-Instanz entscheidet anhand dieser Daten, welche Fertigungslinie aus der Schablone zu verwenden ist. Im nachfolgenden Atom-Dokument ist sie durch das Token "{role}" im Element "category" angegeben. Unterstützte Werte sind `provider-tp`, `initiator-tp` und `intermediary-tp`.
- Der Verwaltungsstatus der erstellten Touchpoint-Instanz. Er ist im Atom-Dokument durch das Token "{admin_state}" gekennzeichnet. Die unterstützten Werte sind `enabled` und `disabled`.
- Die Eigenschaftenblatt-XML, die die Konfigurationsparameter für den Service-Connector der Touchpoint-Instanz enthält. Das Element {param_name} wird bei Standardtypen durch die Eigenschaftenblattdefinition des ausgewählten Touchpoint-Typs bzw. bei angepassten Typen durch die Eigenschaftenblattdefinition des Service-Connectors bestimmt. Das Element {param_value} wird je nach gewünschter Konfiguration durch den Benutzer bestimmt. Neben den Konfigurationsparametern können Sie auch den Modus des Service-Connectors festlegen. Hierzu geben Sie einen Parameter mit der Angabe "\$initMode" für das Element {param_name} und einen Zeichenfolgewart mit dem Modusnamen an (z. B. `Iterator`, `AddOnly`). Details zu diesem Parameter enthält der Abschnitt „Eigenschaftenblattdefinitionen“ auf Seite 357.
- Die beiden Parameter **{TouchpointID}** und **{version}** werden bereitgestellt, damit Sie zusätzliche Informationen zur jeweiligen Touchpoint-Instanz angeben können, die nur für den Ersteller von Bedeutung sind. Der Ersteller muss gewährleisten, dass diese Werte im Kontext der Clientanwendung gültig sind. Diese Werte werden vom Touchpoint-Server nicht interpretiert, sondern nur persistent gespeichert.

Das folgende Atom-Dokument wird bei der Erstellung einer Eintragsressource für eine Touchpoint-Instanz mit POST übergeben:

```
<entry xmlns="http://www.w3.org/2005/Atom">
  <id>{id}</id>
  <title>Touchpoint Instance Title</title>
  <author><name>Author Name</name></author>
  <content/>
  <category term="{role}" scheme="http://www.ibm.com/xmlns/prod/scmp#touchpoint-role" />
  <scmp:data
    xmlns:scmp="http://www.ibm.com/xmlns/prod/scmp"
```

```

      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/scmp
      http://localhost:1098/tp/schema/touchpoint.xsd" >
<scmp:touchpoint>
  <scmp:admin-state>{admin_state}</scmp:admin-state>
  <touchpointID>{touchpoint_id}</touchpointID>
  <version>{version}</version>
  <scmp:propertySheet>
    <scmp:property propertyName="{param_name}">
      <scmp:value>{param_value}</scmp:value>
    </scmp:property>
    ...
  </scmp:propertySheet>
</scmp:touchpoint>
</scmp:data>
</entry>

```

Bitte beachten Sie, dass anstelle des Tokens `{id}` auch eine ID für die erstellte Touchpoint-Instanz angegeben werden kann. Ungeachtet des übergebenen Wertes überschreibt der Touchpoint-Server diesen Wert jedoch durch einen automatisch generierten Wert. Auf diese Weise wird die Eindeutigkeit für die ID der Touchpoint-Instanz sichergestellt.

Zielkonfiguration

Nachstehend erhalten Sie Informationen zum Hinzufügen eines Ziels in den Touchpoint-Instanzen durch die Bearbeitung eines Atom-Dokuments.

Für eine Touchpoint-Instanz mit der Rolle "Intermediary" und "Initiator" muss ein Ziel konfiguriert werden, damit die Instanz betriebsbereit ist. Darüber hinaus unterstützen Instanzen mit diesen Rollen mehrere solcher Ziele sowie das Hinzufügen und Entfernen von Zielen während der Laufzeit.

Hierzu wird mit POST ein Atom-Dokument an die Zuführungs-URL für das Touchpoint-Ziel der erstellten Touchpoint-Instanz übergeben. Bitte denken Sie daran, dass für Touchpoint-Instanzen mit der Rolle "Provider" eine solche URL nicht verfügbar ist. Das Atom-Dokument sollte etwa so aussehen:

```

<entry xmlns="http://www.w3.org/2005/Atom">
  <scmp:data
  xmlns:scmp="http://www.ibm.com/xmlns/prod/scmp"
    xsi:schemaLocation="http://www.ibm.com/xmlns/prod/scmp
    http://localhost:1098/tp/schema/touchpoint.xsd" >
    <scmp:destination>
      <scmp:request-out>{request-out_URL}</scmp:request-out>
      <scmp:request-error>{request-error_URL}</scmp:request-error>
    </scmp:destination>
  </scmp:data>
</entry>

```

Aus diesem Snippet ist ersichtlich, dass nur das Token `{request-out_URL}` zum Konfigurieren eines Ziels benötigt wird. Das Token `{request-error_URL}` ist optional.

Kommunikationsprotokoll der Touchpoint-Instanz

Nachstehend erhalten Sie Informationen zu dem Protokoll, das für die Kommunikation mit einer Touchpoint-Instanz verwendet wird. Eine Touchpoint-Instanz wird in den meisten Fällen aus der Touchpoint-Schablone abgeleitet.

Touchpoint-Instanz mit der Rolle "Provider"

Mit der Touchpoint-Instanz mit der Rolle "Provider" können Sie die nachstehend beschriebenen HTTP-Methoden verarbeiten.

Tabelle 33. HTTP-Methoden für Touchpoint-Rolle "Provider"

HTTP-Methode	URL-Abfrageparameter	Connectormodus	Inhalt der HTTP-Anforderung	Inhalt der HTTP-Antwort	Code der HTTP-Antwort
GET	-	Iterator	-	Alle Einträge gefunden	"200 OK", falls mindestens 1 Eintrag gefunden wurde. "404 Not Found", falls keine Einträge gefunden wurden.
GET	Verknüpfungsbedingungen	Lookup	-	Alle Einträge gefunden	"200 OK", falls mindestens 1 Eintrag gefunden wurde. "404 Not Found", falls keine Einträge gefunden wurden.
POST	-	AddOnly	Hinzuzufügender Eintrag	-	"201 Created", falls die Operation erfolgreich ausgeführt wurde.
PUT*	Verknüpfungsbedingungen	Update	Eintrag mit aktualisierten Attributen	-	"201 Created", falls der Eintrag nicht vorhanden war und hinzugefügt wurde. "204 No Content", falls ein einziger Eintrag mit den Verknüpfungsbedingungen übereinstimmt und der Eintrag erfolgreich aktualisiert wurde.
DELETE	Verknüpfungsbedingungen	Delete	-	-	"204 No Content", falls die Operation erfolgreich ausgeführt wurde (die Operation schlägt fehl, wenn mehrere Einträge mit den Verknüpfungsbedingungen übereinstimmen).

(*) Bitte beachten Sie, dass zwischen der vorliegenden Implementierung und der HTTP 1.1-Spezifikation (<http://tools.ietf.org/html/rfc2616#section-9.6>) Unterschiede hinsichtlich der Verarbeitung der Methode PUT bestehen. Gemäß der HTTP-Spezifikation ersetzt eine Anforderung PUT die gesamte Ressource. In der vorliegenden Implementierung wird bei einem vorhandenen Eintrag nicht der gesamte Eintrag ersetzt, sondern es werden nur die angegebenen Attribute ersetzt.

Die Verknüpfungsbedingungen für die Connectoroperationen werden aus den Abfrageparametern der angeforderten URL abgeleitet. Beispiel: Eine Anforderung GET mit der URL "http://localhost/mytp?username=jsmith" führt zu einer Suchoperation mit der Verknüpfungsbedingung "username=jsmith". Jeder Abfrageparameter entspricht einer Bedingung für eine EXAKTE Übereinstimmung. Die aus mehreren Abfrageparametern abgeleiteten Bedingungen werden mit dem logischen Operator AND kombiniert. Beispiel: "?firstname=john&age=50" entspricht ((firstname equals "john") AND (age equals "50")).

Für Anforderungen PUT und DELETE sind Abfrageparameter erforderlich. Bei Anforderungen POST werden Abfrageparameter nicht erwartet. Falls eine Anforderung GET Abfrageparameter enthält, wird sie in eine Operation für den Suchmodus umgesetzt. Andernfalls wird sie in eine Operation für den Iteratormodus umgesetzt.

Bei Anforderungen GET können Sie einen optionalen HTTP-Header namens "X-TDI-TP-SizeLimit" verwenden, um die Anzahl der zurückgegebenen Einträge zu begrenzen. Der Wert des Headers muss eine ganze Zahl größer als 0 sein.

Alle durch die Touchpoint-Standardschablone interpretierten HTTP-Methoden sind im Hinblick auf ihre Sicherheits- und Idempotenzmerkmale mit der HTTP-Spezifikation konform.

Touchpoint-Instanz mit der Rolle "Initiator"

Nachstehend erhalten Sie Informationen zu der Touchpoint-Instanz mit der Rolle "Initiator".

Eine Touchpoint-Instanz mit der Rolle "Initiator" agiert wie ein HTTP-Client. Sie besitzt einen Iteratorconnector, der Eintragsobjekte erzeugt, die von der Fertigungslinie an konfigurierte Ziel-URLs gesendet werden. Für jeden Eintrag wird eine einzelne Anforderung POST gesendet, deren Inhalt eine „Darstellung von Eintragsobjekten als HTTP-Inhalt“ ist.

Touchpoint-Instanz mit der Rolle "Intermediary"

Sie können die Touchpoint-Instanz mit der Rolle "Intermediary" als Vermittler zwischen mehreren Touchpoint-Instanzen verwenden.

Die Touchpoint-Instanz mit der Rolle "Intermediary" ähnelt sowohl der Rolle "Provider" als auch der Rolle "Initiator". Sie akzeptiert Anforderungen wie eine Touchpoint-Instanz mit der Rolle "Provider" an einer bestimmten URL für den Anforderungseingang und sendet die empfangenen Daten wie eine Touchpoint-Instanz mit der Rolle "Initiator" an mehrere Ziele. Aufgrund dieser Weiterleitungsfunktionalität kann sie als Vermittler zwischen mehreren Touchpoint-Instanzen der anderen Rollen verwendet werden.

Darstellung von Eintragsobjekten als HTTP-Inhalt

Mithilfe des hier aufgeführten Beispiels können Sie Eintragsobjekte als HTTP-Inhalt anzeigen.

Beispiel:

```
<tp:data xmlns:tp="http://www.ibm.com/xmlns/prod/tdi/72/tp">
  <tp:entry>
    <tp:attribute name="username">
      <tp:value><![CDATA[jsmith]]</tp:value>
    </tp:attribute>

    <tp:attribute name="mail">
      <tp:value>jsmith@ibm.us.com</tp:value>
      <tp:value>john.smith@gmail.com</tp:value>
    </tp:attribute>
  </tp:entry>
</tp:data>
```

Der HTTP-Inhalt muss in UTF-8 codiert sein. Er muss ein einziges Element data enthalten, das 0 oder mehr Elemente entry enthalten kann.

XML-Schemabeschreibung für das Touchpoint-Datenformat

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  targetNamespace="http://www.ibm.com/xmlns/prod/tdi/72/tp"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  xmlns=http://www.w3.org/2001/XMLSchema xmlns:tns=http://www.ibm.com/xmlns/prod/tdi/72/tp >

  <element name="data" type="tns:TouchpointDataType" />

  <element name="entry" type="tns:EntryType" />

  <element name="attribute" type="tns:AttributeType" />

  <element name="property" type="tns:PropertyType" />

  <element name="value" type="string" />
```

```

<complexType name="TouchpointDataType">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element ref="tns:entry" />
  </choice>
</complexType>

<complexType name="EntryType">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element ref="tns:property" />
    <element ref="tns:attribute" />
  </choice>
</complexType>

<complexType name="AttributeType">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element ref="tns:property" />
    <element ref="tns:attribute" />
    <element ref="tns:value" />
  </choice>
  <attribute name="name" type="string" use="required" />
  <attribute name="namespaceURI" type="anyURI" />
</complexType>

<complexType name="PropertyType">
  <simpleContent>
    <extension base="string">
      <attribute name="name" type="string" use="required" />
      <attribute name="namespaceURI" type="anyURI" />
    </extension>
  </simpleContent>
</complexType>
</schema>

```

Schema für Touchpoint-Statuseintrag

Sie können den Status der Touchpoint-Instanz durch das Senden einer HTTP-Anforderung GET an die URL des Statuseintrags abrufen.

„Kommunikationsprotokoll der Touchpoint-Instanz“ auf Seite 353.

```

<entry xmlns="http://www.w3.org/2005/Atom">
  <id>Status ID</id>
  <link href="{touchpoint_instance_status_URL}" type="application/atom+xml;type=entry"
    rel="self"/>
  <category scheme="http://www.ibm.com/xmlns/prod/scmp#resource" term="touchpoint"/>
  <category scheme="http://www.ibm.com/xmlns/prod/scmp#aspect" term="status"/>
  <scmp:data
    xmlns:scmp="http://www.ibm.com/xmlns/prod/scmp"
    xsi:schemaLocation="http://www.ibm.com/xmlns/prod/scmp
      http://localhost:1098/tp/schema/touchpoint.xsd" >
    <scmp:touchpoint-status>
      <scmp:request-in>{request-in_URL}</scmp:request-in>
      <scmp:op-state>{op-state}</scmp:op-state>
    </scmp:touchpoint-status>
  </scmp:data>
</entry>

```

Das zurückgegebene Atom-Eintragsdokument enthält ein Element **<data>**, das den Status der Touchpoint-Instanz beschreibt. Das Element "data" gehört zum Namensbereich `http://www.ibm.com/xmlns/prod/scmp`. Es hat die folgende Syntax:

- **{op-state}**: Dieses Zeichenfolgeschlüsselwort beschreibt den aktuellen Status der Touchpoint-Instanz wie im Abschnitt „Touchpoint-Instanz“ auf Seite 338 definiert.
- **{request-in_URL}**: Dies ist die URL für den Zugriff auf Touchpoint-Instanzen mit der Rolle "Provider" und "Intermediary".

Eigenschaftenblattdefinitionen

Die Eigenschaftenblattdefinition ist ein XML-Dokument, das das Schema eines IBM Security Directory Integrator-Connectors bestimmt. Nachstehend erhalten Sie weitere Informationen dazu.

Es kann über einen Link im Eintrag für den Touchpoint-Typ abgerufen werden (siehe „Touchpoint-Schema“ auf Seite 348).

Die Eigenschaftenblattdefinition variiert je nach Touchpoint-Typ:

- Bei einem Standardtyp enthält sie das Schema des IBM Security Directory Integrator-Connectors, der dem Touchpoint-Typ entspricht.
- Bei einem angepassten Typ enthält sie das Schema des Service-Connectors in der angepassten Touchpoint-Schablone.
- Bei einem virtuellen Typ (`virtual://Intermediary`) gibt es keine Eigenschaftenblattdefinition, da mit keinem Drittanbietersystem kommuniziert wird (diese Rolle verwendet ausschließlich HTTP).

Neben den Schemaparametern eines IBM Security Directory Integrator-Connectors enthält die Eigenschaftenblattdefinition die vom Connector unterstützten Modi. Diese werden als optionale Werte für die Eigenschaftsdefinition (`propertyDefinition`) mit dem Namen `$initMode` gespeichert. Ihre Werte stimmen direkt mit den Namen der Connectormodi überein (`Iterator`, `AddOnly`, `CallReply` usw.).

Das folgende Beispiel zeigt eine Eigenschaftenblattdefinition für den Dateiconnector (Touchpoint-Typ `system:/Connectors/ibmdi.LDAP`):

```
<?xml version="1.0" encoding="UTF-8"?>
<propertySheetDefinition xmlns="http://www.ibm.com/xmlns/prod/scmp">
  <propertyDefinition required="true" hidden="false" readOnly="false"
    propertyType="string" multiple="false"
    propertyName="ldapUrl">
    <label label="LDAP URL" lang="en"/>
    <!--one label for the different languages supported by TDI -->
  </propertyDefinition>
  <!--the rest of LDAP Connector's parameters -->
  <propertyDefinition required="false" readOnly="false" propertyType="string"
    multiple="false" propertyName="$initMode">
    <label label="$initMode" lang="en"/>
    <!--one label for the different languages supported by TDI -->
    <option>
      <value>AddOnly</value>
      <label label="AddOnly" lang="en"/>
      <!--one label for the different languages supported by TDI -->
    </option>
    <option>
      <value>Iterator</value>
      <label label="Iterator" lang="en"/>
      <!--one label for the different languages supported by TDI -->
    </option>
    <!--the rest of modes supported by the LDAP Connector -->
  </propertyDefinition>
</propertySheetDefinition>
```

Zur Verkürzung der Liste ist im Beispiel nur einer der Konfigurationsparameter und die Definition der Eigenschaft "`$initMode`" dargestellt. Wie erkennbar ist, besitzt dieser Connector einen Parameter namens `ldapUrl`, der erforderlich ist und dessen Wert aus einer Zeichenfolge besteht. Auch die im Konfigurationseditor angezeigte englische Bezeichnung **LDAP URL** ist zu sehen (die Bezeichnungen für die übrigen von IBM Security Directory Integrator unterstützten Sprachen sind hier nicht dargestellt). Der Parameter "`$initMode`" ist ebenfalls vorhanden. Aus seinen optionalen Werten geht hervor, dass dieser Connector (neben mehreren anderen, hier nicht gezeigten Modi) sowohl den Modus "Iterator" als auch den Modus "AddOnly" unterstützt.

Eigenschaftentblattd Definitionen sind beim Erstellen von Touchpoint-Instanzen von großem Nutzen, da Sie das Schema des Connectors nicht vorab kennen müssen. Ausgehend von den Informationen in diesen Definitionen können Sie diese Task in vielerlei Hinsicht genauso wie die Konfiguration von Connectors im Konfigurationseditor von IBM Security Directory Integrator ausführen, nämlich durch Anzeigen der erforderlichen Parameter, durch Auswählen der erwarteten Werte (Zeichenfolge, Zahl oder Gruppe vordefinierter Werte) und durch deren Angabe in der Konfiguration.

XML-Schemapositionen

Nachstehend erhalten Sie Informationen zum Definieren von XML-Schemapositionen.

Für jedes Element `scmp:data` wird ein XML-Schemadokument bereitgestellt. Im Dokument sind alle Elemente definiert, die innerhalb des Elements `scmp:data` auftreten.

Die Position des Schemadokuments ist in einem Attribut `"xsi:schemaLocation"` (Position der XML-Schemainstanz) angegeben, das im Element `scmp:data` definiert ist. Beispiel:

```
<scmp:data
  xmlns:scmp="http://www.ibm.com/xmlns/prod/scmp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/xmlns/prod/scmp
http://localhost:1098/tp/schema/tdi-connectivity-provider.xsd">

  <scmp:connectivity-provider>
```

Die Position des Schemas ist eine gültige URL, die durch Web-Clients dereferenziert werden kann: Jeder Client des Touchpoint-Servers kann die im Attribut `"xsi:schemaLocation"` angegebene URL auflösen und auf das tatsächliche Schemadokument zugreifen.

Falls das Schema eine Referenz auf ein weiteres Schemadokument enthält (z. B. durch ein XML-Schemaelement `"import"`, `"include"` oder `"redefine"`), können Web-Clients darüber hinaus durch das Auflösen der URL in der Referenz das referenzierte Schema abrufen.

Abläufe bei einem Fehler

Falls ein Fehler ausgelöst wird, werden entsprechende Fehlernachrichten an das Standardprotokoll ausgegeben. Nachstehend erhalten Sie Informationen zu möglichen Fehlersituationen und zur Syntax von XML-Dokumenten.

Mögliche Situationen für das Entstehen eines Fehlers:

- Falsche Konfiguration des Touchpoint-Servers
- Infolge von Kommunikationsfehlern mit dem Persistenzspeicher ausgelöste Ausnahmebedingungen (Dateisystemfehler)
- Fehler bei der Kommunikation mit den Clients des Touchpoint-Servers
- Fehler bei der Kommunikation eines der Touchpoint-Server mit IBM Security Directory Integrator

Wenn der Fehler darauf beruht, dass eine ungültige Information/Anforderung durch den Benutzer gesendet wurde, die das Protokoll des Touchpoint-Servers nicht einhält, wird ein XML-Dokument mit der folgenden Syntax erstellt:

```

<?xml version="1.0" encoding="UTF-8"?>
<ns2:error xmlns:ns2="http://www.ibm.com/xmlns/prod/scmp">
  <creation-time>2010-02-23T14:49:06.384+02:00</creation-time>
  <code>100005</code>
  <details>
    <detail>
      <name>schema</name>
      <value>http://www.ibm.com/xmlns/prod/scmp#touchpoint-role</value>
    </detail>
  </details>
  <native-msgid>ABCD1234E</native-msgid>
  <summary>Missing role category</summary>
</ns2:error>

```

Hierbei gilt Folgendes:

- Das Element **creation-time** gibt den Zeitpunkt für das Auftreten des Fehlers in einem Format an, das in "http://www.w3.org/TR/1998/NOTE-datettime-19980827" angegeben ist.
- Das Element **code** enthält einen der folgenden Codes:
 - 100000: Ein unbekannter Fehler, der nicht weiter eingegrenzt werden kann, ist aufgetreten.
 - 100001: Eine erforderliche Angabe "atom:link" fehlt. Das Element "details" stellt Folgendes bereit: **rel** (Name des Bezugs des erwarteten Links).
 - 100002: Das erforderliche Element "scmp:data" fehlt. Das Element "details" stellt Folgendes bereit: **qname** (QNAME für fehlendes Element)
 - 100003: Ungültiges Element "atom:entry" in Operation POST/PUT (z. B. Analysefehler)
 - 100004: Ungültiger Wert für Element "scmp:data". Das Element "details" stellt Folgendes bereit: **qname** (QNAME für ungültiges Element), **value** (ungültiger angegebener Wert)
 - 100005: Das erforderliche Element "atom:category" fehlt. Das Element "details" stellt Folgendes bereit: **scheme** (erwarteter Schemaname).
 - 100006: Der Wert des Elements "atom:category" ist ungültig. Das Element "details" stellt Folgendes bereit: **scheme** (erwarteter Schemaname); **term** (ungültiger angegebener Begriff).
 - 100007: Es gibt zu viele Elemente "atom:link" für den angegebenen Bezug. Das Element "details" stellt Folgendes bereit: **rel** (Name des Bezugs für den überzähligen Link).
 - 100008: Es gibt zu viele Werte "atom:category" für das Schema. Das Element "details" stellt Folgendes bereit: **scheme** (Name des Schemas mit zu vielen Werten).
 - 330000: Ein Standardfehler des speziellen Konnektivitätsproviders ist aufgetreten (der Fehler kann nicht weiter eingegrenzt werden).
- Das Element **details** enthält weitere Details für den speziellen Fehler. Anhand des entsprechenden Fehlercodes können Sie ermitteln, welche Art von ausführlichen Informationen zu jedem Fehler angegeben sind.
- Das Element **native-msgid** gibt die Kurz-ID der Nachricht an.
- Das Element **summary** enthält die für den Benutzer verständliche Fehlernachricht.

Wenn der Fehler nicht durch ein fehlerhaftes Protokoll, sondern durch eine andere Quelle verursacht wurde, wird die für den Benutzer verständliche Darstellung als einfaches Textformat zurückgegeben. Der Fehler enthält außerdem den Ausnahmsbedingungen-Stack-Trace, damit Sie ihn zur weiteren Behebung an den Administrator des Touchpoint-Servers melden können.

Konfiguration

Nachstehend erhalten Sie Informationen zum Konfigurieren des Touchpoint-Servers.

Der Touchpoint-Server wird innerhalb eines Web-Containers ausgeführt. Der mit IBM Security Directory Integrator ausgelieferte Standard-Web-Container wird durch die folgenden Eigenschaften in der Datei `global.properties` oder `solution.properties` konfiguriert:

- `tp.server.on`: Gibt an, ob der im Produktpaket enthaltene Web-Container und der Touchpoint-Server gestartet werden sollen. Standardwert: *false*.
- `tp.server.port`: Gibt den Port an, an dem der Web-Container empfangsbereit ist. Standardwert: 1098.
- `tp.server.auth`: Gibt an, ob der Touchpoint-Server die HTTP-Basisauthentifizierung verwendet. Standardwert: *false*.
- `tp.server.auth.realm`: Gibt die HTTP-Basisauthentifizierung für den Realm an. Standardwert: "IBM Security Directory Integrator Touchpoint Server".

Der Touchpoint-Server berücksichtigt zuerst den Wert der Eigenschaft `api.remote.bind.address`. Ist diese Eigenschaft nicht festgelegt, wird der Wert der Eigenschaft `com.ibm.di.default.bind.address` hinzugezogen. Auf diese Weise kann der Zugriff auf Hosts mit mehreren Ausgangspositionen wirksam gefiltert werden.

Der Web-Container kann SSL verwenden, um die Transportschicht zu schützen. Er verwendet die Einstellungen der fernen API wieder und wird durch Festlegung der Eigenschaft `api.remote.ssl` aktiviert. Die SSL-Clientauthentifizierung wird durch die Eigenschaft `api.remote.ssl.client.auth.on` aktiviert. Die SSL-Schlüssel des Servers werden mit den bekannten Eigenschaften der fernen API konfiguriert:

- `api.keystore`
- `api.client.keystore.pass`
- `api.client.key.pass`
- `api.client.keystore.type`

Die HTTP-Basisauthentifizierung (<http://tools.ietf.org/html/rfc2617>) kann mit den Eigenschaften `tp.server.auth` und `tp.server.auth.realm` konfiguriert werden. Standardmäßig ist sie inaktiviert. Weitere Informationen zur Authentifizierung können Sie dem Abschnitt „Authentifizierung“ auf Seite 361 entnehmen.

Die Konfiguration des Touchpoint-Servers wird unter Verwendung einer XML-Datei angegeben. Der Pfad zu dieser Datei ist in der Datei `global.properties` oder `solution.properties` mit der Eigenschaft `tp.server.config` angegeben. Im Verzeichnis `etc` der IBM Security Directory Integrator-Installation wird ein Beispiel für die Konfigurationsdatei des Touchpoint-Servers bereitgestellt.

Die folgende Syntax wird von der Konfigurationsdatei des Touchpoint-Servers verwendet:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tp:tpServerConfig xmlns:tp="http://www.ibm.com/xmlns/prod/tdi/72/tp" tp:version="1.0">

  <!-- specifies the encryption settings used when encrypting passwords -->
  <tp:encryptionConfig stash="idisrv.sth">
    <tp:keyStore>testserver.jks</tp:keyStore>
    <tp:keyStoreType>jks</tp:keyStoreType>
    <tp:keyAlias>server</tp:keyAlias>
    <tp:transformation>RSA</tp:transformation>
  </tp:encryptionConfig>

  <tp:templateConfig>
```



```

<tp:baseTemplate>etc/TouchpointTemplate.xml</tp:baseTemplate>

<!-- Specify the path to the directory that holds the Touchpoint templates. -->
<!--
<tp:customTemplatesDir>templates</tp:customTemplatesDir>
-->
</tp:templateConfig>

<!-- specifies the persistence settings that configure the place to persist the state -->
<tp:persistenceConfig>
  <tp:enabled>true</tp:enabled>
  <tp:location>tp_state</tp:location>
</tp:persistenceConfig>

<!-- configures the touchpoint providers (nodes) -->
<tp:nodeConfigs>
  <!-- Default connection to the local server -->
  <tp:tdiNodeConfig tp:local="true" tp:id="default">
    <!-- The host of the remote node which all
    Provider Touchpoint Instances will receive requests on -->
    <tp:providerHost>localhost</tp:providerHost>
    <!-- The port of the remote node which all
    Provider Touchpoint Instances will receive requests on -->
    <tp:providerPort>1097</tp:providerPort>

    <tp:title>Example Touchpoint Provider</tp:title>
    <tp:author>John Doe</tp:author>
    <tp:email>jdoe@example.org</tp:email>
    <tp:summary>Example Touchpoint Provider Atom Entry</tp:summary>

    <tp:contact>Local Administrator</tp:contact>
    <tp:location>Main building, 5th fl.</tp:location>
    <tp:organization>Example Organization</tp:organization>
  </tp:tdiNodeConfig>

  <!-- Here is an example of a remote server connection -->
  <!--
    <tp:tdiNodeConfig id="remote" local="false">
      <tp:title>Example Touchpoint Provider</tp:title>
      <tp:author>John Doe</tp:author>
      <tp:email>jdoe@example.org</tp:email>
      <tp:summary>Example Touchpoint Provider</tp:summary>

      <tp:host>localhost</tp:host>
      <tp:port>1099</tp:port>
      <tp:user>username</tp:user>
      <tp:password protect="true" encrypted="false">password</tp:password>

      <tp:contact>Jack Smith</tp:contact>
      <tp:location>5th fl.</tp:location>
      <tp:organization>Example Organization</tp:organization>

      <tp:providerHost>localhost</tp:providerHost>
      <tp:providerPort>1097</tp:providerPort>
    </tp:tdiNodeConfig>
  -->
</tp:nodeConfigs>
</tp:tpServerConfig>

```

Authentifizierung

Nachstehend erhalten Sie Informationen zu den Authentifizierungsaspekten des Touchpoint-Servers.

Es gibt zwei Fragestellungen, die beim Touchpoint-Server für die Authentifizierung von Bedeutung sind:

- Handelt es sich um einen HTTP-Server?
- Handelt es sich um einen Client der fernen RMI-Server-API der IBM Security Directory Integrator-Server, die als Konnektivitätsprovider konfiguriert sind?

Als HTTP-Server unterstützt der Touchpoint-Server die HTTP-Basisauthentifizierung von HTTP-Clients. Er verwendet keine separate Benutzerregistry. Stattdessen delegiert der Touchpoint-Server die Authentifizierungsanforderungen an die Ser-

ver-API des lokalen IBM Security Directory Integrator-Servers (also des Servers, der als Host für den Touchpoint-Server dient).

Als Client der fernen Server-API muss sich der Touchpoint-Server wie jeder andere Client der Server-API beim fernen IBM Security Directory Integrator-Server authentifizieren. Bitte beachten Sie, dass eine Authentifizierung nicht erforderlich ist, wenn der lokale IBM Security Directory Integrator-Server als Konnektivitätsprovider verwendet wird.

Weitere Angaben über die Authentifizierung bei der Server-API enthält der Anhang *Server API* im Handbuch *Referenzinformationen*.

Beispiele

Mithilfe der hier bereitgestellten Links können Sie auf die im Lieferumfang enthaltenen Beispiele zugreifen und die anderen Beispiele verwenden, um mit dem JDBC-Connector eine Touchpoint-Instanz zu erstellen.

Im Lieferumfang enthaltenes Beispiel

Mit den hier aufgeführten Schritten können Sie das im Lieferumfang enthaltene Beispiel zum Erstellen einer Touchpoint-Instanz verwenden.

Das wichtigste Beispiel, das die Verwendung der Funktionalität des Touchpoint-Servers in IBM Security Directory Integrator veranschaulicht und Beispielschritte für die Erstellung von Touchpoint-Instanzen mit den Rollen "Provider" und "Initiator" enthält, wird mit der Installation ausgeliefert. Die Schritte sind in dem unter *tdi-installationsverzeichnis/examples/Touchpoint-Client/Touchpoint_Example.pdf* zu findenden Dokument aufgeführt. Zusätzlich wird mit einer Beispielimplementierung dieser Schritte demonstriert, wie diese in der Programmiersprache Java aussehen könnten. Der Java-Code ist folgendermaßen in zwei Paketen strukturiert:

- **com.ibm.di.tp.client.api:** Dieses Paket enthält den Code, der veranschaulicht, wie die Kommunikation mit dem Touchpoint-Server ausgeführt wird. Es enthält ausschließlich Methoden für das Abfragen des Touchpoint-Servers.

Anmerkung: Dieser Code ist von Apache HttpClient v3.x abhängig.

- **com.ibm.di.tp.client.gui:** Dieses Paket enthält ein Beispiel für einen Benutzerschnittstellenclient, der das Paket "com.ibm.di.tp.client.api" verwendet, um zur Bereitstellung von Touchpoint-Instanzen mit dem Touchpoint-Server zu interagieren. Sie können diese Benutzerschnittstelle zu Testzwecken einsetzen, wenn Sie Touchpoint-Instanzen bereitstellen. Zum Starten dieses Dienstprogramms können Sie eines der bereitgestellten Scripts `startClient.bat` und `startClient.sh` verwenden.

Beispielschritte für die Erstellung einer Touchpoint-Instanz mit einem JDBC-Connector

Mit den hier aufgeführten Anweisungen können Sie eine Touchpoint-Instanz bereitstellen.

Um eine Touchpoint-Instanz bereitzustellen, müssen Sie an den Touchpoint-Server eine HTTP-Anforderung POST senden. Die URLs für die Anforderungen sind gemäß der Anwendung „Touchpoint-Schema“ auf Seite 348 abrufbar. Für die Zwecke des folgenden Beispiels werden Pseudo-URLs wie <Resource Name URL> verwendet. Die geeignete URL können Sie zur Laufzeit abrufen.

Touchpoint-Instanz mit der Rolle "Provider"

Mithilfe des bereitgestellten Beispiels können Sie eine Eintragsressource für die Touchpoint-Instanz erstellen.

POST <Touchpoint Instance Feed URL>

```
Body:<entry xmlns="http://www.w3.org/2005/Atom">
  <id>some ID</id>
  <title>Provider Touchpoint Instance</title>
  <author><name>author_name</name></author>
  <content/>
  <category term="provider-tp" scheme="http://www.ibm.com/xmlns/prod/scmp#touchpoint-role" />
  <scmp:data xmlns:scmp="http://www.ibm.com/xmlns/prod/scmp" >
    <scmp:touchpoint>
      <scmp:propertySheet>
        <scmp:property propertyName="jdbcSource">
          <scmp:value>some source value</scmp:value>
        </scmp:property>
        <scmp:property propertyName="jdbcDriver">
          <scmp:value>the driver class</scmp:value>
        </scmp:property>
        <scmp:property propertyName="jdbcTable">
          <scmp:value>the table name</scmp:value>
        </scmp:property>
        <!--The rest of the parameters required by a JDBC Connector-->
      </scmp:propertySheet>
      <scmp:admin-state>enabled</scmp:admin-state>
    </scmp:touchpoint>
  </scmp:data>
</entry>
```

Der Touchpoint-Server gibt eine Antwort zurück, die etwa wie folgt lautet:

```
201 Created
Location: <Touchpoint Instance Entry URL>
Body:
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:ns2="http://a9.com/-/spec/opensearch/1.1/"
xmlns:ns3="http://www.w3.org/1999/xhtml">
  <id>generated_id</id>
  <updated>2010-02-17T18:33:55.302+02:00</updated>
  <title>Provider Touchpoint Instance</title>
  <link href="<Touchpoint Instance Entry URL>" type="application/atom+xml;type=entry" rel="self"/>
  <link href="<Touchpoint Instance Entry URL>" type="application/atom+xml;type=entry" rel="edit"/>
  <link href="<Touchpoint Type Entry URL>"
type="application/atom+xml;type=entry" rel="http://www.ibm.com/xmlns/prod/scmp#resource-type"/>
  <link href="<Touchpoint Instance Status Entry URL>" type="application/atom+xml;type=entry"
rel="http://www.ibm.com/xmlns/prod/scmp#status"/>
  <author><name>author_name</name></author>
  <category scheme="http://www.ibm.com/xmlns/prod/scmp#touchpoint-type" term="system:/Connectors/ibmDI.JDBC"/>
  <category scheme="http://www.ibm.com/xmlns/prod/scmp#touchpoint-role" term="provider-tp"/>
  <category scheme="http://www.ibm.com/xmlns/prod/scmp#resource" term="touchpoint"/>
  <ns2:data xmlns:ns2="http://www.ibm.com/xmlns/prod/scmp">
    <ns2:touchpoint>
      <ns2:admin-state>enabled</ns2:admin-state>
      <ns2:propertySheet>
        <ns2:property propertyName="jdbcSource" xmlns="" xmlns:ns5="http://www.w3.org/2005/Atom">
          <ns2:value>some source value</ns2:value>
        </ns2:property>
        <ns2:property propertyName=" jdbcDriver " xmlns="" xmlns:ns5="http://www.w3.org/2005/Atom">
          <ns2:value>the driver class</ns2:value>
        </ns2:property>
        <!--The rest of the parameters required by a JDBC Connector-->
      </ns2:propertySheet>
    </ns2:touchpoint>
  </ns2:data>
```

Beachten Sie, dass der Touchpoint-Server die ID des Eintrags ändert, um ihre Eindeutigkeit zu gewährleisten.

Die für den Zugriff auf die erstellte Touchpoint-Instanz verwendete URL kann über die URL für den Statuseintrag abgerufen werden. Hierzu senden Sie eine HTTP-Anforderung GET an die URL <Touchpoint Instance Status Entry URL>. Die empfangene Antwort sieht etwa folgendermaßen aus:

```
200 OK
Location: <Touchpoint Instance Status Entry URL>
Body:
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:ns2="http://a9.com/-/spec/opensearch/1.1/"
xmlns:ns3="http://www.w3.org/1999/xhtml">
```

```

<id>generated_id</id>
<link href="Touchpoint Instance Status Entry URL" type="application/atom+xml;type=entry" rel="self"/>
<category scheme="http://www.ibm.com/xmlns/prod/scmp#resource" term="touchpoint"/>
<category scheme="http://www.ibm.com/xmlns/prod/scmp#aspect" term="status"/>
<ns2:data xmlns:ns2="http://www.ibm.com/xmlns/prod/scmp">
  <ns2:touchpoint-status>
    <ns2:request-in>Touchpoint Provider Request-in URL</ns2:request-in>
    <ns2:op-state>available</ns2:op-state>
  </ns2:touchpoint-status>
</ns2:data>
</entry>

```

Touchpoint-Instanz mit der Rolle "Initiator"

Mithilfe des bereitgestellten Beispiels können Sie eine Touchpoint-Instanz mit der Rolle "Initiator" erstellen.

POST <Touchpoint Instance Feed URL>

```

Body:
<entry xmlns="http://www.w3.org/2005/Atom">
  <id>some ID</id>
  <title>Initiator Touchpoint Instance</title>
  <author><name>author_name</name></author>
  <content/>
  <category term="initiator-tp" scheme="http://www.ibm.com/xmlns/prod/scmp#touchpoint-role" />
  <scmp:data xmlns:scmp="http://www.ibm.com/xmlns/prod/scmp" >
    <scmp:touchpoint>
      <scmp:propertySheet>
        <scmp:property propertyName="jdbcSource">
          <scmp:value>some source value</scmp:value>
        </scmp:property>
        <scmp:property propertyName="jdbcDriver">
          <scmp:value>the driver class</scmp:value>
        </scmp:property>
        <scmp:property propertyName="jdbcTable">
          <scmp:value>the table name</scmp:value>
        </scmp:property>
        <!--The rest of the parameters required by a JDBC Connector-->
      </scmp:propertySheet>
      <scmp:admin-state>enabled</scmp:admin-state>
    </scmp:touchpoint>
  </scmp:data>
</entry>

```

Der Touchpoint-Server gibt eine Antwort zurück, die etwa wie folgt lautet:

```

201 Created
Location: <Touchpoint Instance Entry URL>
Body:
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:ns2="http://a9.com/-/spec/opensearch/1.1/"
xmlns:ns3="http://www.w3.org/1999/xhtml">
  <id>generated_ID</id>
  <updated>2010-02-17T18:33:55.302+02:00</updated>
  <title>Initiator Touchpoint Instance</title>
  <link href="Touchpoint Instance Entry URL" type="application/atom+xml;type=entry" rel="self"/>
  <link href="Touchpoint Instance Entry URL" type="application/atom+xml;type=entry" rel="edit"/>
  <link href="Touchpoint Type Entry URL"
type="application/atom+xml;type=entry" rel="http://www.ibm.com/xmlns/prod/scmp#resource-type"/>
  <link href="Touchpoint Instance Status Entry URL"
type="application/atom+xml;type=entry" rel="http://www.ibm.com/xmlns/prod/scmp#status"/>
  <link href="Touchpoint Instance Destination Feed URL"
type="application/atom+xml;type=feed" rel="http://www.ibm.com/xmlns/prod/scmp#tp-destination"/>
  <author><name>author_name</name></author>
  <category scheme="http://www.ibm.com/xmlns/prod/scmp#touchpoint-type" term="system:/Connectors/ibmdi.JDBC"/>
  <category scheme="http://www.ibm.com/xmlns/prod/scmp#touchpoint-role" term="initiator-tp"/>
  <category scheme="http://www.ibm.com/xmlns/prod/scmp#resource" term="touchpoint"/>
  <ns2:data xmlns:ns2="http://www.ibm.com/xmlns/prod/scmp">
    <ns2:touchpoint>
      <ns2:admin-state>enabled</ns2:admin-state>
    <ns2:propertySheet>
      <ns2:property propertyName="jdbcSource" xmlns="" xmlns:ns5="http://www.w3.org/2005/Atom">
        <ns2:value>some source value</ns2:value>
      </ns2:property>
      <ns2:property propertyName="jdbcDriver" xmlns="" xmlns:ns5="http://www.w3.org/2005/Atom">
        <ns2:value>the driver class</ns2:value>
      </ns2:property>
      <!--The rest of the parameters required by a JDBC Connector-->
    </ns2:propertySheet>
  </ns2:touchpoint>
</ns2:data>
<content/>
</entry>

```

Beachten Sie, dass der Touchpoint-Server die ID des Eintrags ändert, um ihre Eindeutigkeit zu gewährleisten.

Darüber hinaus enthält die Antwort des Touchpoint-Servers dieses Mal eine Zuführungs-URL für das Ziel der Touchpoint-Instanz, die zum Konfigurieren der Touchpoint-Ziele benötigt wird.

Als Nächstes wird ein Ziel zur Touchpoint-Instanz mit der Rolle "Initiator" hinzugefügt:

```
POST <Touchpoint Instance Destination Feed>

Body:
<entry xmlns="http://www.w3.org/2005/Atom">
  <scmp:data xmlns:scmp="http://www.ibm.com/xmlns/prod/scmp" >
    <scmp:destination>
      <scmp:request-out>Request-out URL</scmp:request-out>
    </scmp:destination>
  </scmp:data>
</entry>
```

Der Touchpoint-Server gibt eine Antwort zurück, die etwa wie folgt lautet:

```
201 Created
Location: <Touchpoint Instance Destination Entry URL>
Body:
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:ns2="http://a9.com/-/spec/opensearch/1.1/"
xmlns:ns3="http://www.w3.org/1999/xhtml">
  <id>generated_id</id>
  <updated>2010-02-18T10:52:35.108+02:00</updated>
  <link href="<Touchpoint Instance Destination Entry URL>" type="application/atom+xml;type=entry" rel="self"/>
  <link href="<Touchpoint Instance Destination Entry URL>" rel="edit"/>
  <category scheme="http://www.ibm.com/xmlns/prod/scmp#resource" term="tp-destination"/>
  <ns2:data xmlns:ns2="http://www.ibm.com/xmlns/prod/scmp">
    <ns2:destination>
      <ns2:request-out>Request-out URL</ns2:request-out>
    </ns2:destination>
  </ns2:data>
</entry>
```

An diesem Punkt wird die Ausführung der Touchpoint-Instanz mit der Rolle "Initiator" gestartet.

Touchpoint-Instanz mit der Rolle "Intermediary"

Die zur Erstellung einer Touchpoint-Instanz mit dieser Rolle erforderlichen Schritte stellen eine Kombination der Schritte für die Rollen "Provider" und "Initiator" dar.

Als Erstes wird eine Eintragsressource für die Touchpoint-Instanz erstellt. In diesem Fall gibt es eine konkrete Zuführungs-URL für die Touchpoint-Instanz, nämlich die URL "http://<tp-server-host>:<tp-server-port>/<kontextstammverzeichnis>/tp-node/default/tp-type/virtual__Intermediary/tp-inst".

```
POST <Touchpoint Instance Feed URL>

Body:
<entry xmlns="http://www.w3.org/2005/Atom">
  <id>some ID</id>
  <title>Intermediary Touchpoint Instance</title>
  <author><name>author_name</name></author>
  <content/>
  <category term="intermediary-tp"
scheme="http://www.ibm.com/xmlns/prod/scmp#touchpoint-role" />
  <scmp:data xmlns:scmp="http://www.ibm.com/xmlns/prod/scmp" >
    <scmp:touchpoint>
      <scmp:propertySheet>
        <!--No parameters are required-->
      </scmp:propertySheet>
      <scmp:admin-state>enabled</scmp:admin-state>
    </scmp:touchpoint>
  </scmp:data>
</entry>
```

Der Touchpoint-Server gibt eine Antwort zurück, die etwa wie folgt lautet:

```
201 Created
Location: <Touchpoint Instance Entry URL>
Body:
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:ns2="http://a9.com/-/spec/opensearch/1.1/"
xmlns:ns3="http://www.w3.org/1999/xhtml">
  <id>generated_ID</id>
  <updated>2010-02-18T11:20:00.546+02:00</updated>
  <title>Intermediary Touchpoint Instance</title>
```

```

<link href="<Touchpoint Instance Entry URL>" type="application/atom+xml;type=entry" rel="self"/>
<link href="<Touchpoint Instance Entry URL>" type="application/atom+xml;type=entry" rel="edit"/>
<link href="<Touchpoint Type Entry URL>" type="application/atom+xml;type=entry"
rel="http://www.ibm.com/xmlns/prod/scmp#resource-type"/>
<category scheme="http://www.ibm.com/xmlns/prod/scmp#touchpoint-type" term="virtual://Intermediary"/>
<link href="<Touchpoint Status Entry URL>" type="application/atom+xml;type=entry"
rel="http://www.ibm.com/xmlns/prod/scmp#status"/>
<link href="<Touchpoint Destinations Feed URL>" type="application/atom+xml;type=feed"
rel="http://www.ibm.com/xmlns/prod/scmp#tp-destination"/>
<author><name>author_name</name></author>
<category scheme="http://www.ibm.com/xmlns/prod/scmp#touchpoint-role" term="intermediary-tp"/>
<category scheme="http://www.ibm.com/xmlns/prod/scmp#resource" term="touchpoint"/>
<ns2:data xmlns:ns2="http://www.ibm.com/xmlns/prod/scmp">
  <ns2:touchpoint>
    <ns2:admin-state>enabled</ns2:admin-state>
    <ns2:propertySheet/>
  </ns2:touchpoint>
</ns2:data>
<content/>
</entry>

```

Beachten Sie, dass der Touchpoint-Server die ID des Eintrags ändert, um ihre Eindeutigkeit zu gewährleisten.

Als Nächstes wird ein Ziel zur Touchpoint-Instanz mit der Rolle "Intermediary" hinzugefügt:

POST <Touchpoint Instance Destinations Feed>

```

Body:
<entry xmlns="http://www.w3.org/2005/Atom">
<scmp:data xmlns:scmp="http://www.ibm.com/xmlns/prod/scmp" >
  <scmp:destination>
    <scmp:request-out>Request-out URL</scmp:request-out>
  </scmp:destination>
</scmp:data>
</entry>

```

Der Touchpoint-Server sollte eine Antwort zurückgeben, die etwa wie folgt lautet:

```

201 Created
Location: <Touchpoint Instance Destination Entry URL>
Body:
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:ns2="http://a9.com/-/spec/opensearch/1.1/"
xmlns:ns3="http://www.w3.org/1999/xhtml">
  <id>generated_id</id>
  <updated>2010-02-18T10:52:35.108+02:00</updated>
  <link href="<Touchpoint Instance Destination Entry URL>" type="application/atom+xml;type=entry" rel="self"/>
  <link href="<Touchpoint Instance Destination Entry URL>" rel="edit"/>
  <category scheme="http://www.ibm.com/xmlns/prod/scmp#resource" term="tp-destination"/>
  <ns2:data xmlns:ns2="http://www.ibm.com/xmlns/prod/scmp">
    <ns2:destination>
      <ns2:request-out>Request-out URL</ns2:request-out>
    </ns2:destination>
  </ns2:data>

```

Abschließend wird die URL abgerufen, die zum Senden der Anforderung an die Touchpoint-Instanz mit der Rolle "Intermediary" verwendet wird. Wie bei einer Touchpoint-Instanz mit der Rolle "Provider" erfolgt dies über den Stauseintrag.

Senden Sie eine HTTP-Anforderung GET an die URL <Touchpoint Instance Status Entry URL>, die im Eintrag für die Touchpoint-Instanz verfügbar ist.

Der Touchpoint-Server gibt eine Antwort zurück, die etwa wie folgt lautet:

```

200 OK
Location: <Touchpoint Instance Status Entry URL>
Body:
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:ns2="http://a9.com/-/spec/opensearch/1.1/"
xmlns:ns3="http://www.w3.org/1999/xhtml">
  <id>generated_id</id>
  <link href="<Touchpoint Instance Status Entry URL>" type="application/atom+xml;type=entry" rel="self"/>
  <category scheme="http://www.ibm.com/xmlns/prod/scmp#resource" term="touchpoint"/>
  <category scheme="http://www.ibm.com/xmlns/prod/scmp#aspect" term="status"/>
  <ns2:data xmlns:ns2="http://www.ibm.com/xmlns/prod/scmp">
    <ns2:touchpoint-status>
      <ns2:request-in>Touchpoint Intermediary Request-in URL</ns2:request-in>
      <ns2:op-state>available</ns2:op-state>
    </ns2:touchpoint-status>
  </ns2:data>
</entry>

```

Kapitel 18. Tombstone Manager

Mit Tombstones können Sie den Exitstatus und andere Informationen abrufen. Nachstehend erhalten Sie Informationen zu dessen Funktionen.

IBM Security Directory Integrator kann verfolgen, ob Konfigurationen oder Fertigungslinien beendet wurden. Auf diese Weise können Sie die letzte Ausführung der Fertigungslinien feststellen, ohne hierzu die Protokolle der einzelnen Fertigungslinien aufrufen zu müssen.

Ermöglicht wird dies durch die Komponente *Tombstone Manager* von IBM Security Directory Integrator, die für jede Fertigungslinie und Konfiguration bei deren Beendigung *Tombstones* erstellt. Tombstones enthalten den Exitstatus und weitere Informationen, die Sie über die Server-API anfordern können. Außerdem führt Tombstone Manager Folgendes aus:

- Der Status einer vollständigen IBM Security Directory Integrator-Konfiguration wird in einem AMC-Statusfenster angezeigt.
- Wiederholte Ausführungen von Fertigungslinien in Action Manager (z. B. alle 24 Stunden) werden sichergestellt.
- Statusinformationen zu synchron ausgeführten Fertigungslinien werden für Clients der Server-API bereitgestellt.

Die Tombstone Manager-API ist in der Java-API-Dokumentation dokumentiert (suchen Sie dort die Informationen zur Klasse `com.ibm.di.api.Tombstone`).

Tombstones konfigurieren

Wählen Sie die erforderlichen Optionen aus, um die Tombstone-Erstellung zu konfigurieren. Sie können dazu auch die Liste der Switches in Ihrer Konfiguration verwenden.

Die Erstellung von Tombstones für Fertigungslinien und Konfigurationsinstanzen wird mit Markierungsfeldern in einer Reihe von Anzeigen des Konfigurationseditors sowie einer Reihe von Optionen in der Datei `global.properties` oder `solution.properties` konfiguriert.

Nachdem das Konfigurieren ausgeführt wurde, enthalten Ihre Konfigurationen die folgenden Switches:

Auf Konfigurationsebene:

- Switch "Konfiguration": Gibt an, ob für die eigentliche Konfigurationsinstanz Tombstones erstellt werden oder nicht.
- Switch "Alle Fertigungslinien": Gibt an, ob für alle Fertigungslinien aus dieser Konfiguration Tombstones erstellt werden.

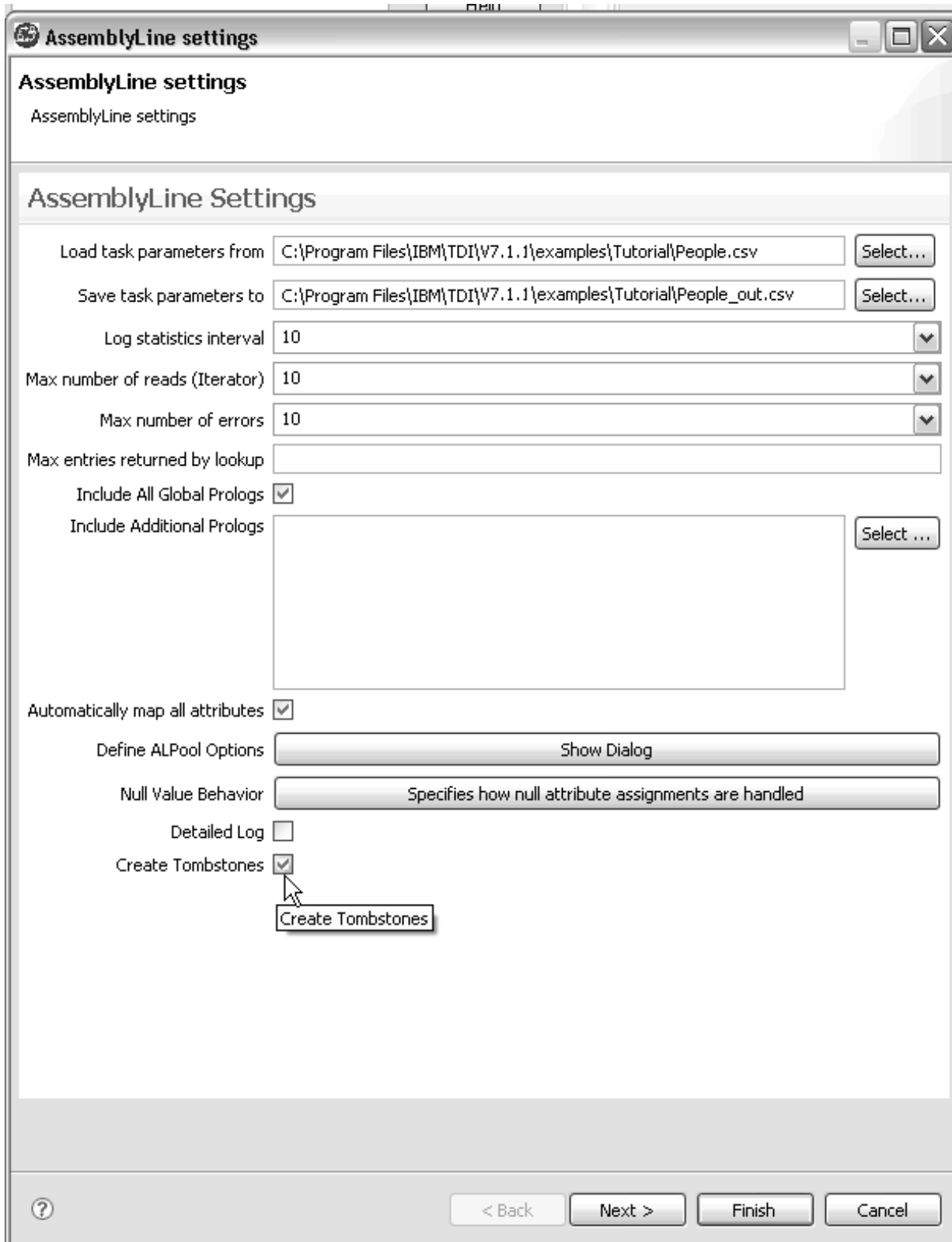
Auf Fertigungslinienebene:

Ein Switch, der angibt, ob für diese bestimmte Fertigungslinie Tombstones erstellt werden. Dieser Switch wird nur dann berücksichtigt, wenn der Switch "Alle Fertigungslinien" auf Konfigurationsebene inaktiviert ist.

Konfigurationsanzeige des Konfigurationseditors

Im angezeigten Konfigurationsfenster für Fertigungslinien können Sie die Tombstone-Erstellung für eine Fertigungslinie konfigurieren.

Die Option "Tombstones erstellen" befindet sich im unteren Bereich des Fensters.



Bei Auswahl des Markierungsfeldes **Tombstones erstellen** werden für diese Fertigungslinie bei ihrer Ausführung Tombstones generiert (sogar dann, wenn der Hauptschwitch für Fertigungslinien inaktiviert ist).

Konfigurationsanzeige für Fertigungslinien

Nachstehend erhalten Sie Informationen zum Tombstonedatensatz und zu der Liste der Attribute, mit denen der Tombstone aktiviert wird.

Wenn Tombstones für Fertigungslinien mit der zuvor dargestellten Konfigurationsoption inaktiviert sind, kann die Tombstonegenerierung dennoch für einzelne Fertigungslinien individuell aktiviert werden. Hierzu wird die entsprechende Option **Tombstones erstellen** in der Konfigurationsanzeige für die Fertigungslinie verwendet.

Ein Tombstonedatensatz könnte beispielsweise folgendermaßen aussehen:

Tabelle 34. Tombstonedatensatz

Feldname	Wert
Component Type ID (Komponententyp-ID)	1
Event Type ID (Ereignistyp)	0
StartTime (Startzeit)	11.11.2005 11:11:54
TombstoneCreateTime (Erstellungszeit für den Tombstone)	11.11.2005 17:22:45
Component Name (Komponentenname)	"ActiveDirectoryChangeLogSynchronizer"
Configuration	"C:\TDI_SOL_DIR\rs.xml"
Exit Code (Exit-Code)	0
Error Description (Fehlerbeschreibung)	""
GUID	"432640786324026346432"
Statistics (Statistik)	[get:571] [add:571] [err:0]

Die zurückgegebene Statistik kann eines oder mehrere der folgenden Attribute enthalten:

Tabelle 35. In Tombstonedatensatz zurückgegebene Statistik

Attribut	Beschreibung
add	Die Gesamtzahl der Einträge, die von der Fertigungslinie hinzugefügt wurden (ausgeführt durch Connectors im Modus "AddOnly").
mod	Die Gesamtzahl der Einträge, die von der Fertigungslinie modifiziert wurden (ausgeführt durch Connectors im Modus "Update").
del	Die Gesamtzahl der Einträge, die von der Fertigungslinie gelöscht wurden (ausgeführt durch Connectors im Modus "Delete").
get	Die Gesamtzahl der Einträge, die von der Fertigungslinie abgerufen wurden (ausgeführt durch Connectors im Modus "Iterator").
request	Die Gesamtzahl der akzeptierten Anforderungen, wenn die Fertigungslinie einen Connector im Modus "Server" enthält.
callReply	Die Gesamtzahl der CallReply-Operationen, die die Fertigungslinie durchgeführt hat (ausgeführt durch Connectors im Modus "CallReply").
err	Die Gesamtzahl der festgestellten Fehler.
skip	Die Gesamtzahl der Einträge, die von der Fertigungslinie übersprungen wurden.

Tabelle 35. In Tombstonedatensatz zurückgegebene Statistik (Forts.)

Attribut	Beschreibung
lookup	Die Gesamtzahl der Suchoperationen, die die Fertigungslinie durchgeführt hat (ausgeführt durch Connectors im Modus "Update", "Delete" oder "Lookup").
ignore	Die Gesamtzahl der Einträge, die von der Fertigungslinie ignoriert wurden (ausgeführt durch Connectors im Modus "Update" oder "Delta").
reconnect	Die Gesamtzahl der Versuche, die die Fertigungslinie unternommen hat, um eine Verbindung zu einem anderen Client herzustellen.
exception	Der Ausnahmetext, falls die Fertigungslinie mit einer Ausnahmebedingung beendet wurde.
getTries	Die Gesamtzahl der Versuche, die die Fertigungslinie unternommen hat, um einen Eintrag abzurufen (ausgeführt durch Connectors im Modus "Iterator").
getClientTries	Die Gesamtzahl der Versuche, die die Fertigungslinie unternommen hat, um den nächsten verbundenen Client zu erreichen (ausgeführt durch Connectors im Modus "Server").
nochange	Die Gesamtzahl der Einträge, die von der Fertigungslinie verarbeitet, jedoch nicht geändert wurden.
branchtrue	Die Gesamtzahl der Verzweigungskomponenten, die die Fertigungslinie ausgeführt hat, weil der entsprechende Ausdruck mit "true" (= wahr) ausgewertet wurde.
branchfalse	Die Gesamtzahl der Verzweigungskomponenten, die die Fertigungslinie übersprungen hat, weil der entsprechende Ausdruck mit "false" (= falsch) ausgewertet wurde.
loopstart	Die Gesamtzahl der Schleifenkomponenten, die durch die Fertigungslinie ausgeführt wurden.
loopcycles	Die Gesamtzahl der Zyklen, die für alle Schleifenkomponenten ausgeführt wurden, die über mehr als einen Zyklus in einer Fertigungslinie verfügten.
reconnectTime	Die Zeit in Millisekunden seit dem letzten Versuch der Fertigungslinie, die Verbindung wiederherzustellen.

Tombstone Manager

Sie können mit Tombstone Manager eine Reihe von Tasks ausführen. Das Tool verweist auf Werte von Konfigurationseigenschaften. Nachstehend erhalten Sie detaillierte Informationen dazu.

Tombstone Manager überwacht die Anzahl der Tombstonedatensätze zur Laufzeit und löscht alte Datensätze gemäß den Werten für die Konfigurationseigenschaften `com.ibm.di.tm.autodel.age`, `com.ibm.di.tm.autodel.records.trigger.on` und `com.ibm.di.tm.autodel.records.max`.

- Tombstone Manager protokolliert Stoppereignisse für Konfigurationsinstanzen und Fertigungslinien.
- Zur Registrierung für Ereignisbenachrichtigungen und zum Empfang von Stoppereignissen für Konfigurationsinstanzen und Fertigungslinien verwendet Tombstone Manager Aufrufe der lokalen Server-API.
- Für die Datenspeicherung verwendet Tombstone Manager den IBM Security Directory Integrator-Systemspeicher.
- Die Server-API-Schnittstellen (dokumentiert in der Java-API-Dokumentation) enthalten Aufrufe zum Abfragen von Tombstone Manager nach verschiedenen Daten, beispielsweise nach Tombstones für Fertigungslinien und Konfigurationsinstanzen.
- Tombstone Manager bietet Optionen für das Löschen von alten Tombstonedatensätzen.

Der Lebenszyklus eines Tombstones für eine Fertigungslinie könnte beispielsweise folgendermaßen aussehen:

- Tombstone Manager empfängt ein Server-API-Ereignis, dass eine Fertigungslinie beendet wurde (dies setzt voraus, dass die Server-API und Tombstone Manager aktiviert sind und dass in der Konfigurationsdatei die Erstellung von Tombstones für diese Fertigungslinie definiert ist).
- Tombstone Manager extrahiert die erforderlichen Daten aus dem Server-API-Ereignis und erstellt im Systemspeicher einen entsprechenden Tombstonedatensatz für die Datenbank.
- Solange der Tombstonedatensatz im Systemspeicher verfügbar ist, können Abfragen über Aufrufe der Server-API ausgeführt werden, die alle im Tombstonedatensatz enthaltenen Informationen bereitstellen.
- Der Tombstonedatensatz wird aus dem Systemspeicher gelöscht, wenn entweder ein expliziter Bereinigungsaufruf für die Server-API ausgeführt wird, der den Datensatz löscht, oder wenn der Datensatz durch die Logik für das automatische Löschen alter Tombstonedatensätze erfasst wird. Keines dieser Ereignisse muss zwangsläufig eintreten. Der Tombstonedatensatz kann daher theoretisch unbegrenzt fortbestehen.

Tombstone Manager

Die Tombstone Manager-Task wird durch Eigenschaften in der Datei `global.properties` oder `solution.properties` für Ihre Konfigurationsinstanz konfiguriert.

Anmerkung: Damit Tombstone Manager ordnungsgemäß arbeitet, muss die Server-API aktiviert sein (die Eigenschaft `api.on` muss also auf `true` gesetzt sein). In diesem Zusammenhang sind die folgenden Eigenschaften relevant:

com.ibm.di.tm.on

Dies ist der Master-Switch für Tombstone Manager. Gültige Werte sind **on** und **off**. Bei der Einstellung "off" werden selbst bei einer entsprechenden Festlegung in der Konfigurationsdatei Tombstones weder generiert noch verwaltet (auch eine Abfrage mit der Server-API oder AMC ist in diesem Fall nicht möglich).

Der Standardwert dieser Eigenschaft ist **false**.

com.ibm.di.tm.autodel.age

Diese Eigenschaft gibt die Lebensdauer eines Tombstones in Tagen an. Wenn sie vorhanden ist und einen ganzzahligen Wert größer als 0 enthält, löscht Tombstone Manager automatisch alle Tombstonedatensätze, die älter als die angegebene Anzahl von Tagen sind.

Die Logik für das Löschen von Tombstonedatensätzen wird beim Start des IBM Security Directory Integrator-Servers sowie bei einem IBM Security Directory Integrator-Server mit langer Ausführungsdauer ein Mal pro Tag ausgelöst.

Der Standardwert dieser Eigenschaft ist **0**.

com.ibm.di.tm.autodel.records.trigger.on

Diese Eigenschaft gibt die Gesamtzahl von Tombstonedatensätzen an, bei der die Logik für das Begrenzen der Anzahl von Tombstonedatensätzen auf eine bestimmte Anzahl ausgelöst wird.

Der Standardwert dieser Eigenschaft ist **10000**.

com.ibm.di.tm.autodel.records.max

Diese Eigenschaft gibt an, wie viele Tombstones beibehalten werden sollen, sobald der durch den vorherigen Parameter `com.ibm.di.tm.autodel.records.trigger.on` angegebene Auslöser überschritten wird.

Der Standardwert dieser Eigenschaft ist **5000**.

com.ibm.di.tm.create.all

Diese Eigenschaft fungiert als Überschreibungsswitch für die Werte, die in den Konfigurationsdateien angegeben sind. Ist sie auf **true** gesetzt, erstellt Tombstone Manager für jede Fertigungslinie und Konfigurationsinstanz ungeachtet der in den Konfigurationen angegebenen Werte Tombstones. Dies ist nützlich, um die Tombstone-Erstellung für Konfigurationen aus einer Version vor Version 6.1, die keine Tombstonewerte besitzen, ohne eine Modifizierung der Konfigurationen zu aktivieren.

Die Logik für die automatische Bereinigung, die durch die Eigenschaft `com.ibm.di.tm.autodel.age` festgelegt wird, ist von der Logik für die automatische Bereinigung, die durch die Eigenschaften `com.ibm.di.tm.autodel.records.trigger.on` und `com.ibm.di.tm.autodel.records.max` definiert wird, unabhängig.

Tombstone Manager verwendet das Protokollierungsframework von IBM Security Directory Integrator und protokolliert seine Nachrichten im Hauptprotokoll des IBM Security Directory Integrator-Servers.

Ein Abschnitt in der Datei `global.properties` oder `solution.properties` könnte beispielsweise wie folgt aussehen:

```
com.ibm.di.tm.on=true  
com.ibm.di.tm.autodel.age=90  
com.ibm.di.tm.autodel.records.trigger.on=50000  
com.ibm.di.tm.autodel.records.max=25000  
com.ibm.di.tm.create.all=false
```

Diese Gruppe von Konfigurationseigenschaften gibt Folgendes an: Tombstone Manager ist aktiviert und Tombstones, die älter als 90 Tage sind, werden automatisch gelöscht. Wenn die Tombstonegesamtzahl 50000 erreicht, werden außerdem die ältesten 25000 Tombstonedatensätze automatisch gelöscht.

Kapitel 19. Mehrere IBM Security Directory Integrator-Services

IBM Security Directory Integrator-Services können als unterschiedliche Services registriert werden. Nachstehend erhalten Sie Informationen dazu.

IBM Security Directory Integrator als Windows-Dienst

Sie können mit dem Windows-Dienst verschiedene Tasks ausführen. Nachstehend erhalten Sie detaillierte Informationen dazu.

In IBM Security Directory Integrator gibt es einen Mechanismus, der die Registrierung mehrerer IBM Security Directory Integrator-Serverinstanzen als Windows-Dienste ermöglicht. Für jede Instanz ist ein separates Lösungsverzeichnis erforderlich. Nach der Erstellung eines Lösungsverzeichnisses muss ein Dienstprogramm in das Verzeichnis kopiert werden. Dieses Programm heißt `ibmdiservice.exe`. Die Konfiguration des Dienstprogramms und des Windows-Dienstes erfolgt durch eine Eigenschaftendatei namens `ibmdiservice.props`. Jedes Lösungsverzeichnis muss eine Datei mit Konfigurationseigenschaften enthalten.

Für jeden Windows-Dienst muss ein separater Name vergeben werden. Mit einer Eigenschaft namens "servicename" wird in der Eigenschaftendatei ein Name angegeben, der bei der Erstellung des Namens und des Anzeigenamens für den Windows-Dienst verwendet wird. Der Name des Windows-Dienstes wird gebildet, indem dem Wert der Eigenschaft **servicename** das Präfix "ibmdisrv-" vorangestellt wird. Der Anzeigename für den Windows-Dienst wird gebildet, indem der Wert der Eigenschaft "servicename" zwischen den Klammern der Zeichenfolge "IBM Security Directory Integrator ()" eingefügt wird. Hat die Eigenschaft "servicename" beispielsweise den Wert "test", lautet der Name des Windows-Dienstes "ibmdisrv-test" und der Anzeigename für den Windows-Dienst "IBM Security Directory Integrator (test)". Wenn die Eigenschaft "servicename" nicht vorhanden ist oder keinen Wert enthält, werden Standardnamen verwendet. Die Standardnamen für den Namen und den Anzeigenamen des Windows-Dienstes lauten "ibmdisrv" und "IBM Security Directory Integrator".

Es gibt eine Eigenschaft, mit der konfiguriert werden kann, ob der Windows-Dienst beim Starten von Windows automatisch gestartet werden soll oder ob der Windows-Dienst manuell gestartet werden muss. Diese Eigenschaft heißt **autostart**; ihre gültigen Werte sind "true" und "false".

Anmerkung: Diese Eigenschaft wird bei der Installation und Deinstallation sowie bei der Ausführung des Dienstes verwendet. Aus diesem Grund darf der Eigenschaftswert nach der Installation des Windows-Dienstes nicht geändert werden.

Weitere Informationen zur Eigenschaftendatei für die Konfiguration von IBM Security Directory Integrator als Windows-Dienst enthält der Abschnitt „Dienst konfigurieren“ auf Seite 377.

Dienst installieren und deinstallieren

Führen Sie die hier aufgeführten Schritte aus, um den Service zu installieren und zu deinstallieren.

Dienst installieren

Mit den hier aufgeführten Schritten können Sie den Dienst installieren.

Informationen zu diesem Vorgang

So installieren Sie den IBM Security Directory Integrator-Dienst:

Vorgehensweise

1. Vergewissern Sie sich, dass IBM Security Directory Integrator installiert ist. Der Installationsordner von IBM Security Directory Integrator wird als *stammverzeichnis* bezeichnet. Siehe Installationsprogramm für Windows-Plattformen.
2. Wählen Sie einen Lösungsordner aus, der von IBM Security Directory Integrator verwendet wird, wenn TDI als Windows-Dienst gestartet wird. Diesen Ordner können Sie frei wählen. Sobald IBM Security Directory Integrator als Dienst installiert wurde, kann der von Ihnen verwendete Lösungsordner nicht mehr geändert werden, bis eine Deinstallation als Dienst vorgenommen wird. Bitte beachten Sie, dass die Auswahl des Lösungsordners für den Windows-Dienst die Ausführung von IBM Security Directory Integrator mit einem beliebigen anderen Lösungsordner nicht verhindert.
3. Nach Auswahl des Lösungsordners kopieren Sie in diesen Ordner alle Dateien aus dem Ordner *stammverzeichnis/win32_service* (dies sind die Dateien "ibmdiservice.exe", "ibmdiservice.props" und "Log4J.properties").
4. Führen Sie den folgenden Befehl ausgehend von dem Lösungsordner aus, den Sie für den Windows-Dienst ausgewählt haben: `ibmdiservice.exe -i`

Dienst deinstallieren

Mit den hier aufgeführten Schritten können Sie den Dienst deinstallieren.

Informationen zu diesem Vorgang

Anmerkung: Damit die Version des Dienstprogramms "ibmdiservice.exe" von IBM Security Directory Integrator Version 7.2 verwendet werden kann, müssen alle zuvor registrierten Windows-Dienste aus älteren Versionen als IBM Security Directory Integrator 7.2 deinstalliert werden und anschließend muss der Windows-Dienst von IBM Security Directory Integrator 7.2 installiert werden. Dies ist notwendig, weil der Windows-Dienst von IBM Security Directory Integrator 7.2 im Vergleich zu den Vorversionen mit "ibmdisrv" einen anderen Standardnamen für den Namen des Windows-Dienstes verwendet. Bei den älteren Versionen lautete der Standardname "IBM Security Directory Integrator".

So deinstallieren Sie den IBM Security Directory Integrator-Dienst:

1. Stellen Sie sicher, dass der IBM Security Directory Integrator-Dienst gestoppt wurde.
2. Führen Sie den folgenden Befehl ausgehend von dem Lösungsordner aus, in dem Sie den Dienst installiert haben:

```
ibmdiservice.exe -u
```

Anmerkung:

1. Durch die Deinstallation des IBM Security Directory Integrator-Dienstes wird IBM Security Directory Integrator selbst nicht deinstalliert. IBM Security Directory Integrator kann weiterhin verwendet werden, ist jedoch nicht mehr als Windows-Dienst registriert und kann nicht als solcher ausgeführt werden. Sie können den IBM Security Directory Integrator-Dienst zu einem späteren Zeitpunkt erneut installieren.

2. Falls der IBM Security Directory Integrator-Dienst installiert ist und Sie IBM Security Directory Integrator vollständig (also nicht nur den Dienst) deinstallieren wollen, gehen Sie folgendermaßen vor:
 - a. Deinstallieren Sie den Windows-Dienst.
 - b. Deinstallieren Sie IBM Security Directory Integrator (siehe Deinstallationsprogramm für Windows-Plattformen).

Dienst starten und stoppen

Starten und stoppen Sie den Dienst mit den aufgeführten Optionen.

Der IBM Security Directory Integrator-Dienst startet IBM Security Directory Integrator automatisch beim Systemboot. Im Anschluss an die Installation des Dienstes wird IBM Security Directory Integrator jedoch nicht automatisch gestartet. Nach der Installation des Dienstes können Sie den Dienst mit einem der drei folgenden Verfahren starten:

- Starten Sie den Computer erneut.
- Starten Sie den IBM Security Directory Integrator-Dienst über das Windows-Fenster "Dienste".
- Verwenden Sie die Befehlszeile. Informationen hierzu finden Sie unter „Befehlszeilenunterstützung“ auf Seite 382.

Dienst manuell starten und stoppen

Sie können den IBM Security Directory Integrator-Dienst über das Windows-Fenster "Dienste" manuell starten und stoppen.

Hierzu müssen Sie im Fenster **Dienste** den IBM Security Directory Integrator-Dienst auswählen und je nach Windows-Version entweder auf die Schaltfläche **Starten/Stoppen** klicken oder mit der rechten Maustaste auf den Namen des Dienstes klicken und die Option **Starten/Stoppen** auswählen.

Die Verwendung der Befehlszeile ist ebenfalls möglich (siehe „Befehlszeilenunterstützung“ auf Seite 382).

Starttyp für Dienst ändern

Der IBM Security Directory Integrator-Dienst ist standardmäßig so konfiguriert, dass er beim Systemboot gestartet wird.

Sie können den Startmodus für den Dienst manuell im Windows-Fenster "Dienste" in die Einstellung **Manuell** oder **Deaktiviert** ändern.

Protokollierung

Der IBM Security Directory Integrator-Dienst protokolliert alle Nachrichten (**Fehlernachrichten**, **Informationen** und **Debugnachrichten**) im Windows-Systemprotokoll der Anwendung. Zum Anzeigen dieser Nachrichten verwenden Sie die Windows-Ereignisanzeige.

Dienst konfigurieren

Sie können die Eigenschaften in der Datei `ibmdiservice.props` angeben, um den IBM Security Directory Integrator-Service zu konfigurieren.

Der IBM Security Directory Integrator-Dienst wird mithilfe der Datei `ibmdiservice.props` konfiguriert, die während der Installation des Protokollservices im ausgewählten Lösungsverzeichnis abgelegt wird.

Anmerkung: Stellen Sie vor der Ausführung des Dienstes sicher, dass diese Datei wie im vorliegenden Abschnitt beschrieben ordnungsgemäß konfiguriert wurde. Der Dienst könnte fehlschlagen, falls die Datei falsche Werte enthält. Die folgenden Eigenschaften werden in der Datei `ibmdiservice.props` angegeben:

path Gibt die Umgebungsvariable `PATH` an, die zum Ausführen des IBM Security Directory Integrator-Prozesses verwendet wird (diese Eigenschaft ist normalerweise mit der Variablen `PATH` aus "`ibmdisrv.bat`" identisch, kann jedoch von Ihnen geändert werden). Diese Eigenschaft ist optional.

ibmdirroot

Gibt den Stammordner von IBM Security Directory Integrator an (z. B. `C:\Program Files\IBM\TDI\V7.2`). Diese Eigenschaft ist erforderlich.

configfile

Gibt den Dateipfad für die Konfigurationsdatei von IBM Security Directory Integrator an. Diese Eigenschaft ist optional.

assemblylines

Gibt als durch Kommas getrennte Liste die Fertigungslinien an, die beim Starten des IBM Security Directory Integrator-Dienstes automatisch gestartet werden. Diese Eigenschaft ist optional.

cmdoptions

Gibt weitere Optionen an, die beim Dienststart direkt an IBM Security Directory Integrator übergeben werden (eine vollständige Liste der IBM Security Directory Integrator-Optionen finden Sie unter Kapitel 13, „Befehlszeilenoptionen“, auf Seite 233).

Eine dieser Optionen ist die Option `-c`. Mit ihr können Sie mehrere (durch Kommas getrennte) Konfigurationsdateien angeben, was beim Parameter **configfile** nicht möglich ist.

Für die Verwendung dieser Konfiguration gelten die folgenden Anforderungen:

- Der vollständige Pfad, entweder in der Windows-Syntax (`\\`) oder in der UNIX-Syntax (`/`), ist für jede Fertigungslinie erforderlich.
- Die Namen der Konfigurationsdateien müssen in einer Gruppe von Anführungszeichen durch Komma getrennt enthalten sein.
- Die Option `-d` ist erforderlich.
- Die Dateinamen dürfen keine Leerzeichen enthalten.

Beispiel:

```
cmdoptions=-c"C:/TDI7.1-Solutions/myConfig/Config1.xml" , C:/TDI7.1-Solutions/AnotherConfig/TechNotes.xml" -d
```

Diese Eigenschaft ist optional.

servicename

Gibt einen Namen an, aus dem der Name und der Anzeigename für den Windows-Dienst gebildet werden. Als Namen für den Windows-Dienst wird der Wert der Eigenschaft **servicename** mit dem Präfix "`ibmdisrv-`" verwendet. Der Anzeigename für den Windows-Dienst wird erstellt, indem der Wert der Eigenschaft **servicename** zwischen den Klammern im Ausdruck "`IBM Security Directory Integrator ()`" eingefügt wird.

Ist der Eigenschaftswert beispielsweise "test", lautet der Name des Windows-Dienstes "ibmdisrv-test" und der Anzeigename für den Windows-Dienst "IBM Security Directory Integrator (test)". Falls die Eigenschaft **servicename** nicht vorhanden ist oder keinen Wert enthält, werden Standardnamen verwendet. Der Standardname für den Windows-Dienst ist "ibmdisrv" und der Standardanzeigename für den Windows-Dienst lautet "IBM Security Directory Integrator".

Anmerkung: Diese Eigenschaft wird bei der Installation und Deinstallation sowie bei der Ausführung des Dienstes verwendet. Aus diesem Grund darf der Eigenschaftswert nach der Installation des Windows-Dienstes nicht geändert werden.

autostart

Gibt an, ob der Windows-Dienst beim Starten von Windows automatisch gestartet wird oder ob er manuell gestartet werden muss. Die gültigen Werte für diese Eigenschaft sind **true** und **false**. Der Wert **true** gibt an, dass der Windows-Dienst beim Windows-Start gestartet wird. Der Wert **false** gibt an, dass der Dienst manuell gestartet werden muss. Falls diese Eigenschaft nicht vorhanden ist oder keinen Wert enthält, wird der Standardwert **true** verwendet.

Diese Eigenschaft wird bei der Installation des Windows-Dienstes verwendet. Nachdem der Windows-Dienst installiert wurde, hat eine etwaige Änderung dieser Eigenschaft keine Auswirkung.

controlledshutdown

Gibt an, ob der Windows-Dienst den Server ordnungsgemäß beendet oder aber den Serverprozess abbricht. Die gültigen Werte für diese Eigenschaft sind "true" und "false". Der Wert "true" gibt an, dass der Windows-Dienst den IBM Security Directory Integrator-Server ordnungsgemäß beendet. Der Wert "false" gibt an, dass der Serverprozess abgebrochen wird. Falls diese Eigenschaft nicht vorhanden ist oder keinen Wert enthält, wird der Standardwert "false" verwendet.

debug Gibt mit dem Wert **true** oder **false** an, ob Debuginformationen aktiviert bzw. inaktiviert werden. Bei einer Aktivierung der Debuginformationen wird für ausführliche Tracenachrichten ein Speicherauszug im Windows-Systemprotokoll für die Anwendung erstellt. Diese Eigenschaft ist optional.

Anmerkung: Bei der Angabe von Eigenschaften in der Konfigurationsdatei müssen Sie jede Eigenschaft in einer separaten Zeile angeben und das folgende Format verwenden:

```
<eigenschaftsname>=<eigenschaftswert>
```

Vor und nach dem Gleichheitszeichen (=) darf kein Leerzeichen angegeben werden.

Beispiel für eine vollständige Datei `ibmdiservice.props`:

```
path=C:\Program Files\IBM\TDI\V7.2\jvm\jre\bin;  
C:\Program Files\IBM\TDI\V7.2\libs;  
ibmdiroot=C:\Program Files\IBM\TDI\V7.2  
configfile=rs.xml  
assemblylines=AssemblyLine1,AssemblyLine2  
cmdoptions=  
debug=false  
controlledshutdown=false
```

Anmerkung: Falls Sie eine der Eigenschaften in der Datei `ibmdiservice.props` ändern, müssen Sie den Dienst erneut starten, damit die Änderungen wirksam werden.

IBM Security Directory Integrator als Linux/UNIX-Dienst

Sie können mit dem Linux/UNIX-Dienst verschiedene Tasks ausführen. Nachstehend erhalten Sie detaillierte Informationen dazu.

Implementierungsmethoden

Bei Linux- und UNIX-Plattformen gibt es zwei unterschiedliche Möglichkeiten, um sicherzustellen, dass bestimmte Systemjobs oder "Dämonen" bei der Systeminitialisierung und Systembeendigung gestartet bzw. gestoppt werden.

1. Sie können ein Script in */etc/init.d* verwenden, das die Logik enthält, mit der die gewünschten Dämonen gestartet und gestoppt werden. Dieses Script stellt dann (feste) Verknüpfungen mit Scripts in */etc/rc3.d* her. Die Namen dieser Scripts beginnen mit *SXX...* und *KXX...*, wobei *XX* ein Numeral ist, das eine korrekte Abfolge der Dateien im Verzeichnis */etc/rc3.d* bewirkt. Die mit dem Buchstaben "S" beginnenden Scripts werden aufgerufen, wenn das System beim Systemstart die Ausführungsphase 3 erreicht. Die mit dem Buchstaben "K" beginnenden Scripts werden bei der Beendigung des Systems aufgerufen.
2. Sie können die Datei */etc/inittab* bearbeiten.

Der zweite Prozess ist nachfolgend beschrieben. Einige der vorgestellten Informationen können auch zur Erstellung von Scripts für den Einsatz der ersten Implementierungsmethode verwendet werden.

Datei *"/etc/inittab"* anpassen

Damit IBM Security Directory Integrator-Dämonprozesse beim Starten des UNIX-/Linux-Betriebssystems gestartet werden, müssen in die Datei */etc/inittab* entsprechende Einträge aufgenommen werden. Der Registrierung von IBM Security Directory Integrator als Windows-Dienst unter Windows entspricht das Hinzufügen einer Textzeile zur Datei */etc/inittab* unter UNIX/Linux. Der Deinstallation des IBM Security Directory Integrator-Windows-Dienstes unter Windows entspricht das Entfernen der entsprechenden Einträge aus der Datei */etc/inittab*. Für jeden IBM Security Directory Integrator-Dämonprozess, der beim Systemstart gestartet werden muss, muss 1 Textzeile zur Datei */etc/inittab* hinzugefügt werden. Das Format und die Bedeutung der Einträge in dieser Datei sind nachfolgend beschrieben. Jeder Eintrag in der Datei */etc/inittab* hat das folgende Format:

kennung:ausführungsebene:aktion:befehl

Beschreibung der einzelnen Felder:

- Das Feld **kennung** ist eine Zeichenfolge (Mindestlänge: 1 Zeichen), die ein Objekt eindeutig kennzeichnet. Diese Zeichenfolge wird zur eindeutigen Kennzeichnung des entsprechenden Befehls verwendet.
- Das Feld **ausführungsebene** ist die Ausführungsebene, in der dieser Eintrag verarbeitet werden kann. Ausführungsebenen entsprechen quasi einer Konfiguration von Prozessen im System. Jedem durch einen Befehl "init" gestarteten Prozess werden eine oder mehrere Ausführungsebenen zugeordnet, auf denen er ausgeführt werden kann. Eine Ausführungsebene wird durch die Zahlen 0 bis N angegeben. Hierbei ist N eine positive ganze Zahl, die bei den verschiedenen UNIX-/Linux-Betriebssystemen unterschiedlich ist (bei einigen AIX-Computern steht N beispielsweise für 9, bei RedHat Linux steht N für 6 usw.). Falls beispielsweise das Betriebssystem auf der Ausführungsebene 3 ausgeführt wird, werden nur die für die Ausführungsebene angegebenen Prozesse gestartet.

Im Feld **ausführungsebene** können mehrere Ausführungsebenen für einen Prozess definiert werden. Hierzu werden mehrere Ausführungsebenen in einer beliebigen Kombination von 0 bis N ausgewählt. Falls IBM Security Directory Integrator beispielsweise auf den Ausführungsebenen 3 und 6 ausgeführt werden muss, muss die Ausführungsebene mit dem Wert "36" angegeben werden. Falls keine Ausführungsebene angegeben ist, wird davon ausgegangen, dass der Prozess bei allen Ausführungsebenen gültig ist.

Es empfiehlt sich, nur dann Ausführungsebenen anzugeben, wenn die jeweilige IBM Security Directory Integrator-Lösung dies speziell erfordert.

- Das Feld **aktion** ist ein Wert aus einer Gruppe vordefinierter Aktionen, der den Befehl **init** anweist, wie der im Feld **befehl** angegebene Prozess zu verarbeiten ist. Es gibt viele Aktionen, die vom Befehl "init" erkannt werden. Zur Ausführung des IBM Security Directory Integrator-Servers als Dämonprozess empfiehlt sich jedoch die Verwendung der Aktion **once**. Die Aktion **once** hat die folgende Semantik:

Sobald der Befehl "init" eine Ausführungsebene erreicht, die mit der Ausführungsebene des Eintrags übereinstimmt, wird der Prozess gestartet und seine Beendigung wird nicht abgewartet. Nach der vollständigen Beendigung des Prozesses wird dieser nicht erneut gestartet. Wenn das System eine neue Ausführungsebene erreicht und der Prozess bedingt durch eine vorherige Änderung der Ausführungsebene noch ausgeführt wird, wird das Programm nicht erneut gestartet. Bei allen nachfolgenden Lesevorgängen für die Datei `/etc/inittab` ignoriert der Befehl "init" diesen Eintrag, solange er sich auf derselben Ausführungsebene befindet.

- Das Feld **befehl** gibt den auszuführenden Shellbefehl an.

Die folgenden drei Beispielinträge in der Datei `/etc/inittab` beziehen sich auf IBM Security Directory Integrator:

```
tdi1::once:/opt/IBM/TDI711_1/ibmdisrv -c "/opt/IBM/TDI711_1/myconfigs/rs1.xml" -r "testAL1"
tdi2::once:/opt/IBM/TDI711_2/ibmdisrv -c "/opt/IBM/TDI711_2/myconfigs/rs2.xml" -r "testAL2"
tdi3::once:/opt/IBM/TDI711_3/ibmdisrv -c "/opt/IBM/TDI711_3/myconfigs/rs3.xml" -r "testAL3"
```

Dieses Beispiel startet drei IBM Security Directory Integrator-Serverinstanzen, die jeweils in unterschiedlichen Ordnern installiert sind.

Anmerkung: Hinsichtlich des Systemstarts bestehen bei den verschiedenen UNIX-/Linux-Betriebssystemen einige Unterschiede. Aus diesem Grund werden in den hier bereitgestellten Informationen die Hauptaspekte des Startens von IBM Security Directory Integrator auf einem UNIX-/Linux-System abgedeckt, spezielle UNIX-/Linux-Systeme jedoch nicht behandelt.

Als Beispiel für die Datei `/etc/inittab` finden Sie ausführliche Informationen zur Konfigurationsdatei `/etc/inittab` für ein AIX-System unter der Adresse <http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.files/doc/aixfiles/inittab.htm>.

Ordnungsgemäße Beendigung

Bei UNIX-Systemen wird der Service immer ordnungsgemäß beendet. Dies wird durch einen Beendigungshook erreicht, der zur Java-Laufzeit hinzugefügt wird. Wenn der Server mit einem der Signale SIGINT oder SIGTERM gestoppt wird, wird hierdurch dieser Hook ausgeführt und der Server ordnungsgemäß beendet.

Dieses Verfahren hat den weiteren Vorteil, dass dieser Hook auf allen Plattformen aufgerufen wird, wenn der Server durch das Drücken der Tastenkombination Strg+C in seinem Konsolfenster gestoppt wird.

Anmerkung: Sie können auch ein externes Programm angeben, aus dem JVM-Beendigungshook heraus gestartet werden soll. Dieses externe Programm wird in der Datei `global.properties` oder `solution.properties` mit der optionalen Eigenschaft `jvm.shutdown.hook` angegeben. Wenn es konfiguriert ist, wird das externe Programm unmittelbar nach der ordnungsgemäßen Beendigung des Servers gestartet.

Befehlszeilenunterstützung

Nachstehend erhalten Sie Informationen zum Verwenden des Scripts, mit dem der Service gestartet und gestoppt werden kann.

IBM Security Directory Integrator bietet ein Script, mit dem der IBM Security Directory Integrator-Service (Dienst) auf Windows- und UNIX-Systemen gestartet und gestoppt werden kann. Das Script befindet sich im Verzeichnis *tdi-installationsverzeichnis/bin*; es heißt "servicemgr.bat" (.sh).

Das Script wird folgendermaßen verwendet:

```
servicemgr servicename_bzw._dienstname start|stop
```

Hierbei gilt Folgendes:

servicename_bzw._dienstname

Steht für den Namen des Service bzw. Dienstes. Bei Windows-Systemen ist dies standardmäßig `ibmdisrv` oder der Wert der Eigenschaft `servicename` in der Datei `ibmdiservice.props`. Bei UNIX-Systemen ist dies der Wert im Feld "kennung" aus der Datei `/etc/inittab`.

start|stop

Der gewünschte Befehl, der für den Service bzw. Dienst ausgeführt werden soll.

Anhang A. Beispiele für Eigenschaftendateien

Die können die Eigenschaftendateien für die Installation von IBM SDI anpassen. Nachstehend erhalten Sie Informationen zu den verfügbaren Textdateien und dem Lösungsverzeichnis.

Eine Installation von IBM Security Directory Integrator wird in großem Maß durch eine Reihe von Textdateien angepasst, die eine oder mehrere **Eigenschaften** enthalten. Deren Format besteht aus einem Schlüsselwort oder einer Kennung, auf das/ die ein Wert folgt. Die folgenden Textdateien mit globalen Eigenschaften befinden sich auf der Ebene `root/etc` des IBM Security Directory Integrator-Installationsverzeichnisses:

- „Log4J.properties“ auf Seite 384
- „jlog.properties“ auf Seite 385
- „derby.properties“ auf Seite 387
- „global.properties“ auf Seite 387

Die Eigenschaften, die in diesen Dateien festgelegt sind, bilden für alle Benutzer auf diesem Computer eine Referenzkonfiguration für die gesamte IBM Security Directory Integrator-Installation. Falls Sie jedoch ein Lösungsverzeichnis verwenden, bei dem es sich nicht um das Installationsverzeichnis handelt, können Sie in Ihrem Lösungsverzeichnis eine Gruppe von Textdateien verwenden, die ihre Entsprechungen im Installationsverzeichnis spiegeln. Eine Eigenschaft, die in einer dieser Dateien aufgeführt ist, setzt die Einstellung in einer der zuvor aufgeführten globalen Eigenschaftendateien der Installation außer Kraft. Darüber hinaus hat eine Java-Eigenschaft, die in einer Konfigurationsdatei festgelegt ist, die höchste Vorrangstellung und setzt alle Angaben in einer globalen Eigenschaftendatei oder in den Eigenschaftendateien im Lösungsverzeichnis außer Kraft.

Zur Angabe des Lösungsverzeichnisses haben Sie mehrere Möglichkeiten:

- Sie können die Umgebungsvariable `TDI_SOLDIR` vor dem Starten des Konfigurationseditors oder des Servers festlegen.
- Sie können den Parameter `-s` für das Script `ibmditk` zum Starten des Servers angeben. Dieses Verfahren hat Vorrang vor der Angabe der Umgebungsvariablen `TDI_SOLDIR`.

Falls die Umgebungsvariable `TDI_SOLDIR` mit dem Installationsverzeichnis identisch ist, entspricht das Verhalten den älteren Versionen von IBM Security Directory Integrator: Alle Eigenschaftendateien werden aus dieser Position gelesen und die Anmerkungen zu Eigenschaftendateien im Lösungsverzeichnis gelten nicht.

In allen anderen Fällen kopiert der IBM Security Directory Integrator-Server bei seiner erstmaligen Ausführung alle Eigenschaftendateien in Ihr Lösungsverzeichnis (gegebenenfalls bereits vorhandene Dateien werden hierbei nicht überschrieben). Anschließend können Sie diese Dateien ganz nach Bedarf und ohne Einfluss auf die Eigenschaftendateien im Installationsverzeichnis anpassen. Die im Lösungsverzeichnis verbliebenen Dateien bilden weiterhin eine Referenzkonfiguration für andere IBM Security Directory Integrator-Instanzen.

Anmerkung: Die Datei `global.properties` wird in Ihrem Lösungsverzeichnis in eine Datei namens `solutions.properties` kopiert. Andere Dateien (z. B. `Log4J.properties` sowie die Dateien in den Ordnern `amc` und `serverapi`) werden unter ihrem eigenen Namen kopiert.

Falls Ihr Lösungsverzeichnis während der Produktinstallation mit dem IBM Security Directory Integrator-Installationsprogramm eingerichtet wurde, enthält die Installation zusätzlich eine funktionsfähige Konfiguration für eine Systemwarteschlange. Wurde das Lösungsverzeichnis anderweitig erstellt (entweder manuell oder durch den Server mit der Option "-s"), müssen Sie entweder die Systemwarteschlange in Ihrer Datei `solution.properties` inaktivieren oder selbst eine Systemwarteschlange einrichten (siehe hierzu „Konfiguration der Systemwarteschlange“ auf Seite 177).

Log4J.properties

Diese Datei legt eine Referenzkonfiguration für die Protokollstrategie des Servers (`ibmdisrv`) fest.

Protokolloptionen, die auf der Registerkarte "Protokollierung" im Konfigurationseditor konfiguriert werden, werden in die Konfigurationsdatei geschrieben und ergänzen oder ersetzen die folgenden Angaben:

```
# This file controls the logging strategy for the server (ibmdisrv) when started
# from the command line.
# Look at executetask.properties for the logging strategy of the server when started
# from the Configuration Editor (ibmditk).
# Look at ce-log4j.properties for the logging behavior of the Configuration Editor (ibmditk).
#
# You will normally configure the logging strategy of the server by adding appenders
# using the Configuration Editor (ibmditk). This file only defines the baseline
# that is independent of the configuration files you are using.
#
# See the IDI documentation for more information on the contents of this file.
#

log4j.rootCategory=INFO, Default

# This is the default logger, you will see that it logs to ibmdi.log
log4j.appender.Default=org.apache.log4j.FileAppender
log4j.appender.Default.file=logs/ibmdi.log
log4j.appender.Default.layout=org.apache.log4j.PatternLayout
log4j.appender.Default.layout.ConversionPattern=%d{ISO8601} %-5p [%c] - %m%n
log4j.appender.Default.append=false

#Example settings for changing the default logger

#####ROLLING FILE SIZE APPENDER
##RollingFileAppender rolls over log files when they reach a certain size specified by the
##MaxFileSize parameter

#log4j.appender.Default=org.apache.log4j.RollingFileAppender
#log4j.appender.Default.File=logs/ibmdi.log
#log4j.appender.Default.Append=true
#log4j.appender.Default.MaxFileSize=10MB
#log4j.appender.Default.MaxBackupIndex=10
#log4j.appender.Default.layout=org.apache.log4j.PatternLayout
#log4j.appender.Default.layout.ConversionPattern=%d{ISO8601} %-5p [%c] - %m%n

#####DAILY OUTPUT LOG4J SETTINGS
## With the DailyRollingFileAppender the underlying file is rolled over at a user chosen frequency.
##The rolling schedule is specified by the DatePattern option

#log4j.appender.Default=org.apache.log4j.DailyRollingFileAppender
#log4j.appender.Default.file=logs/ibmdi.log
#log4j.appender.Default.DatePattern='.'yyyy-MM-dd
#log4j.appender.Default.layout=org.apache.log4j.PatternLayout
#log4j.appender.Default.layout.ConversionPattern=%d{ISO8601} %-5p [%c] - %m%n

# You may change the logging category of these subsystems to DEBUG
# if you want to investigate particular problems. This may
# generate a lot of output.
# ...com.ibm.di.config describes the loading of the configuration file (.xml),
# and how the internal configuration structure is built.
# ...com.ibm.di.loader gives information about jar files, and where classes are found.
# It also loads idi.inf files, which provides Connectors/Parsers/EH information
# for the Configuration Editor.
```



```

log4j.logger.com.ibm.di.config=WARN
log4j.logger.com.ibm.di.loader=WARN

# Uncomment the lines below to activate them

# Here is an example on how to make a logger that logs to the console
#log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
#log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
#log4j.appender.CONSOLE.layout.ConversionPattern=%d [%t] %-5p - %m%n

# Here is an example that logs to myFile.log
#log4j.appender.fileLOG=org.apache.log4j.FileAppender
#log4j.appender.fileLOG.file=myFILE.log
#log4j.appender.fileLOG.layout=org.apache.log4j.PatternLayout
#log4j.appender.fileLOG.layout.ConversionPattern=%d{ISO8601} %-5p [%c] - %m%n
#log4j.appender.fileLOG.append=false

# Finally, make use of the loggers defined above:
# Tell AssemblyLines myAL to log using CONSOLE logger defined above.

# log4j.logger.AssemblyLine.AssemblyLines/myAL=INFO, CONSOLE

# Or you could log to myFile.log

# log4j.logger.AssemblyLine.AssemblyLines/myAL=INFO, fileLOG

```

jlog.properties

Sie können die JLOG-basierten Werte des IBM SDI-Servers konfigurieren und ändern.

Diese Datei konfiguriert für den IBM Security Directory Integrator-Server die JLOG-basierte Traceerstellung und First-Failure Data Capture (FFDC, Erfassung von Fehlerdaten beim ersten Auftreten), die in Kapitel 15, „Traceerstellung und First-Failure Data Capture“, auf Seite 263 erläutert ist. Diese Werte können (während der Ausführung des Servers) unter Verwendung des Scripts "LogCmd" dynamisch modifiziert werden, falls die Eigenschaft `jlog.noLogCmd` beim Starten des Servers auf `false` gesetzt war.

Anmerkung: Normalerweise wird Log4J verwendet, um einen Trace des Ausführungsablaufs in Ihrer Lösung zu erstellen. Die JLOG-basierte Traceerstellung und First-Failure Data Capture ist dazu gedacht, der IBM Unterstützungsfunktion bei ihrer Arbeit zu helfen, falls Sie im Zusammenhang mit IBM Security Directory Integrator Probleme feststellen.

```

#####
# This file controls the tracing and First Failure Data Capture (FFDC) strategy for ITDI 7.2
# See the IDI documentation for more information on the contents of this file.
#####

#-----
# Enable the JLOG's command server
#
# If the jlog.noLogCmd is set to false, then the JLOG LogManager will listen on the
# default port (9992) for JLOG log commands.
# Setting this property to false will enable you to modify the JLOG properties dynamically using the
# logcmd scripts. The logcmd scripts are placed under ITDI_HOME directory.
# The default value is set to true.
#-----
jlog.noLogCmd=true

#-----
# Set listen port for JLOG's command server
#
# If you want LogManager to listen on different port than the default one (9992) you should
# uncomment the property jlog.logCmdPort and set it to the desired port. If not uncommented
# the LogManager will listen on the default port - 9992.
#-----
#jlog.logCmdPort=9992

#-----
# Configure Jlog FileHandler for tracing into a file.
#
# By default the FileHandler is not attached to the Jlog Logger.
# Uncomment the properties with the prefix jlog.filehandler below to configure a FileHandler.
# After uncommenting this you need to add the filehandler to the logger's listeners names as shown
# below
# e.g: jlog.logger.listenerNames=jlog.snapmemory jlog.snaphandler jlog.filehandler

```

```

#-----
jlog.filehandler.className=com.ibm.log.FileHandler
#jlog.filehandler.description=JLOG File Handler for Logging and Tracing
#jlog.filehandler.encoding=UTF8
#jlog.filehandler.maxFiles=10
#jlog.filehandler.maxFileSize=2048
#jlog.filehandler.appending=true
#jlog.filehandler.fileDir=logs/
#jlog.filehandler.trace.fileName=trace.log
#-----

#-----
# create a level filter.
# The level filter is used to define the level at which JFFDC action will be triggered.
# For JFFDC to be meaningful this should be set to either FATAL or ERROR (case-insensitive).
# NOTE: Setting the trigger level to other levels such as DEBUG_MIN will trigger unwanted JFFDC
# action causing a performance drop.
#-----
jlog.levelflt.className=com.ibm.log.LevelFilter
jlog.levelflt.level=FATAL

#-----
# Configure the SnapMemoryHandler for tracing into a memory buffer.
# The SnapMemoryHandler traces into a memory buffer and dumps the contents of the memory to a file on
# trigger of a event (as defined by the level filter above) and writes the content to the specified
# file
# Properties:
# jlog.snapmemory.queueCapacity : Sets the nnumber of LogEvents that can be buffered in the memory
# jlog.snapmemory.snapFile : name of the file to which the contents of the memory will be dumped
# jlog.snapmemory.baseDir : The directory where the snapFile is placed.
#     daily subdirectories will be created under this base directory, as:
#     [baseDir]/[YYYY-MM-DD]/
#     Note: MS-DOS style path names need to be be escaped with backslashes
#     eg: c:\\CTGI\\FFDC
# jlog.snapmemory.userSnapFile : The name of the file to which the user initiated (from logcmd) dumps
#     will be written to.
# jlog.snapmemory.userSnapDir : The directory where the userSnapfile is placed.
# jlog.snapmemory.msgIds : The list of TMS IDs
# jlog.snapmemory.msgIDRepeatTime : The minimum time, in milliseconds, after passing a log event with a
#     given TMS message id, before another log event with the same id can
#     be passed.
#-----
jlog.snapmemory.className=com.tivoli.log.SnapMemoryHandler
jlog.snapmemory.description=Memory handler used to trace to memory
jlog.snapmemory.queueCapacity=10000
jlog.snapmemory.dumpEvents=true
jlog.snapmemory.snapFile=trace.log
jlog.snapmemory.baseDir=CTGDI/FFDC/
jlog.snapmemory.userSnapFile=userTrace.log
jlog.snapmemory.userSnapDir=CTGDI/FFDC/user/
jlog.snapmemory.triggerFilter=jlog.levelflt
jlog.snapmemory.msgIds=*E
jlog.snapmemory.msgIDRepeatTime=10000

#-----
# Configure the JLogSnapHandler taking a snapshot of the SnapMemoryHanlders buffer
# The JLogSnapHanlder takes a snapshot of the associated SnapMemoryBuffer.
#-----
jlog.snaphandler.className=com.tivoli.log.JLogSnapHandler
jlog.snaphandler.description=snaphandler to dump the memory trace
jlog.snaphandler.baseDir=CTGDI/FFDC/
jlog.snaphandler.snapMemoryHandler=jlog.snapmemory
jlog.snaphandler.triggerFilter=jlog.levelflt

#-----
# Configure the PDLogger (Problem Determination) Object and attach the Listeners to it.
# jlog.logger.level can be FATAL | ERROR | WARNING | INFO | DEBUG_MIN | DEBUG_MID | DEBUG_MAX
# The heirarchy of the log levels is from the most severe (FATAL) to the least severe (DEBUG_MAX)
# The value for this property is case-insensitive
#-----
jlog.logger.level=FATAL
#jlog.logger.listenerNames=jlog.snapmemory jlog.snaphandler
jlog.logger.listenerNames=jlog.filehandler.trace
jlog.logger.className=com.ibm.log.PDLogger

#-----
# Configure the PDLogger for the Config Editor and attach the Listeners to it.
# By default, no listeners are attached
#-----
jlog.logger.config-editor.level=FATAL
jlog.logger.config-editor.listenerNames=

```

derby.properties

Diese Datei enthält einige Standardwerte für Derby im Netzmodus.

Die meisten IBM Security Directory Integrator-bezogenen Derby-Parameter werden nicht in dieser Datei verwaltet, sondern in den Dateien `global.properties` und `solution.properties`. Weitere Informationen zu diesen Parametern können Sie der Derby-Dokumentation entnehmen.

```
# This is a sample properties file provided to show the proper format.
# We're also setting one property which make sure that
# Derby adds to the error log instead of overwriting it.
# This mode is useful for development.
derby.drda.logConnections=true
derby.drda.maxThreads=0
derby.drda.portNumber=1527
derby.drda.traceAll=true
derby.drda.timeSlice=0
derby.drda.traceDirectory=/trace
```

global.properties

Diese Datei wird beim Start durch den Konfigurationseditor (`ibmditk`) und den Server (`ibmdisrv`) gelesen.

Sie wird gelesen und angewendet, bevor eine Datei namens `solution.properties` aus Ihrem Lösungsverzeichnis gelesen und angewendet wird.

Anmerkung:

Möglicherweise ist die folgende Darstellung aufgrund extremer Zeilenlängen nicht vollständig. Ziehen Sie stattdessen eine reale Datei `global.properties` hinzu.

```
##
## This file is read by ibmditk/ibmdisrv on startup
##
## Enter <name>=<value> to set system properties.
## Enter !include <file | url> to include other files
##

com.ibm.di.securityTransformation=DES/ECB/NoPadding

##
## Modify the line below to add your own jar/zip files.
## The property may specify several directories or jar files, separated by the Java Property "path.separator",
## which is ":" on Linux and ";" on Windows
## Directories will be searched recursively by the TDILoader for jar files containing classes and resources.
## Only files with a ".zip" or ".jar" extension are searched.
# com.ibm.di.loader.userjars=c:\myjars

##
## Modify the line below to enable the config autoload feature.
## When this property is defined, the "ibmdisrv -d" command
## line will look for *.xml files in the directory specified by this property and start each one.
##
# com.ibm.di.server.autoload=autoload.tdi

##
## SYSTEM STORE
##

## Location of the database (embedded mode) - Cloudscape 10
#com.ibm.di.store.database=TDISysStore
#com.ibm.di.store.jdbc.driver=org.apache.derby.jdbc.EmbeddedDriver
#com.ibm.di.store.jdbc.urlprefix=jdbc:derby:
#com.ibm.di.store.jdbc.user=APP
#{protect}-com.ibm.di.store.jdbc.password=APP

## Location of the database to connect (networked mode) - Cloudscape 10 - DerbyClient driver
## The macro $soldir$ will be replaced by the value of the actual Solution Directory
com.ibm.di.store.database=jdbc:derby://localhost:1527/$soldir$/TDISysStore;create=true
com.ibm.di.store.jdbc.driver=org.apache.derby.jdbc.ClientDriver
com.ibm.di.store.jdbc.urlprefix=jdbc:derby://localhost:1527/
com.ibm.di.store.jdbc.user=APP
{protect}-com.ibm.di.store.jdbc.password={encr}n+Vum7tN0ZU0KNp7AGy7pkAZqiJMgGpnqwg
/dBhLEL5pDBj5FY/Qp/2OmOkfWDezdSvYGUKag3UkzV+NuSSBvPJ36s3QckFDz72VOTzJalREhIp/j/u9
/3E11ZPIAH1B1gKp77200FPJIB6mbDUUgFwIZ+FmKFH5CW6Nyt+M=

#
## Derby (Cloudscape) properties required for enabling authentication
#
derby.drda.startNetworkServer=true
derby.connection.requireAuthentication=true
derby.authentication.provider=BUILTIN
derby.database.defaultConnectionMode=fullAccess
```

```

##
## Details for starting Cloudscape in network mode.
## Note: If the com.ibm.di.store.hostname is set to localhost then remote connections will not be allowed.
## If it is set to the IP address of the local machine - then remote clients can access this Cloudscape
## instance by mentioning the IP address. The network server can only be started for the local machine.
##
#com.ibm.di.store.start.mode=automatic
com.ibm.di.store.hostname=localhost
com.ibm.di.store.port=1527
com.ibm.di.store.sysibm=true

# the varchar(length) for the ID columns used in system store and pes connector tables
com.ibm.di.store.varchar.length=512

## create statements for system store tables (CloudScape 5.1)
#com.ibm.di.store.create.delta.systable=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL, SEQUENCEID int, VERSION int)
#com.ibm.di.store.create.delta.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL, SEQUENCEID int, ENTRY long varbinary )
#com.ibm.di.store.create.property.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL, ENTRY long varbinary )
#com.ibm.di.store.create.sandbox.store=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL, ENTRY long varbinary )

## create statements for system store tables (CloudScape 10)
com.ibm.di.store.create.delta.systable=CREATE TABLE {0} (ID VARCHAR(VARCHAR_LENGTH) NOT NULL, SEQUENCEID int, VERSION int);
ALTER TABLE {0}
ADD CONSTRAINT IDI_CS_{UNIQUE} PRIMARY KEY (ID)
com.ibm.di.store.create.delta.store=CREATE TABLE {0}
(ID VARCHAR(VARCHAR_LENGTH) NOT NULL, SEQUENCEID int, ENTRY BLOB );
ALTER TABLE {0} ADD CONSTRAINT IDI_DS_{UNIQUE} Primary Key (ID)
com.ibm.di.store.create.property.store=CREATE TABLE {0}
(ID VARCHAR(VARCHAR_LENGTH) NOT NULL, ENTRY BLOB );
ALTER TABLE {0} ADD CONSTRAINT IDI_PS_{UNIQUE} Primary Key (ID)
com.ibm.di.store.create.sandbox.store=CREATE TABLE {0}
(ID VARCHAR(VARCHAR_LENGTH) NOT NULL, ENTRY BLOB )
com.ibm.di.store.create.recal.conops=CREATE TABLE {0}
(METHOD varchar(VARCHAR_LENGTH), RESULT BLOB, ERROR BLOB)

## create statements for system store tables DB2 on z/OS
#com.ibm.di.store.create.delta.systable=CREATE TABLESPACE TS1DSYS LOCKSIZE ROW BUFFERPOOL BP32K;CREATE TABLE {0}
(ID VARCHAR(VARCHAR_LENGTH) NOT NULL, SEQUENCEID int, VERSION int) IN TS1DSYS;CREATE UNIQUE INDEX DST1X1 ON {0} (ID ASC);
ALTER TABLE {0}
ADD CONSTRAINT IDI_DT_{UNIQUE} PRIMARY KEY (ID)
#com.ibm.di.store.create.delta.store=CREATE TABLESPACE TS1DST LOCKSIZE ROW BUFFERPOOL BP32K;CREATE TABLE {0}
(ID VARCHAR(VARCHAR_LENGTH) NOT NULL, SEQUENCEID int, ENTRY BLOB) IN TS1DST; CREATE UNIQUE INDEX DS1X1 ON {0} (ID ASC);
ALTER TABLE {0} ADD CONSTRAINT IDI_DS_{UNIQUE} Primary Key (ID);CREATE LOB TABLESPACE DSENT11 BUFFERPOOL BP32K
LOCKSIZE LOB;CREATE AUX TABLE TBDSEN1 IN DSENT11 STORES {0}
COLUMN ENTRY;CREATE INDEX IXEN1 ON TBDSEN1
#com.ibm.di.store.create.property.store=CREATE TABLESPACE PS3DST
LOCKSIZE ROW BUFFERPOOL BP32K;CREATE TABLE {0}
(ID VARCHAR(VARCHAR_LENGTH) NOT NULL, ENTRY BLOB) IN PS3DST;CREATE UNIQUE INDEX PS1X3 ON {0}
(ID ASC);ALTER TABLE {0}
ADD CONSTRAINT IDI_PS_{UNIQUE} Primary Key (ID);CREATE LOB TABLESPACE PSENT31 BUFFERPOOL BP32K
LOCKSIZE LOB;CREATE AUX TABLE TBPSEN3 IN PSENT31 STORES {0}
COLUMN ENTRY;CREATE INDEX PSIXEN3 ON TBPSEN3
#com.ibm.di.store.create.sandbox.store=CREATE TABLE {0}
(ID VARCHAR(VARCHAR_LENGTH) NOT NULL, ENTRY BLOB)
#com.ibm.di.store.create.recal.conops=CREATE TABLESPACE IM{UNIQUE} LOCKSIZE ROW BUFFERPOOL BP32K;CREATE TABLE {0}
(METHOD VARCHAR(VARCHAR_LENGTH), RESULT BLOB, ERROR BLOB) IN IM{UNIQUE};CREATE LOB TABLESPACE LB{UNIQUE} BUFFERPOOL BP32K LOCKSIZE LOB;CREATE AUX TABLE AT{UNIQUE}

# Set a customized SQL statement for creation of the Tombstone Manager table.
# Keep the same table and field names. This is the default Derby statement.
#com.ibm.di.store.create.tombstones=CREATE TABLE IDI_TOMBSTONE ( ID INT GENERATED ALWAYS AS IDENTITY, COMPONENT_TYPE_ID INT,
EVENT_TYPE_ID INT, START_TIME TIMESTAMP, CREATED_ON TIMESTAMP,
COMPONENT_NAME VARCHAR(1024), CONFIGURATION VARCHAR(1024), EXIT_CODE INT,
ERROR_DESCR VARCHAR(1024), STATS LONG VARCHAR FOR BIT DATA, GUID VARCHAR(1024)
NOT NULL, USER_MESSAGE VARCHAR(1024), UNIQUE (ID, GUID))

# The following two SQL statements could be used when SolidDB is used as System Store
#com.ibm.di.store.create.tombstones=CREATE TABLE IDI_TOMBSTONE (ID INT PRIMARY KEY, COMPONENT_TYPE_ID INT,
EVENT_TYPE_ID INT, START_TIME TIMESTAMP, CREATED_ON TIMESTAMP, COMPONENT_NAME VARCHAR(1024),
CONFIGURATION VARCHAR(1024), EXIT_CODE INT, ERROR_DESCR VARCHAR(1024), STATS LONG VARBINARY,
GUID VARCHAR(1024) NOT NULL, USER_MESSAGE VARCHAR(1024), UNIQUE (ID, GUID));CREATE SEQUENCE IDI_TOMBSTONE_SEQ
#com.ibm.di.store.update.tombstones=INSERT INTO IDI_TOMBSTONE (ID, COMPONENT_TYPE_ID, EVENT_TYPE_ID, START_TIME,
CREATED_ON, COMPONENT_NAME, CONFIGURATION, EXIT_CODE, ERROR_DESCR, STATS, GUID, USER_MESSAGE)
VALUES (IDI_TOMBSTONE_SEQ.NEXT, ?, ?, ?, ?, ?, ?, ?, ?, ?)

# the ibmsnap_commitseq column name used by the RDBMS changelog connector
com.ibm.di.conn.rdbmschlog.cdcolname=ibmsnap_commitseq

## server authentication
javax.net.ssl.trustStore=serverapi/testadmin.jks
(protect)-javax.net.ssl.trustStorePassword={encr}rI2mXgg5vwrBnooxTCYUFEMskCKHb14cGpZuQum20GeCaYJjQgH1SjLuQJdQnNqDNJi+
isW0mJkzw10qud81179x1JtwqLh7zEVfGvuqEwx43ACLoSb9gkG8Je07Jf0Fgp09thj6zTCCzqB4NCsmUv11agBhvGtE8Q73Xh8=
javax.net.ssl.trustStoreType=jks

## client authentication
javax.net.ssl.keyStore=serverapi/testadmin.jks
(protect)-javax.net.ssl.keyStorePassword={encr}QnVT+6gn6nE3zUnvHQEp9TM8k52BQxJsaE85mmwQkechSeScysd0MrWl tXCOMF2RJoZL
cqLp3WpGp0QX+n9XQVBXG0XmIjhwIs7iDr73dpkM1oJzSruYhKpQGK0gfc+801IAfSoMiNggp76iYyqiRqJHKV3sSQ79VB0mM=
javax.net.ssl.keyStoreType=jks

##PKCS11 options
##Set the value of following properties to use PKCS11 enabled devices to store SDI servers private key / certificate.
com.ibm.di.pkcs11cfg=etc/pkcs11.cfg
com.ibm.di.server.pkcs11=false
com.ibm.di.server.pkcs11.library=
com.ibm.di.server.pkcs11.slot=
(protect)-com.ibm.di.server.pkcs11.password={encr}iYbgH/Y/pw4YSUXVqKQnWz1ZHKa52CuCRnHnnBen3/yI0tJ1K+nrepEWBjN2KBDHM
8Z+z1Ps/0Y1Ix9y20X/nvp/QvevgPsAvvxznd7QtwjQyS7KwJT+11BVCbnkhJ0iRXxzIqkwSxty1/oUcdskk5+wMSYltvuSur03J38=

## Turns on java debug
# javax.net.debug=true

## java interpreter override
# com.ibm.di.javacmd=
# com.ibm.di.installDir=

```

```

## Limits the number of threads IDI uses
## Must be set higher than 3 to have any effect

# com.ibm.di.server.maxThreadsRunning=500

com.ibm.di.server.securemode=false

## Following properties modified in SDI 7.1 Added property for
## keystore password and keypassword
## com.ibm.di.server.keystore
## com.ibm.di.server.key.alias

api.keystore=testserver.jks
api.keystore.type=jks
api.key.alias=server
(protect)-api.keystore.password={encr}S8BAYIIIdwly0FHuslpXHN6F4Xc76gCUvm39ZKZHSjRMmZQmY7S1p+7g9YHE3AIquL6gf4n0MzFyBU
S/C6xy2RI1jVbq+HGz2YU+4SNnGjt5o53KAXiegLC2ml+JCUu4UY/P/ASjCXFhL/iJsRI4hxLhFg266p13eIJeVxf1c=
(protect)-api.key.password={encr}

## Encryption properties added in SDI 7.1
com.ibm.di.server.encryption.keystore = testserver.jks
com.ibm.di.server.encryption.key.alias = server
com.ibm.di.server.encryption.keystoretype = jks
com.ibm.di.server.encryption.transformation = RSA

## Web container
web.server.port=1098
web.server.ssl.on=trueweb.server.ssl.client.auth.on=false
# web.server.session.timeout=300

## Touchpoint Server properties
tp.server.on=false
tp.server.config=etc/tp.xml
tp.server.auth=false
tp.server.auth.realm=Security Directory Integrator Touchpoint Server

## Dashboard properties
##
dashboard.on=true
dashboard.templates.folder=dashboard/templates

## Dashboard authentication properties
##
## The values for localhost and remotehost can be:
## none: No authentication is required
## deny: All connections denied
## ldap: Authentication is done by logging into an LDAP server and optionally validating group membership
## properties: Authentication is done using dashboard.auth.user.[username]=[password] properties
##
## dashboard.ldap.url
## Specify the LDAP host port and optionally a search base (ldap://<host>:<port>[/<search-base>])
##
## dashboard.ldap.url.group
## Specify the LDAP host port and optionally a search base (ldap://<host>:<port>[/<search-base>])
##
dashboard.auth=true
dashboard.auth.localhost=properties
dashboard.auth.remote=deny
# dashboard.auth.ldap.url=ldap://localhost:389/ou=users,ou=system
# dashboard.auth.ldap.url.group=ldap://localhost:389/cn=group1,ou=groups,ou=system
#
# Default FDS username/password
(protect)-dashboard.auth.user.admin={encr}SzFT+3+aSNnWrtySrBcCbh1Vp4bB4hKsJqJGuRSwtn69b1f/UiPbRQbWmQhFidmGpxEULTS9
S+x4nX0J7rDY2DPVmdfK0u4xqAWT8euS9Nv1Ep4MfB/whoipQhTWFT3PSVvt+uCc+ONhKun0QuE55IKwAQKdyHPTz+cjkeNM=

## Server API properties
## -----

api.on=true
api.audit.on=false
api.user.registry=serverapi/registry.txt
api.user.registry.encryption.on=false

api.remote.on=true
api.remote.ssl.on=true
api.remote.ssl.client.auth.on=true
api.remote.naming.port=1099
# api.remote.server.ports=8700-8900api.truststore=testserver.jks
api.truststore.type=jks
(protect)-api.truststore.pass={encr}DzTm01+sUaose3wpcbHk9vzZ4JZxHL8aMC2ePub4tWmuS+D70VcLI5aS8sayg0/ktC0cH6ozy6+qxh1an
PpYtu1Dh7mZHDsAGDL+Temard/gJUT1xuG4FKAIr5YsDxhZ3n1d5fLa8h8YMTVDLd8qx6XZ16f/Ag0a0Yzn882wwFI=

## REST API
## -----
api.rest.on=true
api.rest.auth=false
api.rest.auth.realm=Security Directory Integrator REST API

api.rest.jmsdriver.name=com.ibm.di.systemqueue.driver.ActiveMQ
api.rest.jmsdriver.queue.sender.persistence=false
api.rest.jmsdriver.queue.sender.timeToLive=60000
api.rest.jmsdriver.param.jms
oker=vm://localhost?brokerConfig=xbean:etc/activemq.xml
# api.rest.jmsdriver.auth.username
# api.rest.jmsdriver.auth.password

## The properties determine the default bind address and the remote bind address for the Server API.
## * means bind to all network interfaces. The Remote Bind Address overrides the Default one.
## Only one IP address should be set. No hostnames are accepted.
## Mind that the java.rmi.server.hostname property is set implicitly to equal the Remote Bind Address property when used.
##This will cause the client stubs to create sockets on the specified Remote Bind Address.
# com.ibm.di.default.bind.address=*
# api.remote.bind.address=*
## Specifies a list of IP addresses to accept non SSL connections from (host names are not accepted).
## Use space, comma or semicolon as delimiter between IP addresses. This property is only taken into account

```

```

## when api.remote.ssl.on is set to false.
## api.remote.nonssl.hosts= api.jmx.on=false api.jmx.remote.on=false

## The configuration files placed in this folder can be edited through the Server API.
## Configuration files placed in other folders cannot be edited through the Server API.
api.config.folder=configs

## Timeout in minutes for configuration locks. A value of 0 means no timeout.
api.config.lock.timeout=0

## Timeout in minutes for loading a configuration.
api.config.load.timeout=2

## Specifies if the Server API methods for custom method invocation (Session.invokeCustom(...)) are allowed to be used.
## When api.custom.method.invoke.on is set to false and the Server API methods for custom method invocation are used,
## then an exception will be thrown.
## Only classes listed in api.custom.method.invoke.allowed.classes are allowed to be directly invoked.
## The default value is false. api.custom.method.invoke.on=false

## Specifies the list of classes which can be directly invoked by the Server API methods for custom
## method invocation (Session.invokeCustom(...)).
## This property is only taken into account if api.custom.method.invoke.on is set to true.
## The classes in this list must be separated by a space, a comma or a semicolon.
## Example:
## api.custom.method.invoke.allowed.classes=com.ibm.MyClass,com.ibm.MyOtherClass
## In the above example only methods from the com.ibm.MyClass and com.ibm.MyOtherClass classes are
## allowed to be directly invoked. api.custom.method.invoke.allowed.classes=

## Specifies a list of Server notification types, which will be suppressed.
## Notifications of suppressed types will not be propagated by the notifications framework.
## The notification types in the list are separated by spaces. Wildcards may be included.
## Example:
## api.notification.suppress=di.al.* di.ci.start
## The above example will suppress all Assembly Line related notifications as well as
## notifications for starting a configuration instance.
## If the property is missing or is empty, no notifications will be suppressed.
api.notification.suppress=di.server.api.authenticate di.server.api.authorize.*
## api.custom.authentication points to a JavaScript text file that contains custom authentication code.
## For example: api.custom.authentication=ldap_auth.js
## To enable the built-in LDAP Authentication mechanism, set this property to "[ldap]".
## To enable the built-in JAAS Authentication mechanism, set this property to "[jaas]".
## For example: api.custom.authentication=[ldap]

##api.custom.authentication=[ldap]

## LDAP Authentication properties
## -----

## If this parameter is set to "true" and the LDAP Authentication initialization fails,
## the whole Server API will not be started.
## If this parameter is missing or is set to "false" any LDAP Authentication initialization errors will be logged
## and the Server API will be started.
api.custom.authentication.ldap.critical=false

## LDAP Server hostname. api.custom.authentication.ldap.hostname=

## LDAP server port number. (z. B. 389 ohne SSL oder 636 mit SSL).
api.custom.authentication.ldap.port=

## Specifies whether SSL is used to communicate with the LDAP Server.
## When set to "true" SSL will be used, otherwise SSL will not be used.
api.custom.authentication.ldap.ssl=

## Specifies the LDAP directory location where user searches will be performed.
## When this property is not specified user searches will not be performed.
api.custom.authentication.ldap.searchbase=

## Specifies the user id attribute to be used in searches.
## When this property is not specified user searches will not be performed.
api.custom.authentication.ldap.userattribute=

## Specifies an LDAP Server administrator distinguished name that will be used for user searches.
## When this property is not specified anonymous bind will be used for user searches.
api.custom.authentication.ldap.admindn=

## Password for the LDAP Server administrator distinguished name.
(protect)-api.custom.authentication.ldap.adminpassword={encr}

## This property specifies whether LDAP Group authentication is turned on.
## If it is set to 'true', the group membership of the authenticating user will be resolved
## and will be taken into account during authorization.
## If it is missing, the default value 'false' is used.
api.custom.authentication.ldap.groupsupport=false

## Specifies the name of the attribute of a user in LDAP that contains a list of the groups
## of which the user is a member.
## It is taken into account only if 'api.custom.authentication.ldap.groupsupport' is set to true.
api.custom.authentication.ldap.usermembershipattribute=

## Specifies how groups are named in the membership attribute of a user.
## For example, if the user's membership attribute contains values, which correspond to the 'objectSID' attributes
## of groups, set this property to 'objectSID'.
## If the user's membership attribute contains distinguished names of groups, then set this property to 'dn'.
## The property is required in case 'api.custom.authentication.ldap.groupsupport' is set to true.
api.custom.authentication.ldap.usermembershipattributecontent=

## Specifies the name of a group's attribute in LDAP, which corresponds to the way the
## group is named in the SDI User Registry.
## For example, if LDAP groups are addressed in the SDI registry by their common name, then set this property to 'cn'.
## If the User Registry contains the distinguished names of the groups, then set this property to 'dn'.
api.custom.authentication.ldap.groupnameattribute=

## Represents the LDAP directory context, where groups will be searched.
## It is required only when LDAP group support is enabled
api.custom.authentication.ldap.groupsearchbase=

```



```

## Optional property, which represents a list of space-separated attribute names.
## Specifies attributes which have non-string syntax.
## api.custom.authentication.ldap.binaryattributes=

## JAAS Authentication properties
## -----
java.security.auth.login.config=

## Enabling/Disabling FIPS Mode in SDI
## -----
## If the below property is set to true then SDI will be enforced to run in FIPS Compliant Mode.
## The default value is false, i.e. SDI will not run in FIPS Mode by default.
com.ibm.di.server.fipsmode.on=false

## Specify the unique ID for the SDI Server
## -----
## This property helps a client connecting to the SDI server to identify different servers
## running on the same IP and the same port in different time. (Default is DEFAULT_ID)
com.ibm.di.server.id=DEFAULT_ID

## Tombstone Manager properties
## -----

com.ibm.di.tm.on=false
com.ibm.di.tm.autodel.age=0
com.ibm.di.tm.autodel.records.trigger.on=10000
com.ibm.di.tm.autodel.records.max=5000
com.ibm.di.tm.create.all=false

## -----
## Help system properties
## -----

## Name of help server. The Tivoli library is at the following URL:
## http://www-01.ibm.com/support/knowledgecenter/SSCG6F/welcome

## Port for help system
com.ibm.di.helpPort=80

## -----
## AssemblyLinePool: Connector pooling defaults
## -----
##
## Note! These settings are only used when an AssemblyLine uses
## an AssemblyLinePool in combination with a Server mode connector.

## The number of seconds before a pooled connector times (e.g. is closed and no longer reused)
## Less than zero means disable connector pooling
## Zero means never timeout
## Greater than zero sets the number of seconds before a connector is closed
com.ibm.di.server.connectorpooltimeout=42

## Comma separated list of connector interfaces that we never pool
com.ibm.di.server.connectorpoolexclude=com.ibm.di.connector.FileConnector,com.ibm.di.connector.ScriptConnector

## Properties for Windows IPv6 communications.
## Uncomment these properties for Windows IPv6 communication only.
## These properties will not affect IPv4 communication or IPv6 communication on Unices.
#java.net.preferIPv4Stack=false
#java.net.preferIPv6Addresses=true

## -----
## Performance settings
## -----
##
## Enable/Disable performance logging com.ibm.di.server.perfStats=false

### -----
### Used by Config Report
### -----
### set this is you want to override the local language for Config Reports
# com.ibm.di.admin.configreport.translation=en

## -----
## System Queue settings
## -----
## If set to "true" the System Queue is initialized on startup and can be used;
## otherwise the System Queue is not initialized and cannot be used.
systemqueue.on=true

## Specifies the fully qualified name of the class that will be used as a JMS Driver.
# systemqueue.jmsdriver.name=com.ibm.di.systemqueue.driver.IBMMQ
# systemqueue.jmsdriver.name=com.ibm.di.systemqueue.driver.JMSJavaScriptDriver
systemqueue.jmsdriver.name=com.ibm.di.systemqueue.driver.ActiveMQ

### MQ JMS driver initialization properties
# systemqueue.jmsdriver.param.jms
oker=<host:port>
# systemqueue.jmsdriver.param.jms.serverChannel=<channel_name>
# systemqueue.jmsdriver.param.jms.qManager=<queuemanger_name>
# systemqueue.jmsdriver.param.jms.sslCipher=<cipherSuite_name>
# systemqueue.jmsdriver.param.jms.sslUseFlag=false

### JMS Javascript driver initialization properties
## Specifies the location of the script file
# systemqueue.jmsdriver.param.js.jsfile=driver.js

### ActiveMQ driver initialization properties
## Specifies the location of the ActiveMQ initialization file.
## This file is used to initialize ActiveMQ on SDI server startup.
systemqueue.jmsdriver.param.jms
oker=vm://localhost?brokerConfig=xbean:etc/activemq.xml

## This is the place to put any JMS provider specific properties needed by a JMS Driver,

```

```

## which connects to a 3rd party JMS system.
## All JMS Driver properties should begin with the 'systemqueue.jmsdriver.param.' prefix.
## All properties having this prefix are passes to the JMS Driver on initialization after
## removing the 'systemqueue.jmsdriver.param.' prefix from the property name.
# systemqueue.jmsdriver.param.user.param1=value1
# systemqueue.jmsdriver.param.user.param2=value2
# ...

## Credentials used for authenticating to the target JMS system
# {protect}-systemqueue.auth.username=<username>
# {protect}-systemqueue.auth.password=<password>

## -----
## Logging settings
## -----

## When false, all log calls made through the SDI Log class will be discarded.
com.ibm.di.logging.enabled=true

## -----
## IBM JavaScript Engine settings
## -----

## Set the type of platform - required by the IBM JS Engine when caching is used.
com.ibm.commons.platform=com.ibm.commons.platform.GenericPlatform

##
## Set this property to a directory to enable auto dumps of assemblylines that fails
##
# com.ibm.tdi.autodump.directory=<dump-directory>

##
## Server API client properties
##
api.client.ssl.custom.properties.on=true
api.client.keystore=serverapi/testadmin.jks
{protect}-api.client.keystore.pass={encr}E/1TC2+UF4B9BACaVcdKoa1Yj38LFi26UncEFTsWP+qj68fuJH8EwCs392IoLxvIAKH1PaWwJo7h
ySkeBxESorVACAF8oNbHXIBQOS2nvNPKx1kQZK32CaR1g5Wq8TNamQxHF+++UewBuDEAU1ki5zZxidD2g8g0sN6cK1IhJo=
api.client.keystore.type=jks
{protect}-api.client.key.pass={encr}EU1+JvyysxVts7Yn236FysDdxV7IP7TmjCOy/si0m6x8H6Hcy1epHjM6yunQcqiQeN/KpL7M0a3uqzkwk
cWMURmrOWR+08xyoN+hMpoAu1EvmXjubtd7jdBJvncesL5BYiSwcxGeTsnQJ/MN84RiCfGc1FzwYxvL53npGeyGXk=
api.client.truststore=serverapi/testadmin.jks
{protect}-api.client.truststore.pass={encr}Y8Np19khRasEUfwYSaAM5RkJ+NO00KezRkXgbLyZtU1V0sFijJfCkeLmw8+MndHvtVkPM/3N1n
g/y+zv9NKFABVqBVYTJxK5RZx3YV/IgQJMpJK/YhT2hSR8w5XSM7meJ01JK3NjC+Cy9my42ioyT+svLUgpQfs740XyL8482ww=
api.client.truststore.type=jks
## -----
## Mail properties
## -----

## This property needs to be set to a valid SMTP host to be able
## to send mail using the system methods.
## mail.smtp.host=

## -----
## Enabling/Disabling NIST Mode in SDI
## -----
## If the below property is set to true then SDI will be enforced to run in NIST Compliant Mode.
## The default value is false, i.e. SDI will not run in NIST Mode by default.
com.ibm.di.server.NIST.on=false

```

Anhang B. Überwachung mit externen Tools

Sie können IBM SDI mit externen Tools wie Tivoli Monitoring und Tivoli Netcool/OMNIbus überwachen. Nachstehend erhalten Sie Informationen zu deren Funktionsweise und den entsprechenden Komponenten.

Die hier beschriebene Funktionalität ist eine Art "Einstieg" in die Integration von IBM Security Directory Integrator, IBM Tivoli Monitoring und IBM Tivoli Netcool/OMNIbus. Ausgangspunkt war der Machbarkeitsnachweis für dieses Integrations-szenario. Die im vorliegenden Dokument beschriebenen Integrationsfunktionen für IBM Security Directory Integrator mit IBM Tivoli Monitoring und IBM Security Directory Integrator mit OMNIbus sind vollständig unterstützte Lösungen, die mit IBM Security Directory Integrator ausgeliefert werden. Die Lösungen werden zwar im Verzeichnis "examples" bereitgestellt, werden aber vollständig unterstützt.

Für die Kommunikation zwischen IBM Security Directory Integrator und ITM wurde JMX gewählt, weil IBM Security Directory Integrator eine sofort einsatzfähige JMX-Schnittstelle bereitstellt und somit auf der Seite von IBM Security Directory Integrator keine Entwicklung erforderlich war.

Zur Überwachung von IBM Security Directory Integrator über Tivoli Netcool/OMNIbus wurde eine Fertigungslinie entwickelt, damit IBM Security Directory Integrator-Ereignisse erkannt und an OMNIbus gesendet werden können. Im vorliegenden Abschnitt ist dargestellt, wie IBM Security Directory Integrator durch die folgenden Produkte überwacht werden kann:

- Tivoli Monitoring mit Verwendung der JMX-Schnittstelle von IBM Security Directory Integrator
- Tivoli Netcool/OMNIbus

Beide Integrationsszenarios sind als offizielle IBM Security Directory Integrator-Beispiele im Produktpaket enthalten. Sie befinden sich im Verzeichnis *tdi-installationsverzeichnis/examples/Tivoli_Monitoring*.

Für diese Beispiele wurden ITM 6.2.0 und Tivoli Netcool/OMNIbus 7.2.1 verwendet.

Zur Realisierung der hier beschriebenen Versuche wurden mehrere Softwarekomponenten benötigt. Diese Komponenten sind, zusammen mit der Referenzdokumentation, die zur Realisierung ihrer Installation eingesetzt wurde, in der folgenden Liste angegeben:

- ITM Agent Builder 6.2 - ITM Agent Builder 6.2 Benutzerhandbuch
- ITM Tivoli Enterprise Portal - Onlinedokumentation von ITM Tivoli Enterprise Portal
- Tivoli Netcool/OMNIbus 7.2.1 - Onlinedokumentation von Tivoli Netcool/OMNIbus

Für die Kommunikation zwischen IBM Security Directory Integrator und Tivoli Monitoring wird JMX verwendet. Auf der IBM Security Directory Integrator-Seite stellt Tivoli Monitoring eine Verbindung zur JMX-Schicht der Server-API her.

Für die Kommunikation zwischen IBM Security Directory Integrator und Tivoli Netcool/OMNIbus werden zwei Connectors eingesetzt. Mit einem Serverbenachrichtigungsconnector wird eine Reihe von IBM Security Directory Integrator-Serverbenachrichtigungen empfangen, zum Senden von Ereignissen an OMNIbus wird ein EIF-Connector verwendet.

IBM Security Directory Integrator mit ITM überwachen

Nachstehend erhalten Sie Informationen zur Architektur, zum Import einer vorhandenen Konfiguration, zur Erstellung von Agenten sowie zu vielen weiteren Aspekten von ITM.

Kurzdarstellung der ITM-Architektur

Mit ITM-Agenten können Sie Daten erfassen und das System überwachen. Nachstehend erhalten Sie Informationen zu den Agententypen.

Der mit ITM bereitgestellte Browser stellt im Kern Daten dar, die durch Agenten erfasst werden.

ITM-Agenten werden gemäß der ITM-Dokumentation durch die folgende Definition charakterisiert:

Die Agenten (die als "verwaltete Systeme" bezeichnet werden), sind auf dem System oder Subsystem installiert, für das eine Datenerfassung und Überwachung benötigt wird. Die Agenten sind für die Erfassung von Daten und Verteilung von Attributen an die Überwachungsserver zuständig. Dies schließt auch die Initialisierung des Überwachungssignalstatus ein.

Es kann verschiedene Typen von Agenten geben, nämlich Agenten für die Überwachung von Betriebssystemen bzw. bestimmten Anwendungen oder speziell angepasste Agenten (d. h. unter Verwendung der Schnittstelle "Universal Agent"). Das folgende, aus der ITM-Dokumentation stammende Diagramm verdeutlicht sowohl die Architektur als auch den Implementierungsprozess der Agenten:

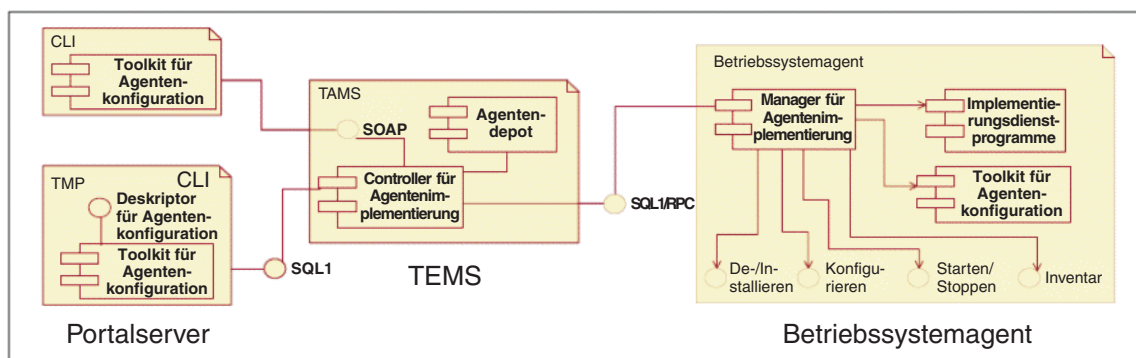


Abbildung 2. Diagramm der ITM-Agenten

TEMS = Tivoli Enterprise Monitoring Services
TEP = Tivoli Enterprise Portal

Vorhandene Agentenkonfiguration in ITM Agent Builder 6.2 importieren

Führen Sie die nachstehend beschriebenen Schritte aus, um eine vorhandene Agentenkonfigurationsdatei zu importieren.

Falls eine XML-Konfigurationsdatei für einen ITM-Agenten vorhanden ist, können Sie diese in ITM Agent Builder 6.2 importieren und auf diese Weise das ITM-Agentenprojekt automatisch erstellen lassen. Zum Importieren einer solchen Datei klicken Sie mit der rechten Maustaste auf eine Stelle im Arbeitsbereich von ITM Agent Builder. Wählen Sie die Option **Importieren** aus und wählen Sie anschließend die Option zum Importieren von IBM Tivoli Monitoring Agent aus. Zeigen Sie mit dem Mauszeiger auf die XML-Konfigurationsdatei (Standardname: itm_toolkit_agent.xml) und klicken Sie auf **Fertigstellen**. Hierdurch wird ein ITM Agent Builder-Projekt mit dem entsprechenden Namen erstellt.

Anmerkung: Falls Sie den Agenten eigenständig erstellen wollen, fahren Sie mit den Anweisungen im Abschnitt „IBM SDI-Agenten für ITM mit ITM Agent Builder 6.2 erstellen“ fort. Andernfalls wechseln Sie zum Abschnitt „ITM-Agenten generieren“ auf Seite 404.

IBM SDI-Agenten für ITM mit ITM Agent Builder 6.2 erstellen

Mit den Schritten im aufgeführten Beispiel können Sie IBM SDI-Agenten für ITM mit ITM Agent Builder 6.2 erstellen.

ITM Agent Builder ist eine Eclipse-basierte Plattform für die Erstellung von ITM-Agenten. Der im folgenden Beispiel erstellte Agent verwendet die JMX-Schnittstelle. Wählen Sie in ITM Agent Builder die Optionen **Datei -> Neu -> IBM Tivoli Monitoring Agent** aus.

Daraufhin wird der Assistent für IBM Tivoli Monitoring-Agenten aufgerufen. Seine erste Anzeige enthält eine Einführung. Klicken Sie in dieser Anzeige auf **Weiter**. In der zweiten Anzeige werden Sie aufgefordert, einen Projektnamen einzugeben. Im vorliegenden Beispiel wird der Projektname "SDI" verwendet. Nachdem Sie auf **Weiter** geklickt haben, wird die folgende Anzeige ausgegeben:

IBM Tivoli Monitoring Agent Wizard

Agent Information

Specify the general information for the agent.

Service name *Monitoring Agent for* TDI

Company identifier IBM

Agent identifier TDIAgent

Display name TDI

Product code K80

Version 620

Support multiple instances of this agent

Copyright IBM

< Back **Next >** Finish Cancel

Abbildung 3. Anzeige "Agenteninformationen" des Assistenten für IBM Tivoli Monitoring-Agenten

Geben Sie in alle Felder die zutreffenden Angaben ein. Der Produktcode sollte für JMX-Agenten zwischen K80 und K99 liegen. Klicken Sie auf **Weiter**. Wählen Sie in der nächsten Anzeige die Option **Dieser Agent erfasst Daten aus einer Datenquelle** aus und klicken Sie auf **Weiter**. Daraufhin wird das Fenster "Datenquellendefinition" angezeigt:

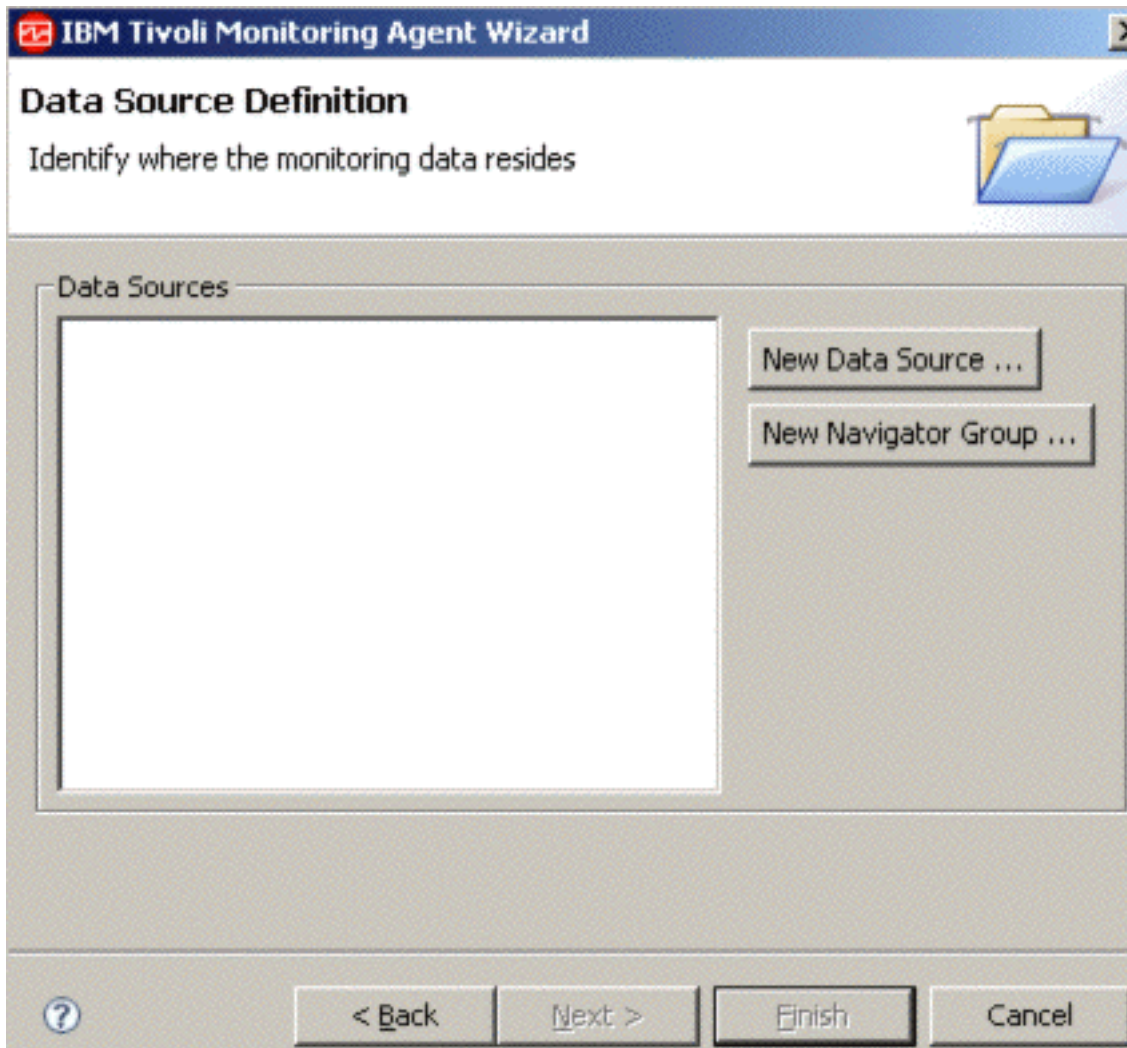


Abbildung 4. Fenster "Datenquellendefinition" des Assistenten für IBM Tivoli Monitoring-Agenten

Um die Konfiguration in diesem Schritt zu vereinfachen, starten Sie einen IBM Security Directory Integrator-Server im Dämonmodus und führen Sie eine Fertigungslinie aus, die in keinem Fall beendet wird (beispielsweise eine Fertigungslinie mit einem HTTP-Server-Connector, der für Verbindungen empfangsbereit ist). Stellen Sie sicher, dass die JMX-API in IBM Security Directory Integrator aktiviert ist (im Beispiel ist an einer späteren Stelle beschrieben, wie Sie dies ausführen).

Klicken Sie auf die Schaltfläche **Neue Datenquelle...** und wählen Sie dann die Option **Daten aus Java Management Extensions (JMX)-MBeans erfassen** aus. Klicken Sie auf **Weiter**. Klicken Sie im nächsten Fenster auf **Durchsuchen**. Daraufhin sollte der Browser für Java Management Extensions (JMX) angezeigt werden:

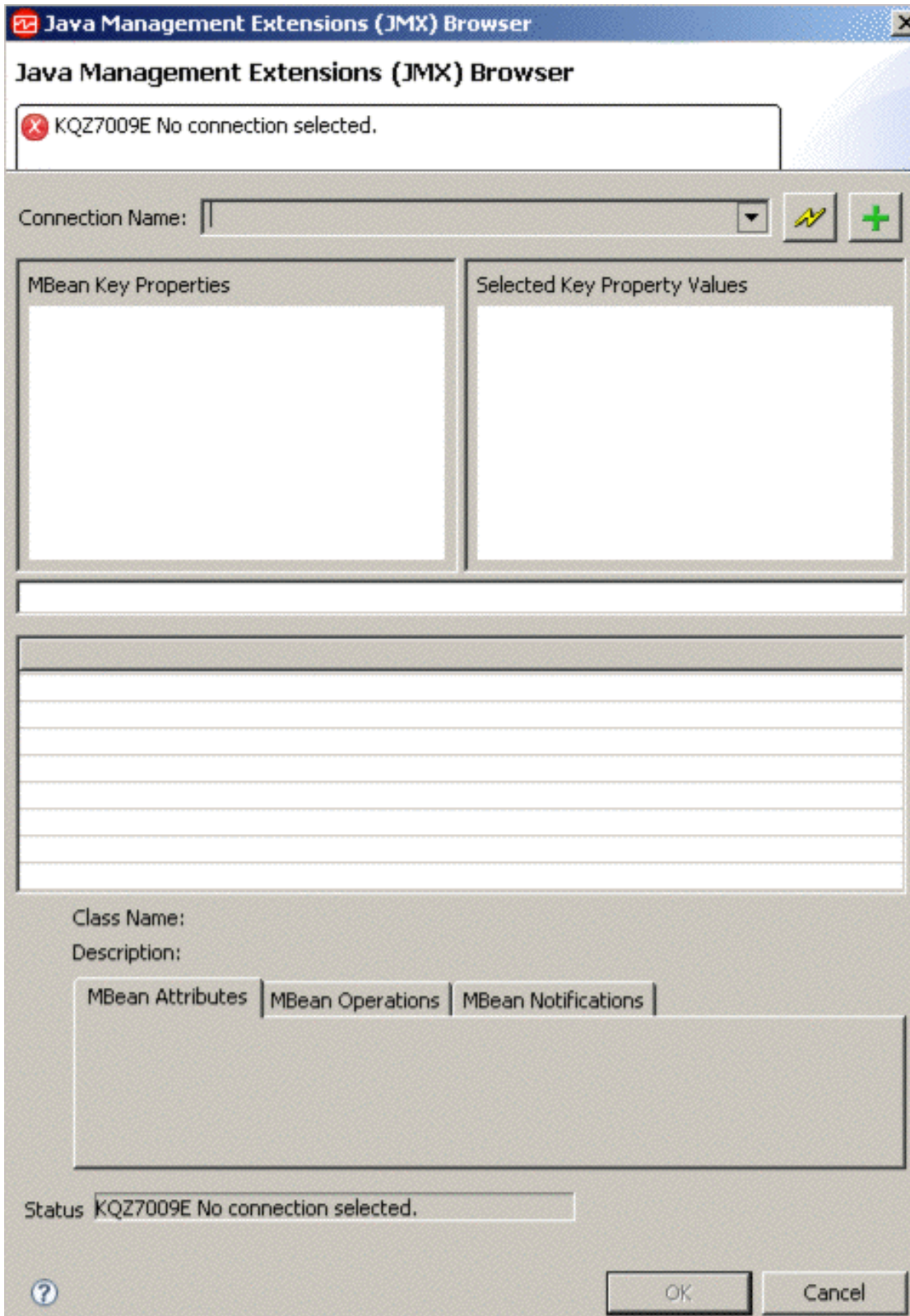


Abbildung 5. Browser für Java Management Extensions (JMX)

Klicken Sie auf die Schaltfläche **Verbindungsdefinitionen bearbeiten**. Dies ist die Schaltfläche mit dem grünen Pluszeichen. Wählen Sie in der nächsten Anzeige die Einstellung für **JMX-Standardverbindungen (JSR-160)** aus und klicken Sie auf

Weiter. Im neuen Assistentenfenster werden die verfügbaren Schablonen angezeigt. Wählen Sie die Einstellung **JSR-160-konformer Server** aus und klicken Sie erneut auf **Weiter**, um die Verbindungseigenschaften des JMX-Servers anzuzeigen.

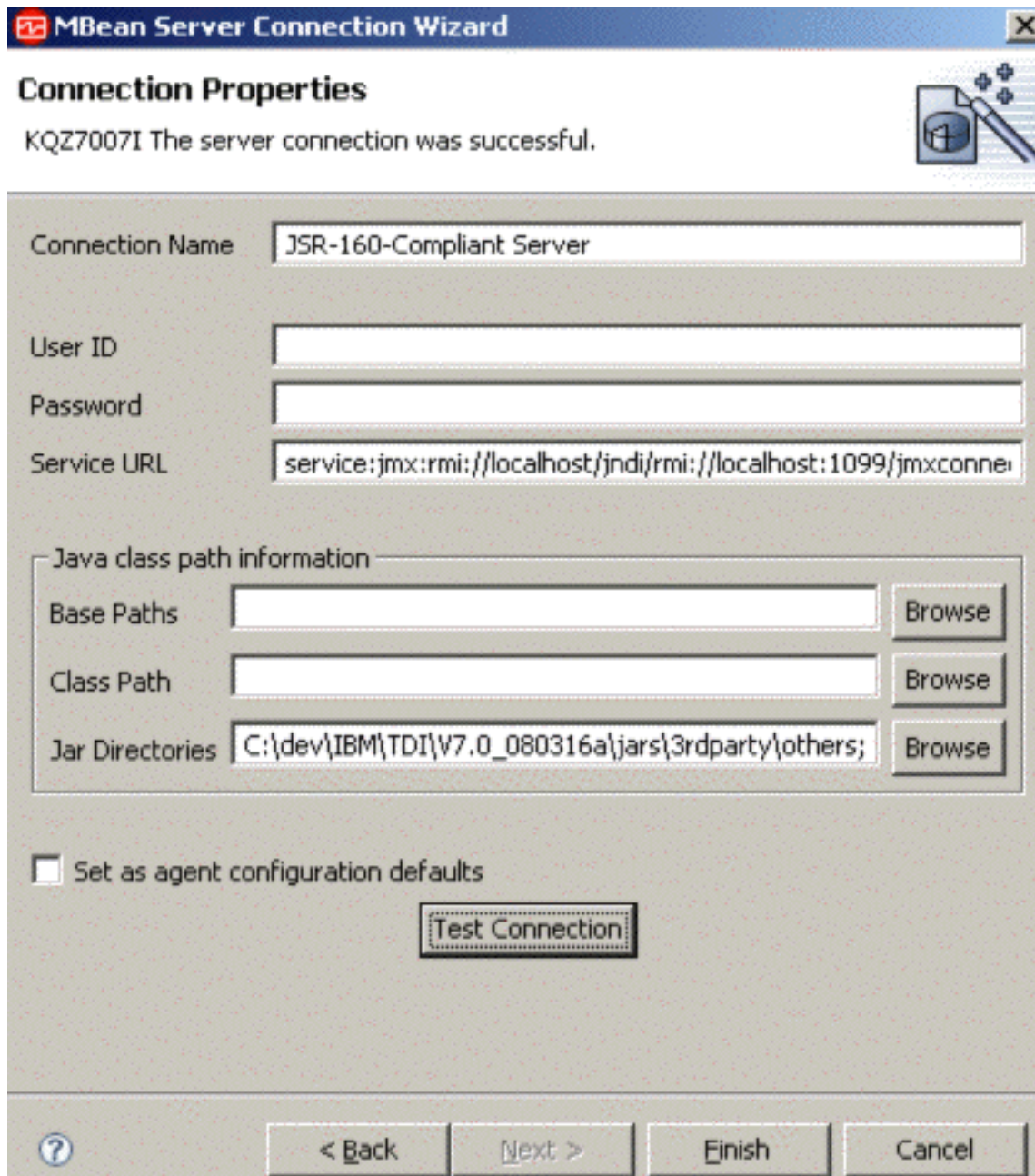


Abbildung 6. Assistent für Serververbindung

Damit eine Verbindung zum JMX-Service von IBM Security Directory Integrator erfolgreich aufgebaut werden kann, müssen Sie eine gültige URL für den JMX-Service eingeben (die Standard-URL des JMX-Service von IBM Security Directory Integrator lautet `service:jmx:rmi:///localhost/jndi/rmi:///localhost:1099/jmxconnector`) und die JAR-Abhängigkeiten konfigurieren, die für die erfolgreiche Erstellung von JMX-MBeans erforderlich sind (für die JMX-MBeans von IBM Security Directory Integrator benötigen Sie die JAR-Dateien in den Verzeichnissen `TDI-installationsverzeichnis\jars\3rdparty\IBM`, `TDI-installationsverzeichnis\jars\`

3rdparty\others und *TDI-installationsverzeichnis\jars\common*). Sie können diese Einstellungen testen, indem Sie auf die Schaltfläche **Verbindung testen** klicken. Falls die gesamte Konfiguration korrekt ist, wird die Nachricht "Die Serververbindung wurde erfolgreich hergestellt" ausgegeben.

Klicken Sie nach diesem Schritt auf **Fertig stellen**. Im Assistenten sollte nun die vorherige Konfigurationsanzeige erneut aufgerufen werden. Dieses Mal ist jedoch die Verbindung zum IBM Security Directory Integrator-JXM-Server hergestellt worden und es werden zusätzliche Informationen angezeigt:

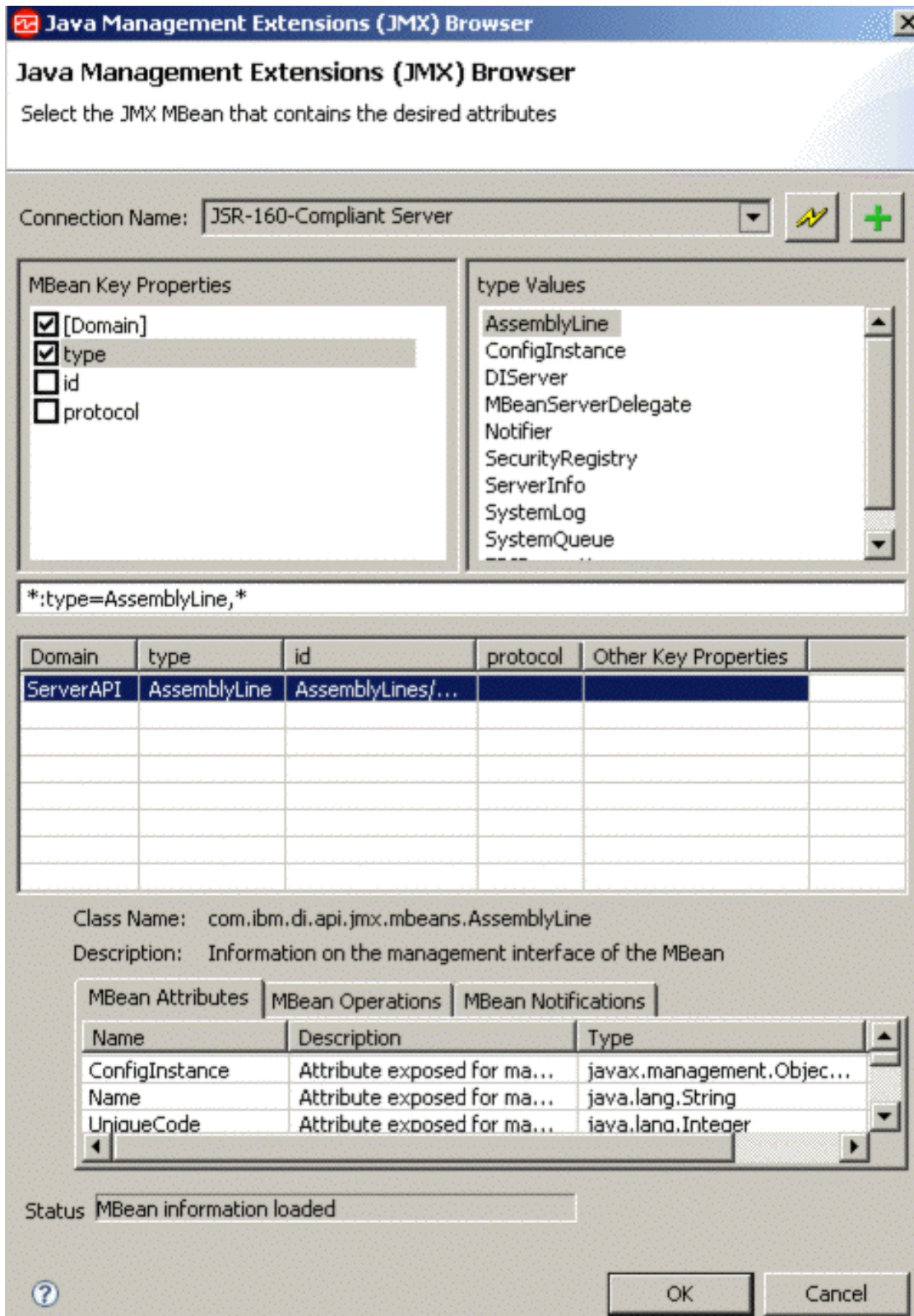


Abbildung 7. IBM Security Directory Integrator-Anzeige im Browser für Java Management Extensions (JMX)

Wählen Sie im Feld "MBean-Schlüsseigenschaften" die Option **type** und im Feld "Werte für type" den Eintrag **AssemblyLine** aus. Damit die MBean-Attribute angezeigt werden, müssen Sie in der darüber angezeigten Tabelle eine Zeile auswählen.

Im Beispiel gibt es nur eine einzige Zeile. Klicken Sie auf **OK** und dann auf **Fertig stellen**, um die Konfiguration dieser Datenquelle abzuschließen.

Erstellen Sie eine oder mehrere Datenquellen, bei dem Sie den Wert **ConfigInstance** für "type" verwenden. Gehen Sie hierzu genauso wie bei der Erstellung der Datenquelle mit dem Typ "AssemblyLine" vor. Diese beiden Datenquellen erfassen Informationen aus dem JMX-Server für aktive Fertigungslinien und gestartete Konfigurationsinstanzen.

Die dritte Datenquelle unterscheidet sich etwas von den beiden anderen Datenquellen. Es handelt sich um eine Art Listener, der für Benachrichtigungen (Ereignisse) empfangsbereit ist, die vom IBM Security Directory Integrator-JMX-Server gesendet werden. Um eine solche Datenquelle zu erstellen, müssen Sie nach dem Klicken auf die Schaltfläche **Neue Datenquelle...** nicht den JMX-Server durchsuchen, sondern als MBean-Muster lediglich die Zeichenfolge `*:type=Notifier,*` eingeben und auf die Schaltfläche **Fertig stellen** klicken. Daraufhin werden zwei Datenquellen (eine für den Benachrichtigungsteil und eine für den statischen MBean-Teil) erstellt. Da der statische Teil für diese Datenquelle nicht benötigt wird, muss er entfernt werden. Klicken Sie hierzu mit der rechten Maustaste und wählen Sie die Option **Datenquelle(n) entfernen** aus.

Nach Abschluss der obigen Schritte haben Sie drei Datenquellen erstellt:

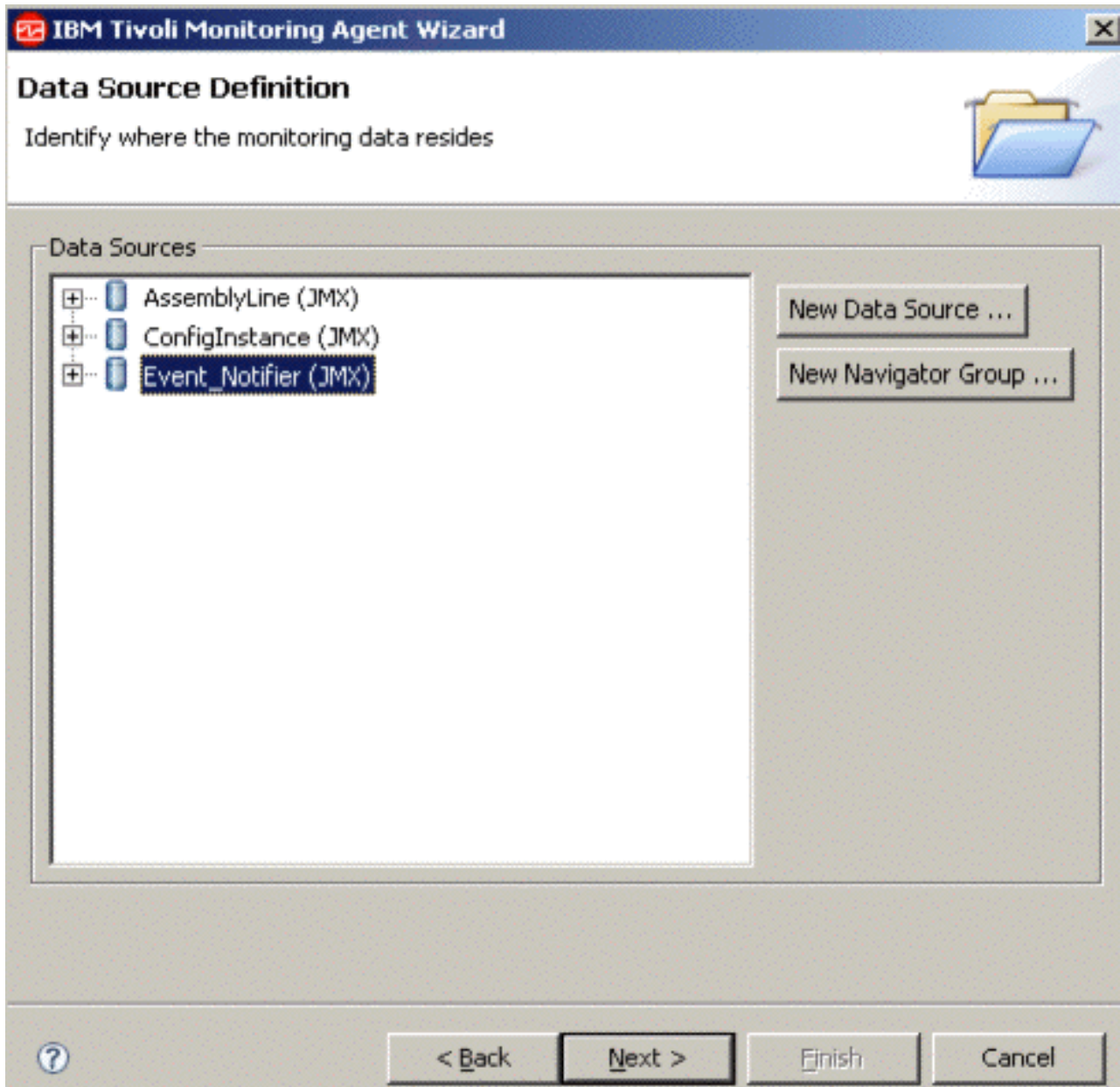


Abbildung 8. Assistent für IBM Tivoli Monitoring-Agenten nach abgeschlossener Datenquellendefinition

Erweitern Sie die Datenquelle "AssemblyLine" und doppelklicken Sie auf das Attribut **ConfigInstance**. Wählen Sie in der Konfiguration des Attributs "ConfigInstance" das Markierungsfeld **Schlüsselattribut** aus.

Erweitern Sie die Datenquelle "ConfigInstance" und doppelklicken Sie auf das Attribut **ConfigId**. Wählen Sie in der Konfiguration des Attributs "ConfigId" das Markierungsfeld **Schlüsselattribut** aus.

Klicken Sie auf **Weiter**, um die agentenweiten Optionen für JMX zu konfigurieren. Wählen Sie das Markierungsfeld "Attributgruppen und Aktionsbefehle für den JMX-Monitor einschließen" ab und wählen Sie unter den Auswahlmöglichkeiten für die Serverkonfiguration die Option **JSR-160-konformer Server** aus.

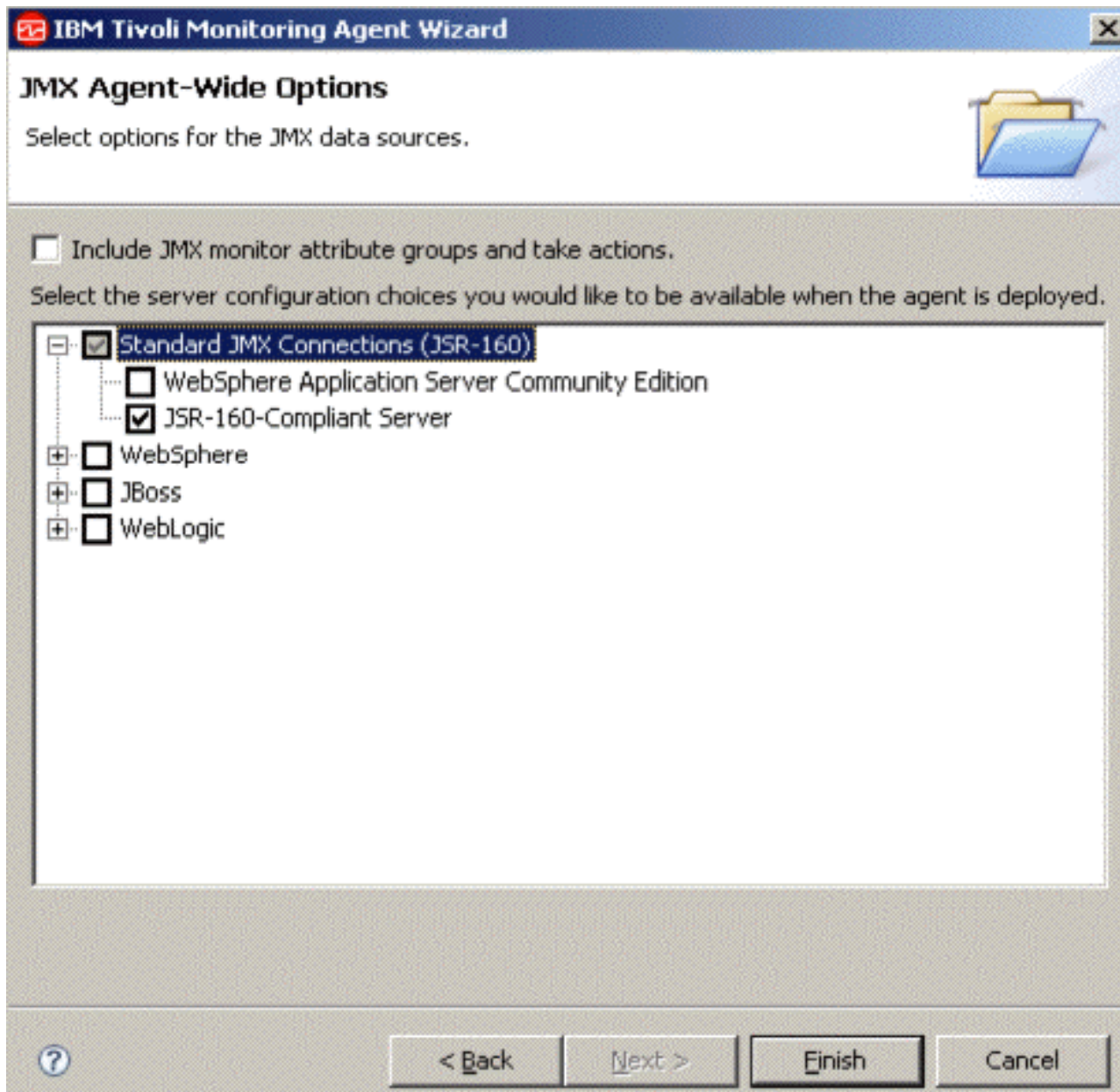


Abbildung 9. Agentenweite Optionen für JMX

Klicken Sie auf **Fertig stellen**, um die Erstellung des ITM-Agenten abzuschließen und den Agenten zu speichern.

ITM-Agenten generieren

Nach der erfolgreichen Erstellung eines ITM-Agenten können Sie ihn mit den nachstehenden Schritten generieren.

Nachdem die Konfiguration des ITM-Agenten erfolgreich erstellt wurde, muss sie generiert werden, damit sie in ITM implementiert werden kann. Wählen Sie im Menü "IBM Tivoli Monitoring Agent Editor" von ITM Agent Builder die Option **Agenten generieren** aus. Daraufhin wird der Assistent "Agenten generieren" aufgerufen. Dieser Assistent bietet mehrere Optionen für die Agentengenerierung. Falls Sie ITM und ITM Agent Builder auf einer einzigen Maschine ausführen, ist die Op-

tion **Agentendateien in ITM-Installation auf diesem System generieren** für Sie geeignet. Das einzige Feld, das Sie konfigurieren müssen, ist das ITM-Installationsverzeichnis. Klicken Sie auf **Fertig stellen**, um den ITM-Agenten zu generieren und in ITM zu implementieren. Dieser Vorgang kann einige Minuten dauern.

Anmerkung: Falls Sie für den Agenten eine andere Maschine verwenden wollen, können Sie eine andere Option für die Agentengenerierung verwenden, nämlich **Komprimierte Datei zur Agenteninstallation auf anderen Systemen erstellen**. Hierdurch wird ein Archiv generiert, das die ITM-Agenteninstallation enthält. Zur Installation eines solchen archivierten Agenten müssen Sie zuerst die Datei auf die Maschine kopieren, auf der ITM installiert ist. Extrahieren Sie die Dateien aus dem Archiv und starten Sie dann an der Eingabeaufforderung die Datei `InstallIRA.bat`. Verwenden Sie hierbei als Parameter den ITM-Installationsordner.

Falls ITM beispielsweise im Ordner `C:\IBM\ITM` installiert ist, lautet der Befehl wie folgt:

```
<agentenverzeichnis>:\>InstallIRA.bat C:\IBM\ITM
```

ITM-Agenten konfigurieren

Sobald der ITM-Agent implementiert wurde, müssen Sie ihn konfigurieren. Nachstehend finden Sie die Schritte und ein Beispiel für die Ausführung dieser Aufgabe.

Im Anschluss an die erfolgreiche Implementierung des Agenten (entweder mit einer Archivdatei oder der ITM Agent Builder-Option für die Implementierung auf derselben Maschine) muss der Agent in ITM konfiguriert werden. Starten Sie hierzu **Manage Tivoli Monitoring Enterprise Services**. In dieser Komponente können Sie alle ITM-Agenten verwalten:

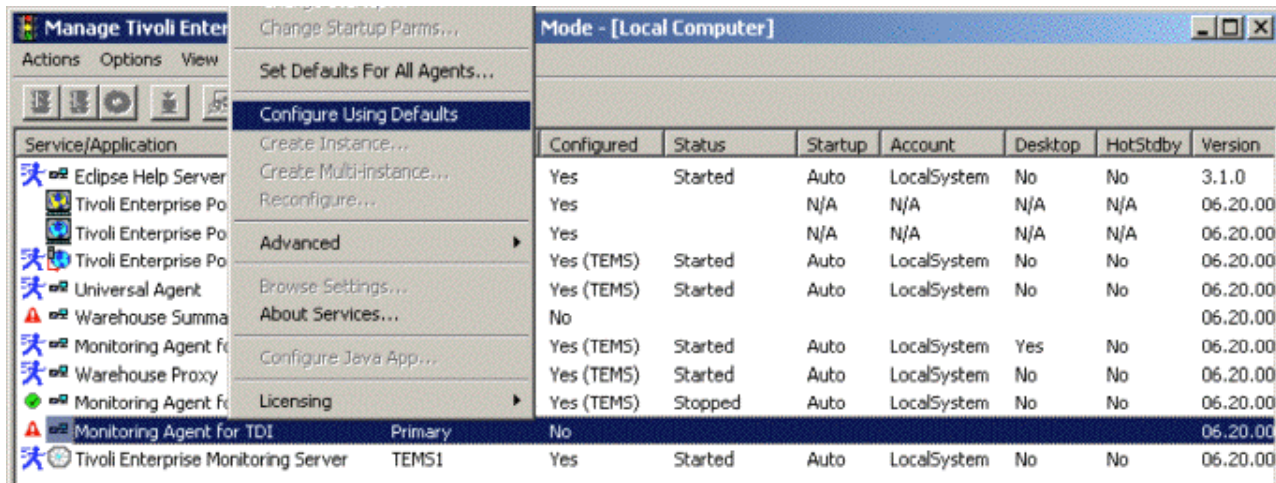


Abbildung 10. Manage Tivoli Monitoring Enterprise Services

Klicken Sie mit der rechten Maustaste auf den IBM Security Directory Integrator-Agenten und wählen Sie die Option **Mit Standardwerten konfigurieren** aus.

Im nächsten Konfigurationsfenster müssen Sie die JVM-Eigenschaften für den Agenten konfigurieren. Suchen Sie nach dem Java-Ausgangsverzeichnis, das Sie verwenden wollen. Die Ebene für den Protokolltrace ist standardmäßig auf "Fehler"

gesetzt. Sie können diese Angabe in eine höhere Ebene ändern, damit zusätzliche Informationen protokolliert werden. Klicken Sie nach Abschluss der Java-Konfiguration auf die Schaltfläche **Weiter**.

In der nächsten Konfigurationsanzeige werden Sie aufgefordert, die Eigenschaften für den JSR-160-konformen Server zu konfigurieren. Geben Sie daher den Benutzernamen, das Kennwort, die Service-URL und die Klassenpfadabhängigkeiten ein. Für das vorliegende Beispiel müssen die Service-URL und die JAR-Verzeichnisse wie bei der Erstellung des Agenten eingegeben werden, also die Service-URL `service:jmx:rmi:///localhost/jndi/rmi:///localhost:1099/jmxconnector` und die JAR-Verzeichnisse `tdi-installationsverzeichnis\jars\3rdparty\IBM, tdi-installationsverzeichnis\jars\3rdparty\others, tdi-installationsverzeichnis\jars\common`.

Klicken Sie auf **OK**, um die Konfiguration des Agenten abzuschließen.

Der IBM Security Directory Integrator-Agent ist nun einsatzbereit. Als Nächstes müssen Sie den Agenten starten. Sie können ihn aus dem Fenster "Manage Tivoli Enterprise Monitoring Services" heraus starten, indem Sie mit der rechten Maustaste auf den IBM Security Directory Integrator-Agenten klicken und die Option **Starten** auswählen. Wenn alle Schritte erfolgreich ausgeführt wurden, ist der IBM Security Directory Integrator-Agent nun aktiv.

IBM Security Directory Integrator-Daten überwachen

Sie können die Daten mit Tivoli Enterprise Portal (TEP) überwachen. Nähere Angaben hierzu finden Sie in den nachfolgend aufgeführten Schritten.

Zur Überwachung von Daten müssen Sie Tivoli Enterprise Portal (TEP) starten. Dieses Produkt ist in der ITM-Installation verfügbar. Die aktiven Agenten werden im TEP-Navigatorfenster angezeigt. Dies trifft auch für den IBM Security Directory Integrator-Agenten zu. Um die speziellen Datenquellen für die Überwachung anzuzeigen, müssen Sie den Agenten erweitern:

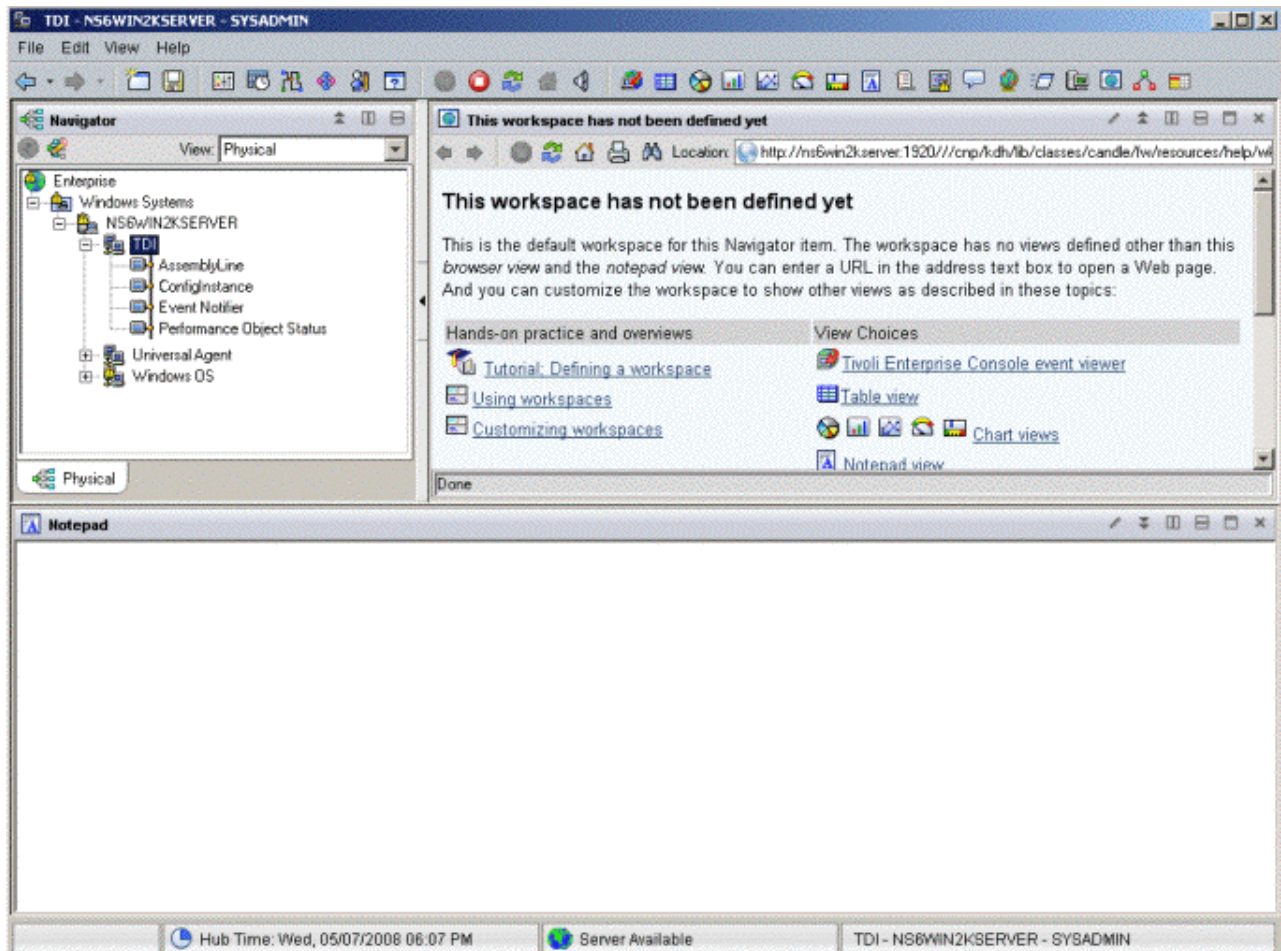


Abbildung 11. Assistenten von Tivoli Enterprise Portal (TEP)

Nach der Erweiterung werden die angepassten Datenquellen (AssemblyLine, ConfigInstance, Event Notifier) hier angezeigt. Wenn ein IBM Security Directory Integrator-Server ausgeführt wird, der eine aktive Fertigungslinie enthält, ist dies in der Berichtstabelle für die Datenquelle "AssemblyLine" erkennbar.

Anmerkung: Damit Daten in der Berichtstabelle der Benachrichtigungsfunktion angezeigt werden, muss der IBM Security Directory Integrator-Agent bereits ausgeführt werden, wenn eine IBM Security Directory Integrator-Benachrichtigung ausgelöst wird.

Dieser Browser kann unterschiedlich angepasst werden. Bei numerischen Daten kann beispielsweise eine besser lesbare Darstellung (z. B. in Form von Diagrammen) verwendet werden. Außerdem ist es möglich, das Layout der Tabellen zu ändern.

Diese Funktionalität ist nicht sehr komplex und in der ITM-Dokumentation gut beschrieben.

Da im vorliegenden Dokument nicht das gesamte ITM-Produkt bis ins letzte Detail beschrieben werden soll, sondern der Schwerpunkt auf der Verwendung liegt, die im Zusammenhang mit IBM Security Directory Integrator möglich ist, werden nachfolgend nur zwei kompliziertere Konzepte vorgestellt, nämlich die Definition von Schwellenwerten und die Erstellung von Links zwischen Tabellen.

Angaben zur übrigen Funktionalität können Sie der ITM-Dokumentation entnehmen.

Schwellenwerte definieren

Das vorliegende Beispiel und die Screenshots enthalten Informationen zum Arbeiten mit dem Schwellenwertmechanismus.

Um die Funktionsweise des Schwellenwertmechanismus zu veranschaulichen, wird das folgende einfache Beispiel erstellt: Es soll eine Warnung ausgegeben werden, wenn gegenwärtig mehr als eine Fertigungslinie ausgeführt wird.

Dieser Schwellenwert ist von den Daten abhängig, die von der Tabelle für "AssemblyLine" bereitgestellt werden. Zunächst müssen Sie eine Situation erstellen. Hierzu klicken Sie mit der rechten Maustaste auf eine Stelle in der Tabelle des Agenten und wählen Sie im Kontextmenü die Option **Situationen** aus.

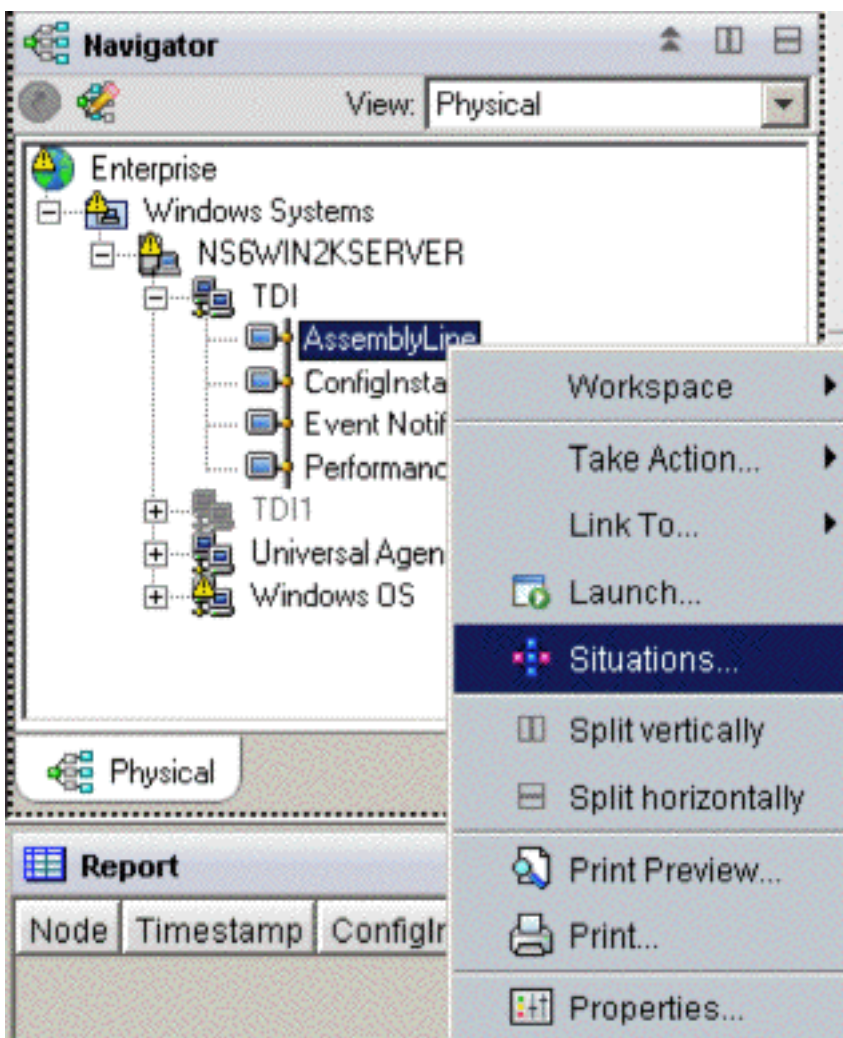


Abbildung 12. Kontextmenü "Situationen"

Klicken Sie in der linken oberen Ecke auf die Schaltfläche **Situation erstellen** und füllen Sie das angezeigte Formular aus:

Create Situation

Name: AssemblyLines

Description: AL warning

Monitored Application: TDI

Correlate Situations across Managed Systems

Situation name:

- 1) Must be 31 characters or less,
- 2) Must start with an alphabetic character (a-z, A-Z),
- 3) May contain any alphabetic, numeric (0-9) or underscore (_) character,
- 4) Must end with an alphabetic or numeric character.

OK Cancel Help

Abbildung 13. Formular "Situation erstellen"

Im Feld "Name" wird der Name eingegeben, der der Warnung zugeordnet sein soll. Hier handelt es sich lediglich um ein Fallbeispiel, aber in realen Situationen sollten Sie aussagekräftige Namen vergeben.

Anschließend wählen Sie aus, welche Tabellenattribute von der Situation verwendet werden sollen:

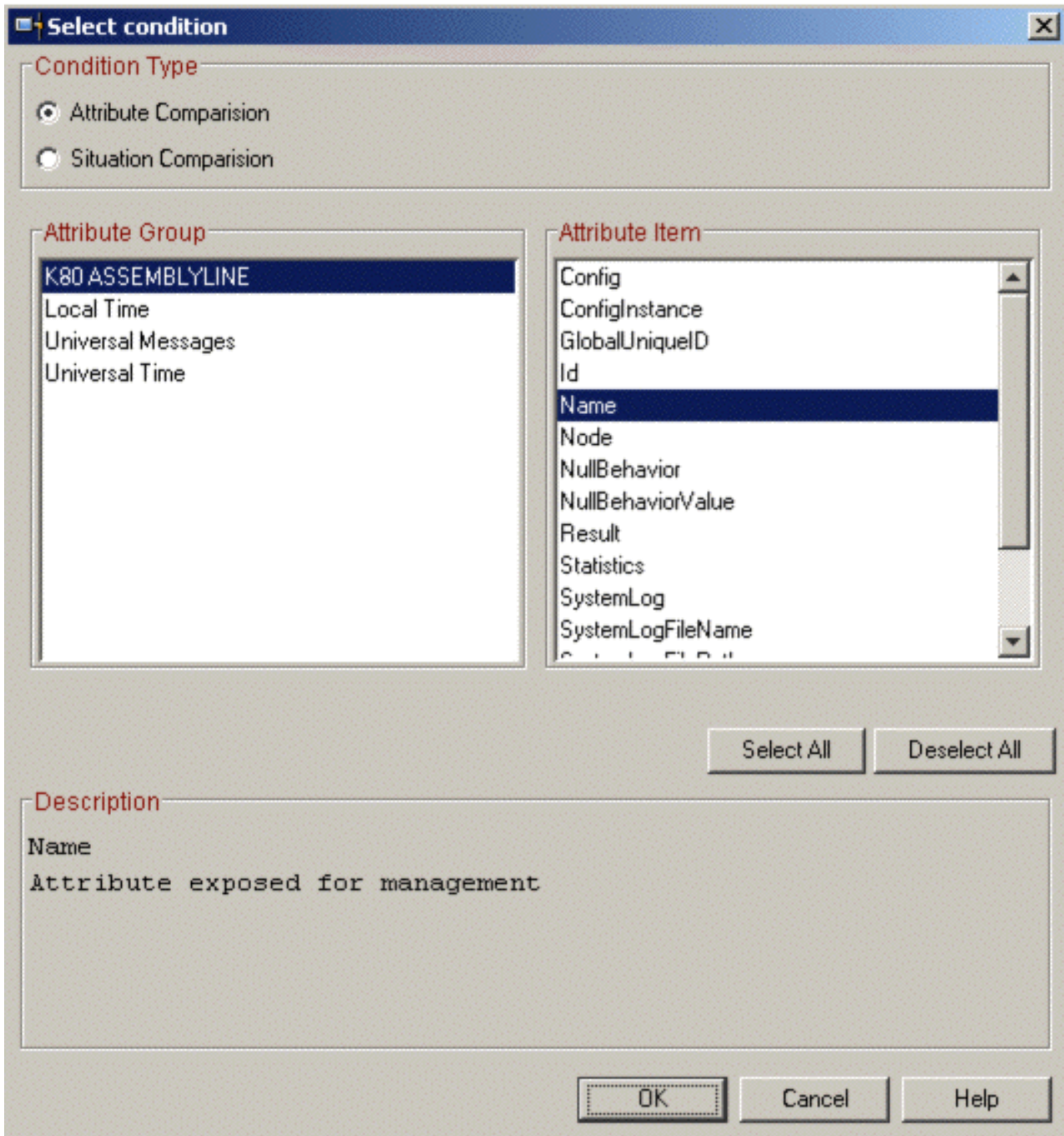
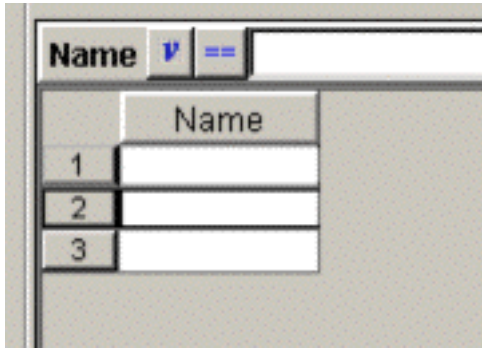


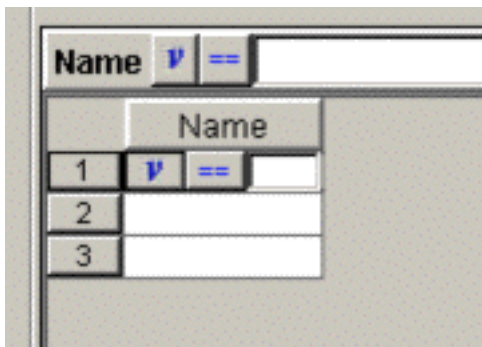
Abbildung 14. Anzeige "Bedingung auswählen" für Situation

Natürlich muss im vorliegenden Fall lediglich der Name der Fertigungslinie berücksichtigt werden, damit sie erkannt werden kann.

Klicken Sie auf eine Stelle in einer der Zellen, beispielsweise auf die Zelle in der Zeile 1:



Die Anzeige ändert sich wie folgt:

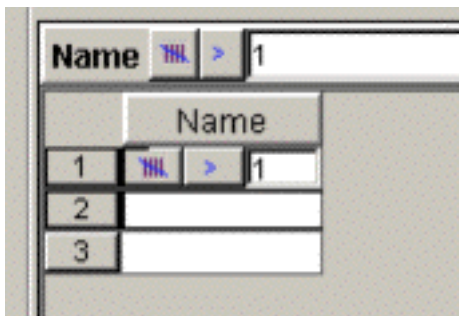


Klicken Sie auf die Schaltfläche *v* und ändern Sie sie in die Schaltfläche "Anzahl der Gruppenmitglieder".

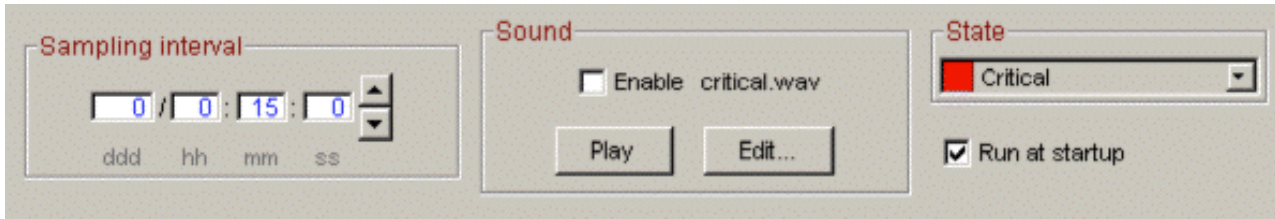
Klicken Sie auf die Schaltfläche "==" und ändern Sie sie in die Schaltfläche ">".

Setzen Sie den rechts verbleibenden Zellenbereich auf 1.

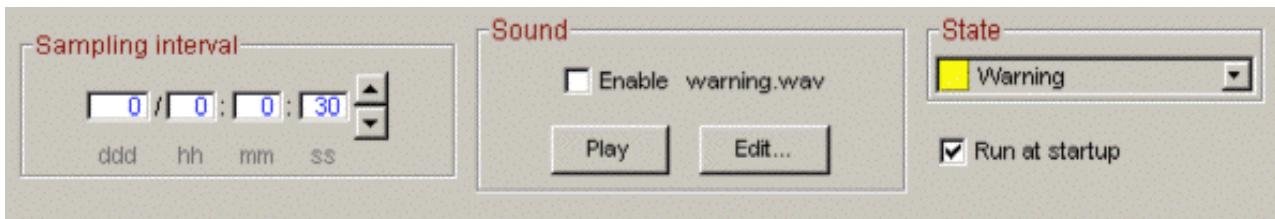
Hiermit haben Sie eine Bedingung für die Spalte **Name** konfiguriert, die mit "wahr" ausgewertet wird, wenn mehr als 1 Fertigungslinie aktiv ist.



Ändern Sie nun die folgenden Standardeinstellungen:



Verwenden Sie bei der Änderung die folgenden Werte:



Die Situation ist hiermit festgelegt. Daher können Sie in diesem Fenster die Option **Anwenden** auswählen, damit die Einstellungen wirksam werden.

Starten Sie den IBM Security Directory Integrator-Server und starten Sie gleichzeitig mindestens zwei Fertigungslinien, beispielsweise zwei HTTP-Server-Connectors, die an unterschiedlichen Ports empfangsbereit sind.

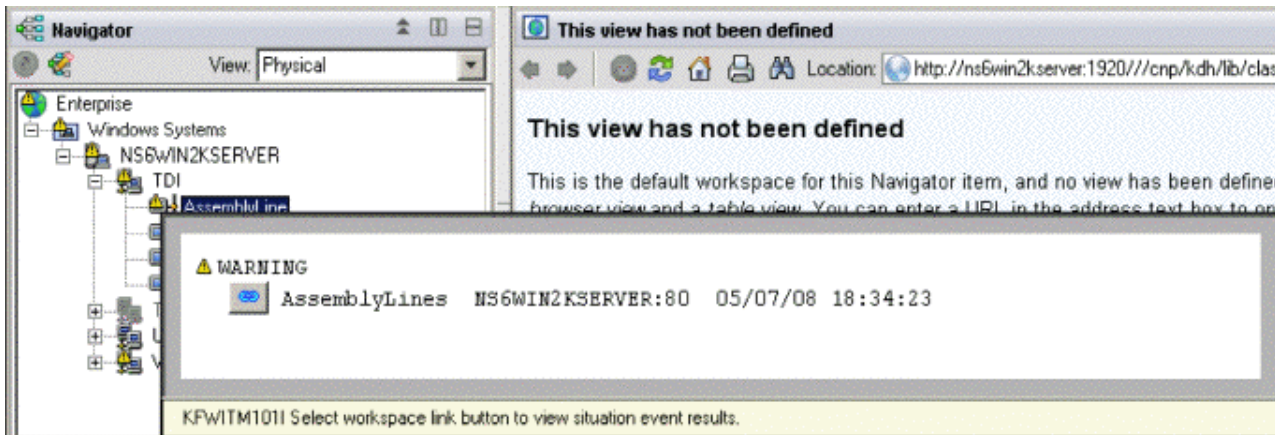


Abbildung 15. In ITM angezeigte Warnung

Das Fenster **Warnung** wird mit hervorgehobenem Warnungssymbol geöffnet.

Links zwischen Tabellen erstellen

Sie können Links zwischen Tabellen in ITM erstellen. Nachstehend erhalten Sie Informationen zum Zweck und zur Vorgehensweise beim Erstellen von Links.

Zweck von Links:

Über den Link können Sie die Daten filtern und direkt eine Teilmenge einer Tabelle anzeigen.

In ITM können zwischen verschiedenen Tabellen Links definiert werden. Wie das folgende Beispiel verdeutlicht, können diese Links auf einigen Kriterien Ihrer Wahl basieren. Wenn ein Link erstellt wird, können Sie automatisch eine Teilmenge der

Tabelle anzeigen, indem Sie den angegebenen Link auswählen. Im vorliegenden Beispiel wird ein von der Tabelle für "EventNotifier" ausgehender Link erstellt, der die gegenwärtig ausgeführte Fertigungslinie in der Tabelle für "AssemblyLine" anzeigt. Der Link ist in der Tabelle für "EventNotifier" nur für Datensätze mit dem Typ "di.al.start" verfügbar. Dieser Typ gibt an, dass eine Fertigungslinie gestartet wurde. Falls der Link ausgewählt wird und die Fertigungslinie noch aktiv ist, wird automatisch die Tabelle für "AssemblyLine" ausgewählt und in der Tabelle nur die entsprechende Fertigungslinie angezeigt. Falls die Ausführung der Fertigungslinie bereits beendet wurde, ist die angezeigte Tabelle leer.

Das folgende Beispiel zeigt eine Tabelle für "EventNotifier" mit einem definierten Link zur Tabelle für "AssemblyLine":

Node	Timestamp	Type	Source	Sequence Number	Time Stamp	Message
NS6WIN2KSERVER-80	05/07/08 18:33:59	di.al.stop	ServerAPI.type=Notifier,id=Notifier	6	1210174439618	AssemblyLine 'AssemblyLines/Server1'
NS6WIN2KSERVER-80	05/07/08 18:33:59	di.al.start	ServerAPI.type=Notifier,id=Notifier	5	1210174439558	AssemblyLine 'AssemblyLines/Server1'
NS6WIN2KSERVER-80	05/07/08 18:33:59	di.ci.start	ServerAPI.type=Notifier,id=Notifier	4	1210174439558	ConfigInstance 'runname' started.
NS6WIN2KSERVER-80	05/07/08 18:32:49	di.ci.start	ServerAPI.type=Notifier,id=Notifier	3	1210174369672	ConfigInstance 'sss' started.
NS6WIN2KSERVER-80	05/07/08 18:32:49	di.ci.start	ServerAPI.type=Notifier,id=Notifier	2	1210174369622	ConfigInstance 'C_dev IBM_TDI_V7.0_0_0_0'
NS6WIN2KSERVER-80	05/07/08 18:32:48	di.al.start	ServerAPI.type=Notifier,id=Notifier	1	1210174368120	AssemblyLine 'AssemblyLines/Server1'
NS6WIN2KSERVER-80	05/07/08 18:32:36	di.al.start	ServerAPI.type=Notifier,id=Notifier	0	1210174355842	AssemblyLine 'AssemblyLines/Server1'

Abbildung 16. Beispiel der Tabelle für "EventNotifier"

Die Tabelle enthält drei geladene Konfigurationen und drei gestartete Fertigungslinien (von denen eine gestoppt wurde). Nach Auswahl des Links **ToTheRunningAssemblyLine** wird die Fertigungslinie in der Tabelle für "AssemblyLine" angezeigt (andere Fertigungslinien werden in der Tabelle nicht angezeigt):

Name	Config	ConfigInstance	GlobalUniqueID	Id	Node
AssemblyLines/Server1	Server1	ServerAPI.type=ConfigInstance,id=C_dev IBM_TDI_V7.0_0_0_0...	11210174355752	AssemblyLines/Server1.1	NS6WIN2KSERVER-80

Abbildung 17. Beispiel für Ereignis

Erstellung von Links:

Mit den hier aufgeführten Schritten können Sie einen Link erstellen.

Zuerst muss in der Tabelle für "AssemblyLine" ein Schlüssel erstellt werden, auf den von der Tabelle für "EventNotifier" aus zugegriffen werden kann und der der Fertigungslinien-ID entspricht. Klicken Sie auf eine Stelle in der Tabelle für "AssemblyLine" und wählen Sie "Eigenschaften" aus.

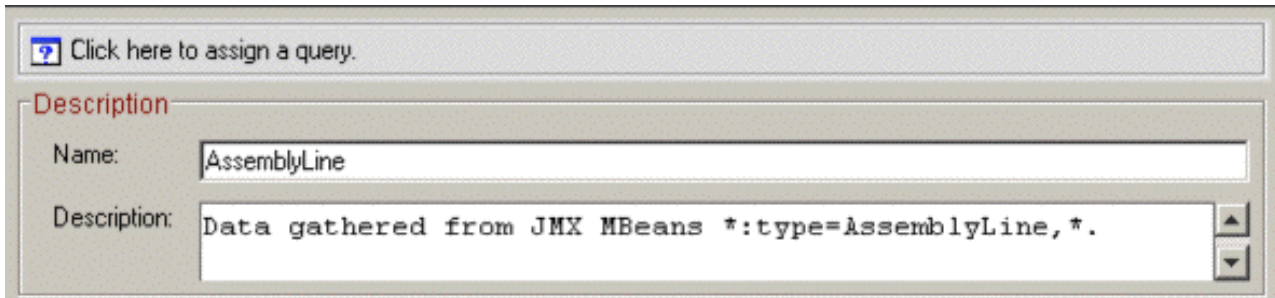


Abbildung 18. Eigenschaften für Fertigungslinie

Ordnen Sie im aufgerufenen Fenster eine neue Abfrage zu, indem Sie auf die entsprechende Schaltfläche klicken (z. B. auf **Zum Zuordnen der Abfrage hier klicken**).

Daraufhin wird der Abfrageeditor geöffnet. Dort müssen Sie eine weitere Abfrage definieren, da die vorhandene Abfrage statisch ist und nicht modifiziert werden kann. Sie werden aufgefordert, einen Namen für die Abfrage einzugeben (z. B. "AssemblyLine2").

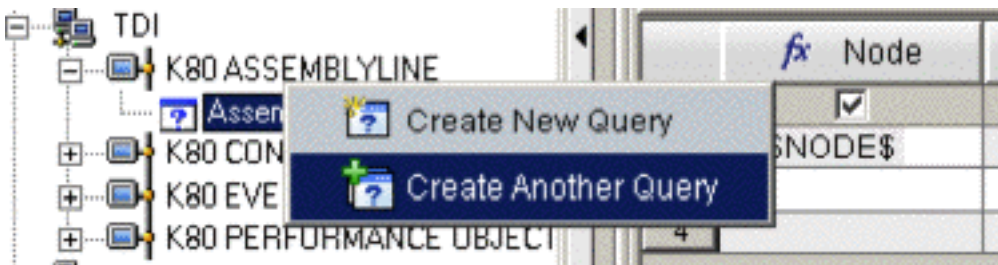


Abbildung 19. Auswahl für Abfrageerstellung

Wechseln Sie in die Spalte **Id** der Tabelle und geben Sie als Wert die Zeichenfolge \$keyid\$ ein.

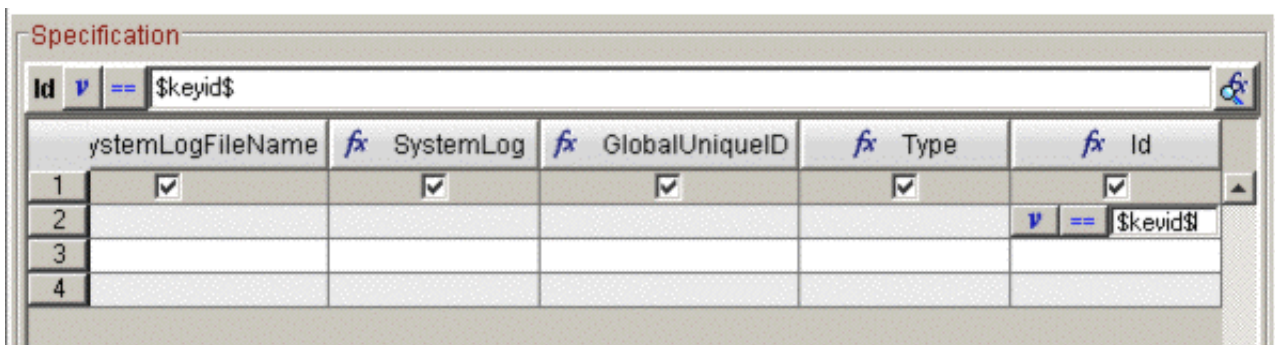


Abbildung 20. Abfrageeditor

Klicken Sie auf **OK** und wenden Sie die Änderungen im Eigenschaftfenster an. Klicken Sie anschließend auf die Schaltfläche **OK**.

Weitere Aktionen müssen in der Tabelle für "AssemblyLine" nicht ausgeführt werden.

Wechseln Sie zur Tabelle für "EventNotifier" (Sie werden aufgefordert, die geänderte Tabelle für "AssemblyLine" zu speichern; klicken Sie hierzu auf **Ja**).

Klicken Sie mit der rechten Maustaste auf eine Stelle in der ausgewählten Zeile der Tabelle für "EventNotifier" und wählen Sie die Optionen **Link zu...** -> **Linkassistent...** aus (vergewissern Sie sich, dass der Spaltentyp der ausgewählten Zeile "di.al.start" lautet).

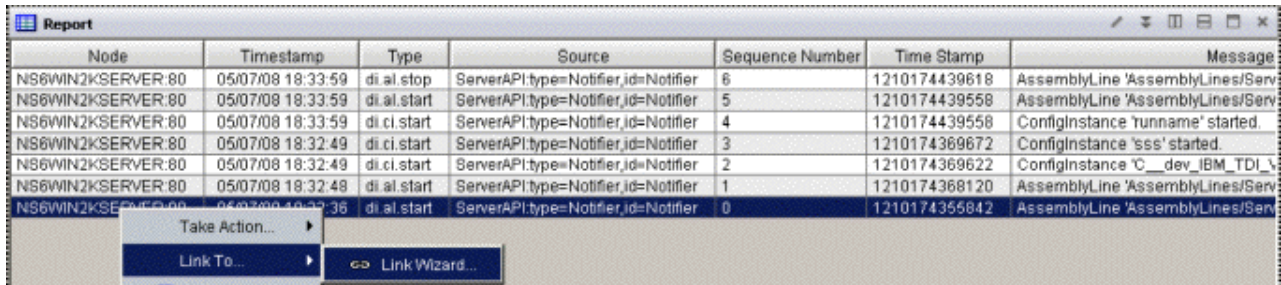


Abbildung 21. Auswahl "Linkassistent"

Nun wird der Linkassistent aufgerufen. Dort müssen Sie angeben, ob Sie einen neuen Link erstellen, einen vorhandenen Link modifizieren oder einen vorhandenen Link löschen wollen. Wählen Sie die Option **Neuen Link erstellen** aus und klicken Sie auf **Weiter**. In der nächsten Anzeige müssen Sie den Namen und die Beschreibung des Links eingeben. Im vorliegenden Beispiel werden der Name **ToTheRunningAssemblyLine** und die Beschreibung "Display the corresponding AssemblyLine in the AssemblyLine table" verwendet. Anschließend muss der Linktyp angegeben werden. Im Beispiel wird der Linktyp **Absolut** verwendet, da der Link zu einem angegebenen nicht dynamischen Arbeitsbereich in der Navigatoransicht erstellt wird. Fahren Sie nun mit dem nächsten Schritt fort, in dem Sie den Arbeitsbereich angeben müssen, zu dem der Link führt (Tabelle für "AssemblyLine").

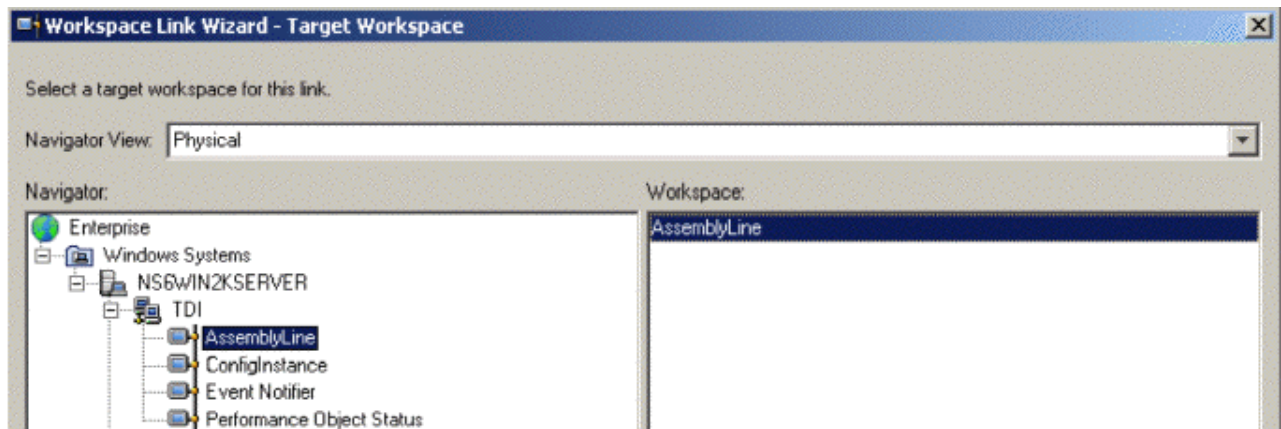


Abbildung 22. Linkassistent - Zielarbeitsbereich

Klicken Sie nach Auswahl des Arbeitsbereichs "AssemblyLine" auf **Weiter**. Es folgt nun der letzte Schritt, in dem Sie die Linkbedingungen erstellen müssen. Zur Erstellung des beabsichtigten Links werden zwei Parameter (*contextIsAvailable* und *keyid*) modifiziert.

Wählen Sie den Parameter **contextIsAvailable** aus und klicken Sie auf die Schaltfläche **Ausdruck ändern...** (oder doppelklicken Sie auf den Parameter). Daraufhin wird das Fenster "Ausdruckseditor" angezeigt. Löschen Sie den aktuellen Inhalt und klicken Sie auf die Schaltfläche **Symbol...** Wählen Sie unter den Symbolen das Attribut **Type** aus:

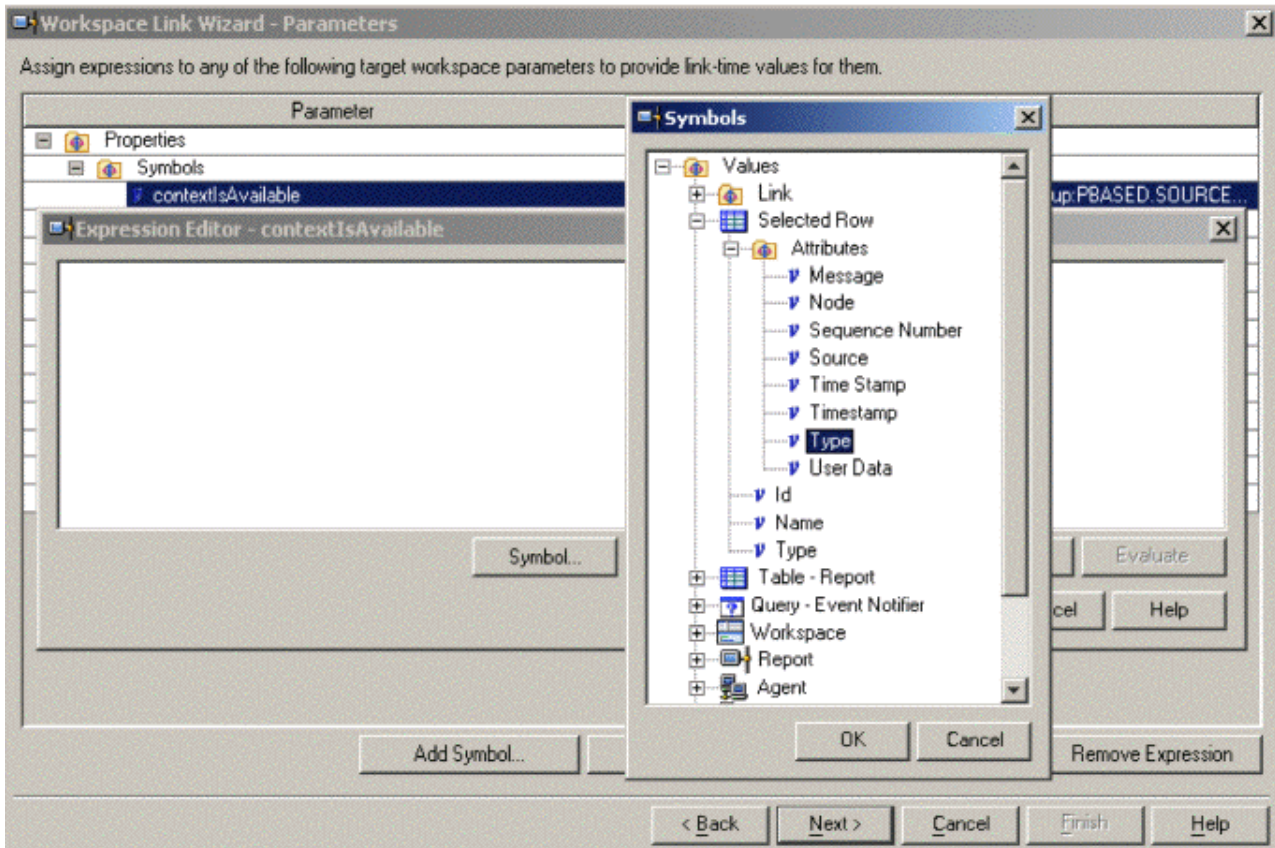


Abbildung 23. Linkassistent - Attribut "Type"

Klicken Sie auf **OK**, um zum Fenster "Ausdruckseditor" zurückzukehren. Fügen Sie den Ausdruck `== "di.al.start"` hinzu, um einen Bedingungsausdruck zu erstellen.

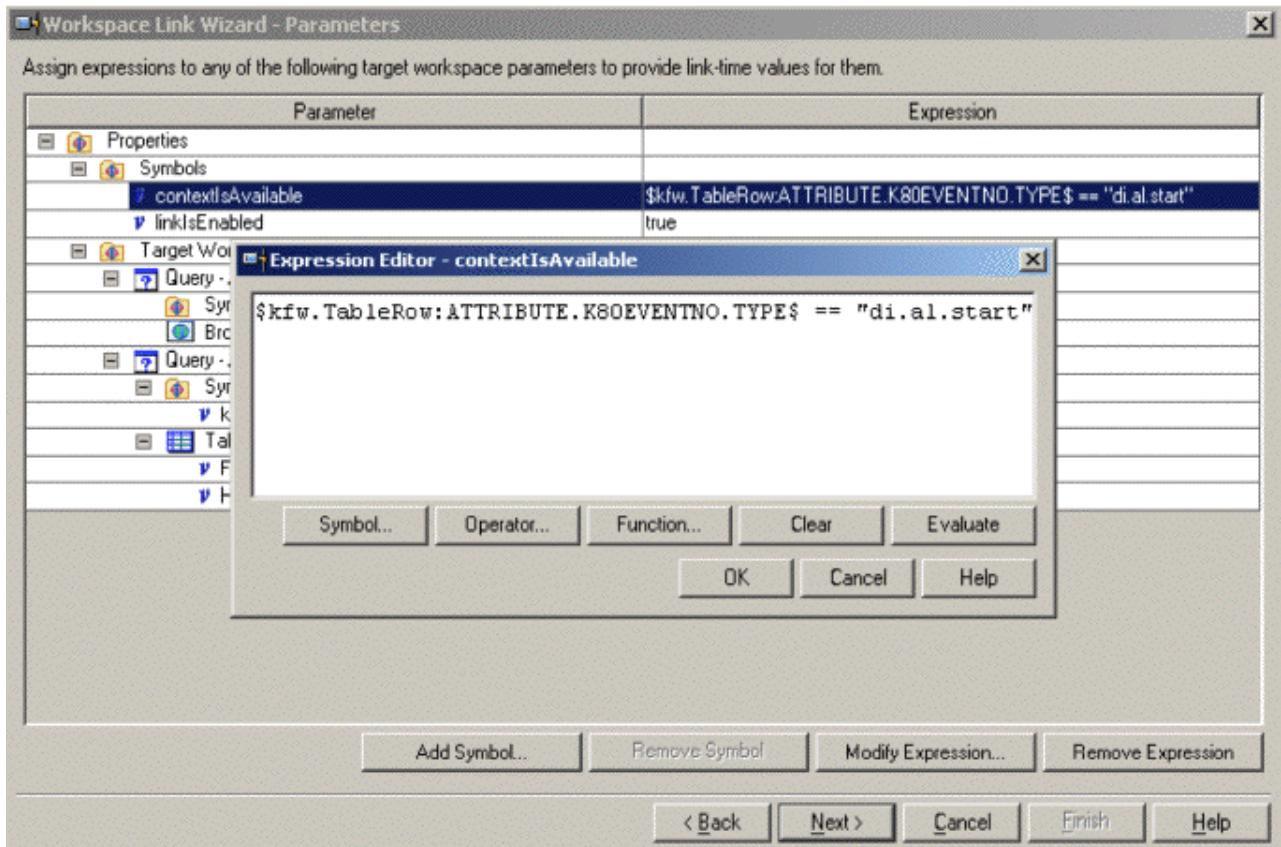


Abbildung 24. Linkassistent - Ausdruckseditor

Klicken Sie auf **OK**, um den Ausdruckswert zu bestätigen.

Öffnen Sie den Ausdruckseditor für das Symbol "keyid" in **Abfrage - AssemblyLine2** und fügen Sie das Symbol "User Data" hinzu.

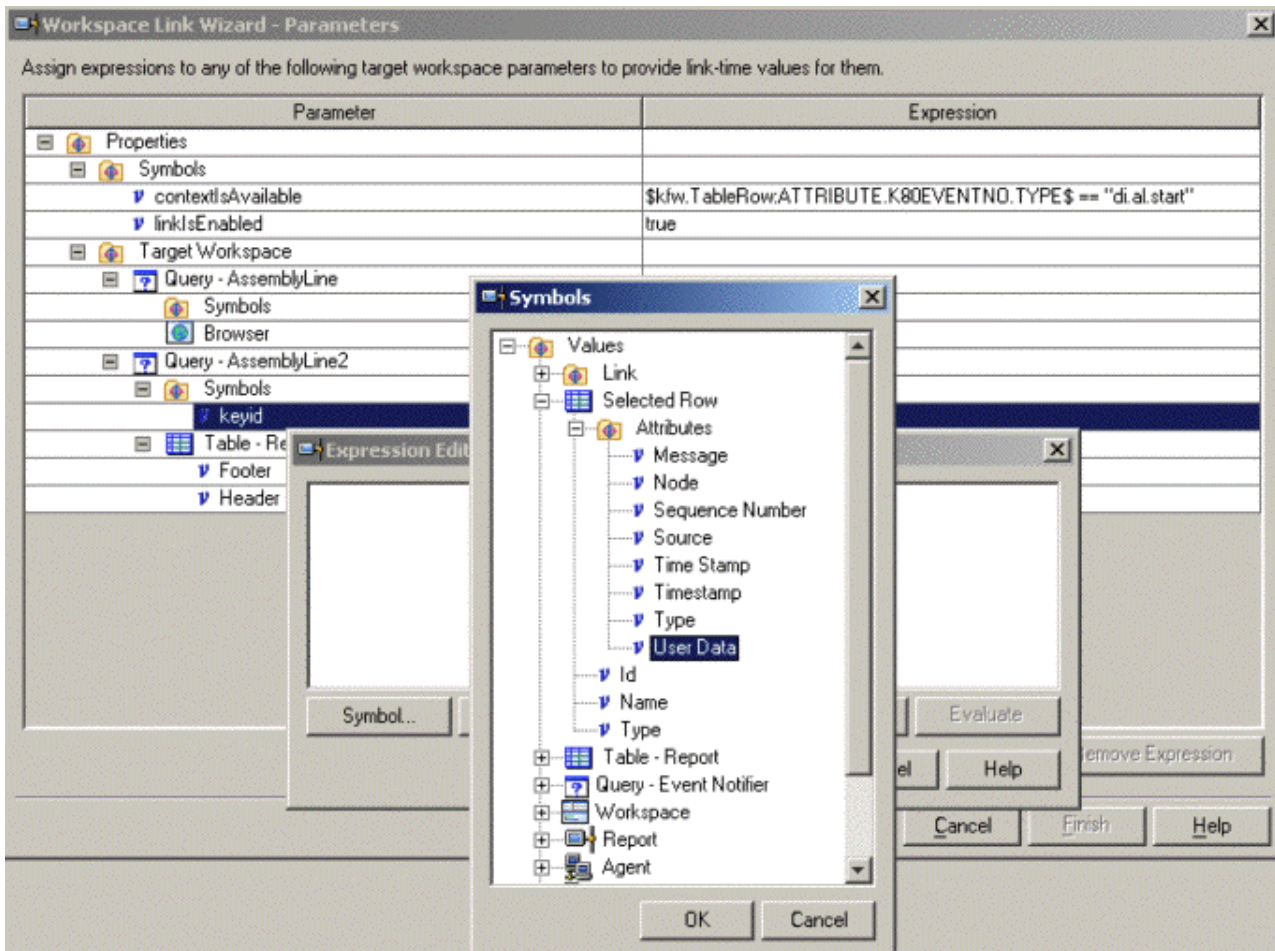


Abbildung 25. Linkassistent - Attribut "User Data"

Klicken Sie im Ausdruckseditor auf **OK** und klicken Sie im Linkassistenten auf **Weiter**. Daraufhin wird eine Zusammenfassung des erstellten Links angezeigt.

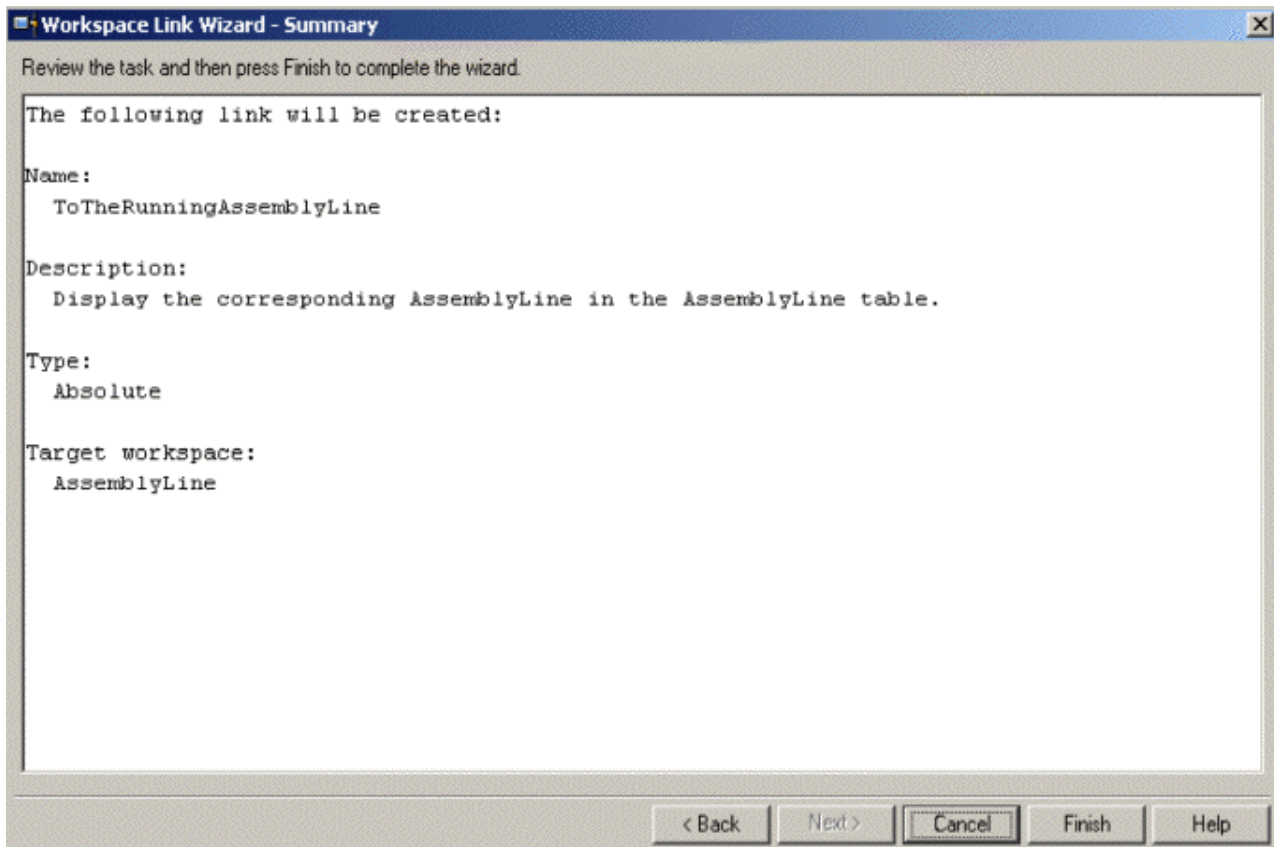


Abbildung 26. Linkassistent - Zusammenfassung

Klicken Sie auf **Fertigstellen**, um den Linkassistenten zu schließen. Nun können Sie die Schritte ausführen, die im Abschnitt „Zweck von Links“ auf Seite 412 beschrieben sind.

Angepasste Benachrichtigungen an ITM senden

Erstellen Sie mit dem hier aufgeführten Code Scripts zum Senden angepasster Benachrichtigungen.

Zusammen mit dem Beispiel wird eine Konfigurationsdatei ausgeliefert, die das Senden von angepassten Benachrichtigungen veranschaulicht. Die Datei befindet sich an der Position `tdi-installationsverzeichnis/examples/Tivoli_Monitoring/TDI_Monitored_by_ITM/ custom_notifications.xml`.

Zum Senden von angepassten Benachrichtigungen müssen Sie ein eigenes Script schreiben, das dies ausführt. Der folgende Code verdeutlicht dies:

```
session.sendCustomNotification(aType, aId, aData);
```

Dieser Codeteil sendet eine angepasste, benutzerdefinierte Benachrichtigung an alle registrierten Listener. Der Parameter **aType** ist der Benachrichtigungstyp. **aId** ist die Benachrichtigungs-ID. **aData** steht für angepasste Benutzerdaten. Bitte beachten Sie, dass dem Parameter "aType" automatisch das Präfix "user" vorangestellt wird. Dies bedeutet, dass eine gesendete Benachrichtigung des Typs **myType** als **user.myType** empfangen wird.

Einschränkungen

Sie können den erstellten Agenten nicht zur Verwaltung des IBM Security Directory Integrator-Servers verwenden.

So kann er beispielsweise Fertigungslinien nicht starten oder stoppen. Er kann nur zu Überwachungszwecken eingesetzt werden, auch wenn die JMX-Schicht des IBM Security Directory Integrator-Servers entsprechende Methoden zur Verfügung stellt.

IBM SDI mit OMNIBus überwachen

Unter dem hier aufgeführten Link erhalten Sie Informationen zur Überwachung von IBM Security Directory Integrator mit OMNIBus.

Einführung

Weiterführende Informationen zu OMNIBus enthält der Abschnitt über den EIF-Connector in der Veröffentlichung *Referenzinformationen*.

Eigenschaftendatei für EIF-Testmonitor konfigurieren

Sie können den Port einstellen, an dem der EIF-Testmonitor empfangsbereit ist. Nachstehend erhalten Sie Informationen zur Ausführung dieser Task.

Um zu gewährleisten, dass der EIF-Testmonitor am erwarteten Port empfangsbereit ist, können Sie den Port manuell festlegen.

Hierzu legen Sie in der Datei `$OMNIHOME/probes/<arch>/tivoli_eif.props` als Wert der Eigenschaft **PortNumber** die Nummer des Ports fest, an dem der EIF-Testmonitor empfangsbereit ist.

Standardmäßig ist der EIF-Testmonitor so konfiguriert, dass der Service nach einer Inaktivitätsdauer (= Zeit ohne einen Empfang von Ereignissen) von 600 Sekunden gestoppt wird. Sie können das Zeitlimit auf einen unendlichen Zeitraum setzen, indem Sie für die Eigenschaft **Inactivity** den Wert 0 angeben.

Ihre EIF-Eigenschaftendatei sollte folgendermaßen aussehen:

```
# BufferEvents           : "YES"
# HandleMalformedAlarms : "true"
# EIFCacheFile          : '$OMNIHOME/var/tivoli_eif.cache' (Unix)
# EIFCacheFile          : '%OMNIHOME%\var\tivoli_eif.cache' (Windows)
# EventCopies           : 1
# Inactivity           : 0
# MaxEventQueueSize     : 10000
# PortMapper            : "false"
# PortMapperNumber      : 100033057
# PortNumber          : 9998
# Retry                 : "false"
# StreamCapture         : "false"
# StreamCaptureFile     : '$OMNIHOME/var/tivoli_eif.stream' (Unix)
# StreamCaptureFile     : '%OMNIHOME%\var\tivoli_eif.stream' (Windows)
```

Sollten Sie sich jedoch dafür entscheiden, die Eigenschaftendatei des EIF-Testmonitors nicht zu modifizieren, müssen Sie bedenken, dass der EIF-Testmonitor (laut OMNIBus-Dokumentation) standardmäßig an Port 9999 für Ereignisse empfangsbereit ist.

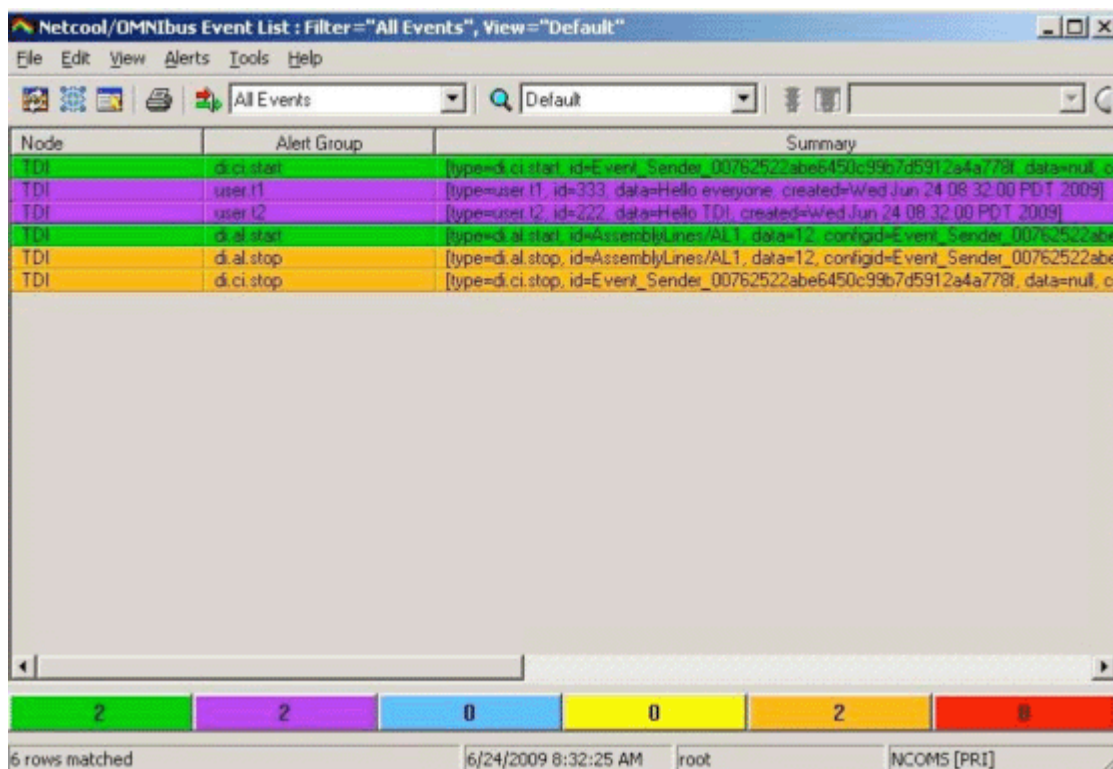
Bewertung der Ereignisse bestimmen

Modifizieren Sie die EIF-Regeldatei, um die Bewertung der Ereignisse zu bestimmen.

Um die Bewertung für die Ereignisse zu bestimmen, müssen Sie einige wenige Modifizierungen an der EIF-Regeldatei vornehmen. Die folgende Kurzbeschreibung erläutert, wie Sie die Bewertung verwalten. Zu diesem Zweck werden **Startereignisse** mit niedriger Bewertung und **Stoppereignisse** mit hoher Bewertung definiert. Anschließend können Sie den folgenden Code in der Regeldatei eingeben:

```
if( regmatch($ClassName, "^.*\.start$") )
{
  @Severity = 0
}
if( regmatch($ClassName, "^.*\.stop$") )
{
  @Severity = 4
}
```

Bitte beachten Sie, dass die Standardbewertung bei angepassten Benachrichtigungen auf 1 gesetzt ist. Dies bewirkt Folgendes:



Node	Alert Group	Summary
TDI	di.ci.start	[type=di.ci.start, id=Event_Sender_00762522abe6450c99b7d5912a4a778f, data=null, ci]
TDI	user.t1	[type=user.t1, id=333, data=Hello everyone, created=Wed Jun 24 08 32 00 PDT 2009]
TDI	user.t2	[type=user.t2, id=222, data=Hello TDI, created=Wed Jun 24 08 32 00 PDT 2009]
TDI	di.al.start	[type=di.al.start, id=AssemblyLines/AL1, data=12, configid=Event_Sender_00762522abe6450c99b7d5912a4a778f, data=null, ci]
TDI	di.al.stop	[type=di.al.stop, id=AssemblyLines/AL1, data=12, configid=Event_Sender_00762522abe6450c99b7d5912a4a778f, data=null, ci]
TDI	di.ci.stop	[type=di.ci.stop, id=Event_Sender_00762522abe6450c99b7d5912a4a778f, data=null, ci]

2 2 0 0 2 0

6 rows matched 6/24/2009 8:32:25 AM root NCOMS [PRI]

Abbildung 27. OMNIBus-Ereignisliste

Mit der Datei "EventPropertyFile.properties" arbeiten

Sie können verschiedene Dinge mit der Datei "EventPropertyFile.properties" tun. Nachstehend erhalten Sie Informationen dazu.

Die Datei EventPropertyFile.properties stellt eine Standardgruppe von Ereignissen bereit, die vom Serverbenachrichtigungsconnector empfangen werden können. Dies versetzt den Benutzer in die Lage, die Fertigungslinie allein über die Eigenschaftendatei zu konfigurieren. Die Eigenschaftendatei besitzt die folgende Struktur:

Schlüssel=Wert

Der *Schlüssel* bestimmt den Typ des Ereignisses, der *Wert* bestimmt, ob dieses Ereignis empfangen wird. Daher werden hauptsächlich die Werte *true* und *false* verwendet. Der Schlüssel *event.customNotifications* erwartet jedoch keinen booleschen Wert. Sein Wert muss aus den Namen der angepassten Ereignisse bestehen, die empfangen werden. Ausführlichere Informationen zu angepassten Benachrichtigungen finden Sie im Abschnitt "Angepasste Benachrichtigungen an OMNIbus senden". Zur Vermeidung von Missverständnissen sollten noch einige andere Aspekte berücksichtigt werden. Das folgende Diagramm gibt einen deutlicheren Aufschluss über die Standardgruppe der Ereignisse:

```
->event.all
|
|-->event.ci.all
|   |-->event.ci.start
|   |-->event.ci.stop
|-->event.ci.fileUpdated
|-->event.al.all
|   |-->event.al.start
|   |-->event.al.stop
|-->event.server.stop
|-->event.hasCustomNotofications
|   |-->event.customNotifications
```

Wie das obige Diagramm zeigt, enthalten einige Ereignisse Unterereignisse. Falls Sie ein Ereignis aktivieren, werden alle Unterereignisse empfangen, und zwar ungeachtet dessen, ob sie auf *true* oder *false* gesetzt sind. Dies bedeutet, dass bei der Angabe

```
event.ci.all=true
event.ci.stop=false
```

das Ereignis "event.ci.stop" empfangen wird, obwohl es auf *false* gesetzt ist. Anders ausgedrückt überschreibt das Ereignis "event.ci.all" seine Unterereignisse. Falls Sie jedoch die Angabe

```
event.ci.all=false
event.ci.stop=true
event.ci.start=false
```

verwenden, wird ausschließlich das Ereignis "event.ci.stop" empfangen.

Die Eigenschaftendatei ist standardmäßig so definiert, dass alle IBM Security Directory Integrator-Serverbenachrichtigungen bereitgestellt werden. Falls Sie diese Gruppe von Ereignissen modifizieren wollen, müssen Sie die booleschen Werte in *true* ändern (falls das entsprechende Ereignis empfangen werden soll) oder aber in *false* (falls das Ereignis nicht empfangen werden soll). Für den Fall, dass Sie beispielsweise alle Ereignisse empfangen wollen, die Benachrichtigungen über den Start einiger Komponenten senden, muss Ihre Eigenschaftendatei wie folgt aussehen:

Properties

Add property Download Upload

Editor Connector

Search: Hide non-local properties

Name	Protected	Local Value	Server Value
event.al.all	false	false	false
event.al.start	false	true	true
event.al.stop	false	false	false
event.all	false	false	false
event.ci.all	false	false	false
event.ci.fileUpdated	false	false	false
event.ci.start	false	true	true
event.ci.stop	false	false	false
event.customNotifications	false		
event.hasCustomNotifications	false	false	false
event.server.stop	false	false	false

Abbildung 28. OMNibus-Eigenschaften

Informationen zum Arbeiten mit der Eigenschaftendatei für den Empfang von angepassten Benachrichtigungen können Sie im folgenden Abschnitt „Angepasste Benachrichtigungen an OMNibus senden“ nachlesen.

Angepasste Benachrichtigungen an OMNibus senden

Sie können die an OMNibus zu sendenden Benachrichtigungen anpassen. Führen Sie die nachstehend beschriebenen Schritte aus, um diese Task auszuführen.

Damit angepasste Benachrichtigungen empfangen werden können, müssen Sie die Eigenschaft `event.hasCustomNotofications` auf `true` setzen. Anschließend müssen Sie die Gruppe der Ereignisse angeben, die empfangen werden sollen. Bitte beachten Sie, dass alle von IBM Security Directory Integrator gesendeten angepassten Ereignisse mit dem Präfix "user." versehen werden. Dies bedeutet, dass Sie Folgendes festlegen müssen, wenn Sie ein angepasstes Ereignis vom Typ `myType` senden wollen:

```
event.customNotifications=user.myType
```

Um mehrere angepasste Ereignisse anzugeben, verwenden Sie zwischen den Ereignissen ein Semikolon (;) als Trennzeichen. Anhand der folgenden Beispielsituation soll dies verdeutlicht werden. Sie wollen alle angepassten Benachrichtigungen des Typs "user.myType1", "user.myType2" und "user.myType3" empfangen. Ihre Eigenschaftendatei sollte - in einem Texteditor geöffnet - in diesem Fall folgendermaßen aussehen:

```
##Determine if Server Shutdown events are received
event.server.stop=false
##Determine what Custom Notification events are received
##This property is used only if event.hasCustomNotofications is enabled
##Note that all custom notifications are prefixed with "user."
event.customNotifications=user.myType1;user.myType2;user.myType3
##Determine if Custom Notification events are received
event.hasCustomNotifications=true
```

Um angepasste Ereignisse jedes beliebigen Typs zu empfangen, muss die Eigenschaft `event.customNotifications` auf den Wert "*" gesetzt sein. Dies bedeutet, dass der Typ der angepassten Ereignisse, für die der Connector empfangsbereit ist, nicht angegeben wird und daher jedes erkannte angepasste Ereignis verarbeitet wird. Das ITM-Beispiel stellt eine Konfiguration bereit, die angepasste Benachrichtigungen senden kann. Sie kann auch zum Senden von angepassten Benachrichtigungen an OMNIBus verwendet werden. Weitere Informationen zu angepassten Benachrichtigungen finden Sie unter „Angepasste Benachrichtigungen an ITM senden“ auf Seite 419.

Anhang C. Eingabehilfefunktionen in IBM Security Directory Integrator

Eingabehilfefunktionen unterstützen Benutzer mit Behinderungen wie z. B. eingeschränkter Beweglichkeit oder eingeschränktem Sehvermögen beim erfolgreichen Einsatz von Informationstechnologieprodukten.

Eingabehilfefunktionen

Im Folgenden sind die wichtigsten Eingabehilfen aufgeführt, die in IBM Security Directory Integrator zur Verfügung stehen:

- Ausschließliche Bedienung über die Tastatur
- Schnittstellen, die verbreitet von Sprachausgabeprogrammen verwendet werden
- Über den Tastsinn wahrnehmbare Tasten, die jedoch nicht durch bloßes Berühren aktiviert werden
- Industriestandardeinheiten für Ports und Connectors
- Alternative Eingabe- und Ausgabeeinheiten als Zubehör

Die Produktdokumentation zu IBM Security Directory Integrator und die zugehörige Referenzliteratur sind für die behindertengerechte Bedienung geeignet. Die Eingabehilfefunktionen der Produktdokumentation sind unter folgender Adresse beschrieben: http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.iehsc.doc/iehs34_accessibility.html.

Navigation über die Tastatur

Dieses Produkt verwendet die Standardnavigationstasten von Microsoft Windows für die Ausführung allgemeiner Aktionen unter Windows, wie zum Beispiel das Zugreifen auf das Menü 'Datei' und auf die Aktionen zum Kopieren, Einfügen und Löschen. Für Aktionen, die spezifisch sind, werden Direktaufrufe über die Tastatur verwendet. Direktaufrufe über die Tastatur wurden für alle Aktionen bereitgestellt, wo dies erforderlich ist.

Schnittstelleninformationen

Die Eingabehilfefunktionen der Benutzerschnittstelle und der Dokumentation umfassen Folgendes:

- Schritte zum Ändern von Schriftarten, Farben und Kontrasteinstellungen im Konfigurationseditor:
 1. Geben Sie **Alt-W** ein, um auf das Menü **Fenster** des Konfigurationseditors zuzugreifen. Wählen Sie mithilfe des Abwärtspfeils die Option **Benutzervorgaben...** aus und drücken Sie die Eingabetaste.
 2. Wählen Sie auf der Registerkarte **Darstellung** die Einstellungen für **Farben und Schriftarten** aus, um die Schriftarten für beliebige Funktionsbereiche im Konfigurationseditor zu ändern.
 3. Wählen Sie unter **Ordner für Sicht und Editor** die Farben für den Konfigurationseditor aus. Bei der Auswahl der Farben können Sie auch den Kontrast ändern.
- Schritte zum Anpassen von Direktaufrufen über die Tastatur, die für IBM Security Directory Integrator spezifisch sind:

1. Geben Sie Alt-W ein, um auf das Menü **Fenster** des Konfigurationseditors zuzugreifen. Wählen Sie mithilfe des Abwärtspfeils die Option **Benutzervorgaben...** aus.
2. Wählen Sie mithilfe des Abwärtspfeils die Kategorie 'Allgemein' aus, öffnen Sie sie mit dem Rechtspfeil und suchen Sie mithilfe des Abwärtspfeils den Eintrag **Tasten**.
Unter dem Auswahlelement **Schema** befindet sich ein Feld, das den Eintrag "Filtertext eingeben" enthält. Geben Sie in das Feld für den Filtertext security directory integrator ein. Nun werden alle für IBM Security Directory Integrator spezifischen Direktaufrufe über die Tastatur angezeigt.
3. Ordnen Sie den von Ihnen gewünschten IBM Security Directory Integrator-Befehlen eine Tastenbelegung zu.
4. Klicken Sie auf **Anwenden**, um die Änderung zu speichern.

Der Konfigurationseditor ist eine spezielle Instanz einer Eclipse-Workbench. Ausführliche Informationen zu Eingabehilfefunktionen von Anwendungen, die mit Eclipse erstellt wurden, finden Sie unter <http://help.eclipse.org/help33/topic/org.eclipse.platform.doc.user/concepts/accessibility/accessmain.htm>.

- Die Produktdokumentation und die zugehörigen Veröffentlichungen sind für die behindertengerechte Bedienung über das JAWS-Sprachausgabeprogramm und den IBM Home Page Reader aktiviert. Alle Funktionen der Dokumentation können sowohl mithilfe der Maus als auch mithilfe der Tastatur ausgeführt werden.

Software anderer Anbieter

IBM Security Directory Integrator enthält bestimmte Software anderer Anbieter, die nicht durch die IBM Lizenzvereinbarung abgedeckt ist. IBM stellt keine Darstellung der Eingabehilfen dieser Produkte zur Verfügung. Informationen zu den Eingabehilfen dieser Produkte erhalten Sie vom entsprechenden Anbieter.

Das Installationsprogramm verwendet die Installationstechnologie von InstallAnywhere 2012 SP1.

Zugehörige Informationen zu Eingabehilfen

Als Alternative zur Online-Produktdokumentation steht eine Softcopy-Dokumentation im Adobe Portable Document Format (PDF) zur Verfügung. Sie können die PDF-Veröffentlichungen mithilfe von Adobe Acrobat Reader anzeigen. In der PDF-Dokumentation können Sie optional die Darstellung in Großschrift sowie Einstellungen für kontraststarke Anzeige verwenden und ausschließlich über die Tastatur navigieren. Für Benutzer von Sprachausgabeprogrammen wird in diesem Fall jedoch kein alternativer Text zur Verfügung gestellt.

Unter der Adresse der IBM Security Directory Integrator-Dokumentation können Sie auf die PDF-Veröffentlichungen für IBM Security Directory Integrator zugreifen oder diese herunterladen.

IBM und behindertengerechte Bedienung

Weitere Informationen zum Engagement von IBM bei der Bereitstellung von Eingabehilfen finden Sie im IBM Human Ability and Accessibility Center.

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 in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieser Dokumentation 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 diesen Informationen 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
2Z4A/101
11400 Burnet Road
Austin, TX 78758 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 Lieferung des im Dokument aufgeführten 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 Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren; sie 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 Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielpprogramme 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 Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit IBM Anwendungsprogrammierschnittstellen konform sind.

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

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corporation abgeleitet. © Copyright IBM Corp. _Jahr/Jahre angeben_. Alle Rechte vorbehalten.

Marken

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der International Business Machines Corp. in den USA und/oder anderen Ländern. Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" unter www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript und alle auf Adobe basierenden Marken sind Marken oder eingetragene Marken der Adobe Systems Incorporated in den USA und/oder anderen Ländern.

IT Infrastructure Library ist eine eingetragene Marke der Central Computer and Telecommunications Agency. Die Central Computer and Telecommunications Agency ist nunmehr in das Office of Government Commerce eingegliedert worden.

Intel, das Intel-Logo, Intel Inside, das Intel Inside-Logo, Intel Centrino, das Intel Centrino-Logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken der Intel Corporation oder ihrer Tochtergesellschaften in den USA und anderen Ländern.

Linux ist eine Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

ITIL ist eine eingetragene Marke, eine eingetragene Gemeinschaftsmarke des Office of Government Commerce und eine eingetragene Marke, die beim US Patent and Trademark Office registriert ist.

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



Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Cell Broadband Engine wird unter Lizenz verwendet und ist eine Marke der Sony Computer Entertainment, Inc. in den USA und/oder anderen Ländern.

Linear Tape-Open, LTO, das LTO-Logo, Ultrium und das Ultrium-Logo sind Marken von HP, IBM Corp. und Quantum in den USA und/oder anderen Ländern.

Index

Sonderzeichen

.registry, Datei
Komponenten 61

A

Abgelaufenes Zertifikat 208
Ablauf
Lösungsansicht erstellen 327
ACL
Konfiguration 298
Action Manager 269
Aktion hinzufügen/modifizieren 312
aktivieren 281
AMC 281
Auslöser 318
Ereignisdaten 317
Auslöser konfigurieren 310
beenden 271
Befehlszeilendienstprogramm 321
Derby-Datenbank 281
Ereignisdaten 316
Ergebnistabelle 307
Fenster 282
hinzufügen 309
Konfigurationsregeln 309
Konfigurierte Aktionen 311
löschen 309
modifizieren 309
starten 271
Status 282
Status überwachen 303
über Fernzugriff 271
Variable ersetzen 316
Active Directory 95
ActiveMQ
Protokollierung 179
Administration and Monitoring Console
Installation 267
Konfiguration 267
Administratorrechte 6
Administratorzugriff 6
Agentenkonfiguration
XML-Datei 395
Aktion auswählen
AMC 292
Aktion hinzufügen/modifizieren
Action Manager 312
Aktualisierung
Details der Lösungsansicht 308
Aktualisierungssite 3
Aktualisierungssite des Konfigurations-
editors
Implementierung von Eclipse 47
Algorithmus
Verschlüsselung 106
Allgemeine Konzepte 1
AMC 3, 268, 269, 271, 321
Abmeldung 289, 291
Action Manager 267, 283
AMC (*Forts.*)
Aktionen 274
Auslöser 274
Threads 274
Aktion auswählen 292
allgemeine Informationen 48
amc.properties 288
angepasstes Laden 303
Anmeldung 289
Arbeitsbereich 290
Auslöser erzwingen 283
Benutzerdefinierte Authentifizie-
rung 286
Benutzerschnittstelle 289, 290, 292
blättern 292
Derby 270
ferne Server 286
filtern
Tabellen 293
Funktionen 283
IBM Security Directory Integrator-Ser-
ver 286
Integrated Solutions Console 267
ISC 273
Kennwort 288
Konfiguration 270, 272
Konfiguration mit SSL 284, 286
Konfiguration ohne SSL 286
Konfigurationsdateien 294, 303
Konsolbenutzerberechtigung 273
Konsoleigenschaften 296
LDAP-Eigenschaften 270
Lösungsansicht 297
Navigationsbereich 290
Navigationslinks 273
Protokolle 272
Rollenverwaltung 287
Server 294
Änderung 295
hinzufügen 295
Sortierung 292
SSL-Eigenschaften 270
SSL-Schlüsselspeicher 284
SSL-Truststore 284
suchen
Tabellen 293
Tabellen 291, 292
Tabellenseiten 292
verschlüsselte Konfigurationen 288
verzögerte Implementierung 48
Webplattformimplementierung 48
AMC-Authentifizierung
Benutzer ohne Rootberechtigung 7
AMC-Implementierung
ISC 51
AMC-Installation 51
Änderung
Server 295
Änderungserkennungsconnector
EventHandler 95

Änderungserkennungsconnector für Sun
Directory 94
Änderungsprotokollconnector
EventHandler 92
Angepasste Benachrichtigungen
OMNIBus 423
Angepasstes Laden
Konfigurationsdateien 303
Anmelden 269
Anzeigenfolge
Deinstallation 34
Installation 13
Komponente hinzufügen 39
Migration 42
Apache ActiveMQ
Parameter 178
Apache Derby
Netzmodus 228
applyUpdates.bat(sh) 59
Auf Benutzername/Kennwort basierende
Authentifizierung
Authentifizierungshook 128
Auffüllung
aktivieren 207
inaktivieren 207
Ausbildung xi
Auslöser erzwingen
AMC 283
Auslöser konfigurieren
Action Manager 310
Ausnahmebedingungen
JDBC-Connector 174
LDAP-Connector 174
Authentifizierung 184
HTTP 166
Authentifizierungshook
auf Benutzername/Kennwort basie-
rende Authentifizierung 128

B

Beaufsichtigte Installation 47
Befehlszeile
Installation 44
Befehlszeilenoptionen
CE 233
CLI - Dienstprogramm "tdisrv-
ctl" 233
Server 233
Befehlszeilenparameter
cryptoutils 153
Befehlszeilenreferenz
allgemeine Optionen 239
Befehlszeilenschnittstelle
Ferne Server-API 238
tdisrvctl, Dienstprogramm 238
Befehlszeilentool
cryptoutils 151
Konfigurationsdatei bearbeiten 152
Verschlüsselung 151

- Beispiel für
 - Lösungsansicht erstellen 327
- Benachrichtigung
 - Benachrichtigungstypen 145
 - Unterdrückung 145
- Benachrichtigung senden
 - Listener 145
 - Zustellungsparameter 145
- Benutzerauthentifizierung
 - Systemspeicher 229
- Benutzerdefinierte Authentifizierung
 - Konfiguration mit SSL 286
 - Konfiguration ohne SSL 286
- Benutzerdefinierte Regeln
 - Beispiele 172
 - Format 172
- Benutzerregistry
 - Server-API 140
- Berechtigungsrollen 138
- Bereitstellungsprotokoll 335
- Bevorzugte anzeigen
 - Lösungsansichten 308
- Bewertung von Ereignissen 421

C

- CE 54
- Connectors 192
- cryptoutils
 - Befehlszeilenparameter 153
 - Befehlszeilentool 151
 - Konfigurationsdatei bearbeiten 152
 - Verschlüsselung 152
- CryptoUtils
 - entschlüsseln 203
 - verschlüsseln 203

D

- Dateisicherung
 - Dateiliste 86
- DB2
 - JDBC-Verbindungsparameter 226
 - Tabellenanweisungen 226
- Deinstallation
 - Anzeigenfolge 34
 - IBM Security Directory Integrator 55
- Derby 271
 - Benutzerberechtigung 155
 - RDBMS 227
 - Systemspeicher 227
- Details der Lösungsansicht
 - Aktualisierung 308
 - Komponenten anzeigen 308
- Detailtabelle
 - Lösungsansicht 305
- DNS-Namen
 - Konfiguration
 - MQ Everyplace 187
- Dokumentation
 - lokal 49
 - online 49
 - Softwarevoraussetzungen 65
- DSMLv2 96

E

- Eclipse-Aktualisierungsmanager
 - Aktualisierung 52
 - Installation 52
- EIF-Regeldatei
 - OMNIBus 421
- EIF-Testmonitor
 - OMNIBus 420
 - Port 420
- Eigenschaft
 - global 318
 - Java 318
 - Lösung 318
- Eigenschaften
 - CE 213
 - global 214
 - Java 215
 - Jlog-Datei 215
 - JVM 215
 - Konfiguration mit SSL 116
 - Lösung 215
 - Lösungsverzeichnis 213
 - PKCS# 116
 - System 216
- Eigenschaftendatei 421
 - Derby-Parameter 387
 - derby.properties 387
 - global.properties 387
 - ibmdisrv 384
 - jlog.properties 385
 - Log4J.properties 384
 - Lösungsverzeichnis 387
 - Protokolloptionen 384
- Eigenschaftendateien
 - extern 153
 - Lösungsverzeichnis 383
 - Verschlüsselung
 - Eigenschaftendateien 153
- Eigenschaftsspeicher
 - Eigenschaftsspeicher des Benutzers 223
 - globale Eigenschaften 318
 - Java-Eigenschaften 318
 - Kennwortspeicher 223
 - Lösungsansicht 318
 - Lösungseigenschaften 318
- Eingabehilfen xi, 425
- Eingabehilfen für dieses Produkt 425
- Einschränkungen des fernen Konfigurationseditors 161
- Ereignisdaten
 - Auslöser 317
- Erforderlicher Plattenspeicherplatz
 - Link 3
- Ergebnistabelle
 - Action Manager 307
- error (Fehler), Aktion 169
- Erstellungsanweisungen
 - Systemspeicher 229
- Erweiterungen für die Traceerstellung
 - Connectors 263
 - Parser 263
- EventHandler 89, 91, 93, 94, 96
- EventPropertyFile.properties, Datei
 - Struktur 421

F

- Fehlerbehebung xii
 - Fehler bei Apache Derby 231
- Ferne Clientsitzung
 - Authentifizierungsmethode 126
 - Server-API-Authentifizierung 126
- Ferne Konfiguration
 - MQ Everyplace 189
- Ferne Server-API 119, 123
- Ferner Konfigurationseditor 160
 - Einschränkungen 161
- Fertigungslinie 92, 96, 169
 - Zusammenfassung der Regeln anzeigen 318
- Fertigungslinien 89, 261
- FileAppender 260
- Fiorano MQ, System
 - JavaScript
 - Konfiguration 183
- FIPS
 - createstash 202
 - cryptoutils 202
 - externe Tools 202
 - keytool/Ikeyman 202
 - Konfiguration 191, 202
 - Konformität
 - Regeln 194
 - Verschlüsselung 192
- FIPS-Modus 191
- Fixes installieren
 - Komponenten 64
 - manuelle Schritte 64
- Fixpack 61
- Funktion für Konfigurationsdatei 157
- Funktionskomponenten 192

G

- Geheime Schlüssel
 - Konfiguration mit SSL 104
 - Schlüsselspeicher 104
- Geltungsbereich der Prüfung 144
- Grafikpakete
 - CE 8
 - UNIX-Systeme 8
- Grafisch orientiertes Installationsprogramm
 - Anweisungen 13
 - Installation 13

H

- Hilfdatei installieren 49
- Hilfesystem 3
- Hinzufügen
 - Server 295
- Hochverfügbarkeit
 - Konfiguration 188
- Hostbasierte Authentifizierung
 - global.properties 134
 - solution.properties 134
- HTTP 93, 335
- HTTP-Basisauthentifizierung 166

I

IBM
 Software Support xii
 Support Assistant xii

IBM SDI
 Client 119
 Debugging 253
 EIF-Connector 420
 JMS-Nachrichtenübertragungssystem 177
 JMX-Schnittstelle 211
 OMNIBus 420
 Protokollierung 253
 Server 119
 Server-API 211
 Systemwarteschlange 177

IBM SDI-Daten
 überwachen 406

IBM SDI-Installation 383

IBM Security Directory Integrator
 Web-Service-Suite 167

IBM Security Directory Integrator-Installation
 plattformspezifisch 13

IBM Security Directory Integrator-Server
 ECB 146
 FIPS 194
 Konfigurationsdateien 146
 Lokale Clientsitzung 126
 RSA 146
 Server-API-Authentifizierung 126
 Verschlüsselungsalgorithmen 146
 Verschlüsselungsmodule 194

IBM Security Directory Integrator-Service
 Deinstallation des Windows-Dienstes 376
 Eigenschaften 378
 Eigenschaftendatei 375
 ibmdiservice.props 378
 Installation des Windows-Dienstes 376
 Protokollierung 377
 Protokollservice 378
 Servicename 375
 Serviceprotokolle 377
 starten 377
 Startscript 382
 stoppen 377
 Stoppscript 382
 UNIX-Systeme 382
 Windows-Dienst 375, 377
 Deinstallationschritte 376
 Deinstallationsverfahren 376
 Installationsschritte 376
 Installationsverfahren 376
 Windows-Systeme 382

IBM Security Directory Integrator-Services
 i5/OS 375
 Linux/Unix 375
 Windows 375
 z/OS 375

IBM Tivoli Monitoring Agent 395

IBM Tivoli Monitoring Agent Editor 404

IBM WebSphere 168
 MQ Everyplace
 Parameter 180

IBM WebSphere (Forts.)
 MQ-Parameter 181

IBMPCKS11
 SSL-Schlüssel 207
 Zertifikate 207

Implementierung
 AMC 268
 UNIX-Prozess 268
 vorhandene Umgebung 269
 WebSphere Application 269
 Windows-Dienst 268
 Migration 49

Installation 6
 Anweisungen 3
 Anzeigenfolge 13
 Befehlszeile 44
 Systemvoraussetzung 3

Installation und Verwaltung 1

Installation von IBM SDI 9

Installationsabschluss 54
 Schritte 47

Installationsposition
 Linux und Unix 57
 Windows 57

Installationsprogramm
 Migration
 automatisch 70
 manuell 70

Installationsprogramm starten
 direktes Installationsprogramm 10
 Launchpad 10

Instanzkonfiguration 352

Integrated Solutions Console
 Implementierung 268

Integration externer Tools 393
 Tivoli Monitoring 393
 Tivoli Netcool/OMNIBus 393

Integrierte Webplattform 3

Intermediary
 Touchpoint-Instanz 365

ITM
 Agent 394
 angepasste Benachrichtigungen 419
 Architektur 394
 AssemblyLine, Tabelle 413
 Datenerfassung 394
 Eclipse 395
 Fertigungslinie 412
 importieren 394
 ITM-Agent 394
 Konfigurationsdatei 419
 Linkerstellung 413
 Tabellen
 Links 412
 überwachen 394

ITM-Agent
 Einschränkung 420
 Generierung 404
 IBM SDI-Server 420
 Implementierung 404
 importieren 395
 Konfiguration 404, 405

ITM Agent Builder 405

ITM Agent Builder 6.2 395

J

Java-API-Dokumentation 3

Java-Eigenschaften 177

Java-Systemeigenschaft 125

JDBC 192

JLOG
 Parameter 266

JLOG-basiert 385

JLOG-Protokollfunktion 265

JMS 192

JMS-Treiber 177
 JavaScript 182
 Parameter für JMS-Scripttreiber 181
 ret, JavaScript 183

JMX-Connector 92

JMX-Schicht
 Server-API 136

JRE 3, 6

JSSE 166

K

Kennwort für Konfiguration 157

Kennwortgeschützte Konfigurationen
 Ausnahmebedingung 212

Kennwortschutz 159

Kennwortspeicher 158

Kennwortsynchronisation 3
 Plug-ins 48

keytool 107

Komponente hinzufügen
 Anzeigenfolge 39

Komponenten
 verfügbar 3

Komponenten anzeigen
 Details der Lösungsansicht 308

Komponentenkennwort 158

Konfiguration
 ActiveMQ
 Parameter 178
 AMC-Protokolle 272
 Apache Derby-Instanzen 228
 Eigenschaften 213
 Intervall 212
 Kennwort 158
 Konfiguration mit SSL 114, 202
 Lösungsverzeichnis 213
 Microsoft Active Directory 114
 PKI 202
 Systemspeicher 228
 Zeitlimit für Ladevorgänge 212

Konfiguration, Beispiel
 Systemwarteschlange 184

Konfiguration der Systemwarteschlange 177

Konfiguration mit SSL 104, 125, 160, 192
 ActiveMQ 179
 Beispiel für 117
 Client 113, 117, 118
 Clientauthentifizierung 114
 Connectors 110
 Eigenschaften 116
 Fernzugriff 124
 IBM Security Directory Integrator-Komponente 117, 118

- Konfiguration mit SSL (*Forts.*)
 - IBM Security Directory Integrator-Komponenten 111, 113
 - JSSE 125
 - JVM 125
 - Konfiguration 110, 114, 166
 - LDAP-Connector 117, 118
 - lokaler Zugriff 124
 - Schlüssel 105
 - Schlüsselspeicher 105, 111, 114
 - Server 111, 117
 - Systemeigenschaft 125
 - Truststore 105, 111, 113, 114
- Konfigurationen 89
- Konfigurationsdateien 160
 - Lösungsansicht 302
 - Verschlüsselung 149
 - Zusammenfassung 162
- Konfigurationseditor 3, 9, 161
 - AIX 8
 - Eclipse 233
 - Eigenschaften 158
 - Optionen für Perspektive 233
 - RPM 8
 - Server beenden 233
 - Tombstone 368
- Konfigurationseinstellungen
 - Migration 88
- Konfigurationsfenster
 - Fertigungslinie 368
- Konfigurationsregeln
 - Action Manager 309
- Konfigurierte Aktionen
 - Action Manager 311
- Konsole 261
- Konsoleigenschaften
 - allgemein 296
 - JDBC 296
 - Konfiguration mit SSL 296
- Körperliche Behinderung 425

L

- Laufzeitserver 3
- LDAP 94
- LDAP-Authentifizierung
 - Benutzername 131
 - global.properties 130
 - Kennwort 131
 - solution.properties 130
- LDAP-Connector
 - Konfiguration mit SSL 117, 118
- Linkerstellung 412
- Linux/Unix-Dienst
 - anpassen 380
 - beenden 380
 - Implementierungsmethoden 380
- list, Befehl 103, 107
- Log4J 260
- Lokale Clientsitzung
 - IBM Security Directory Integrator-Server 126
 - JVM 126
- Lokale Variable
 - Lösungsansicht 299
- Lösungsansicht
 - Benutzer konfigurieren 298

- Lösungsansicht (*Forts.*)
 - bevorzugt 321
 - Detailtabelle 305
 - hinzufügen 297, 299
 - Konfigurationsdatei 299
 - Konfigurationsdateien 302
 - lokale Variable 299
 - modifizieren 297
 - Status überwachen 305
- Lösungsansichten
 - bevorzugte anzeigen 308
- Lösungsverzeichnis 185
 - Server-Hooks 160
- Lotus Domino 166

M

- Mailbox-Connector 91
- Manage Tivoli Monitoring Enterprise Services 405
- Manuelle Migration 71
 - Dateien 72
 - Eigenschaften 72
 - Komponenten 72
 - Konfigurationen 72
 - Scripts 72
 - Sicherungstools 87
- Manuelle Sicherung
 - Arbeitsbereichsdateien 88
 - Derby-Datenbankdateien 88
 - LDAP 88
 - OSGI 88
 - SCIM 88
 - Warteschlangenmanagerdateien 88
- Microsoft Active Directory 114
- Migration
 - andere Position 67, 68
 - Änderung 68
 - Anzeigenfolge 42
 - Arbeitsbereich 69
 - B-Baum-Connector 97
 - B-Baum-Tabellen 97
 - Cloudscape-Datenbank 97
 - Datei 67
 - Dateitypen 67, 68, 69
 - Derby 97
 - durch Installationsprogramm unterstützt
 - manuell 71
 - Implementierung 49
 - Installationsprogramm
 - automatisch 70
 - manuell 70
 - Komponenten 67
 - Konfigurationseinstellungen 88
 - Migrationsarten 70
 - neuere Version 70
 - Scripts 69
 - Systemspeicher 97
 - Szenario 67
 - verschlüsselte Daten 70
- Minizertifikate 168
- MQ Everyplace 184
 - Kennwortänderungen 188
 - Konfigurationsdienstprogramm 185, 189
 - Server für Minizertifikate 186

- MQ Everyplace-Authentifizierung 168
- MQ-Warteschlangenmanager 185
- MS SQL Server
 - JDBC-Verbindungsparameter 225
 - Tabellenanweisungen 225

O

- Öffentliche Schlüssel 104
- OMNIBus
 - angepasste Benachrichtigungen 423
- Operationen
 - event 240
 - Konfigurationsdatei 240
 - prop 240
 - queryop 240
- Oracle
 - JDBC-Treiber
 - Clientbibliothek 224
 - Verbindungsparameter 224

P

- Parameter
 - MQ Everyplace 180
- Parametertyp 157
 - Kennwort 157
- Parser 192
- Password Synchronizer 61
- Persistenz
 - Touchpoint-Instanz 347
- PKCS#11 116
- Plattformspezifisch
 - Installation 13
- Plug-ins
 - Kennwortsynchronisation 48
- Private Schlüssel 104
- Problembestimmung xii
- Protokollebenen 259
- Protokollebenensteuerung 259
- Protokollierung
 - CE 255
 - Fertigungslinie 254, 255
 - FFDC 263
 - Log4J-Standardklasse 255
 - scriptbasiert 254
 - Traceerstellung 263
- Protokollstrategien 261
- Protokollverwaltung
 - Lösungsansicht 320
- Prüffunktionen des Servers
 - Authentifizierung 144
 - Autorisierung 144
 - Benachrichtigungen 144
- Prüfungsfunktion
 - Prinzipien 144

R

- reconnect (Verbindung wiederherstellen), Aktion 169
- Regel zur Verbindungswiederherstellung
 - CE 174
 - Konfiguration 174

- Regeln zum Wiederherstellen einer Verbindung
 - benutzerdefiniert 170
 - integriert 170
- Regelsteuerkomponente für Verbindungs-wiederherstellung 169
- RHEL 6
- RMI 119
- Rollback
 - Update Installer 64
- Rollenverwaltung
 - Administrator 287
 - Ausführen 287
 - Konfigurationsadministrator 287
 - Lesen 287
- RPM
 - AIX 8

S

- Schlüssel
 - Zertifikat 107
- Schlüssel erstellen 103, 107
- Schlüssel verwalten 103
- Schlüsselspeicher 104, 107, 208
 - JCEKS 104
 - JKS 104
 - list, Befehl 107
 - PKCS#12 104
- Schulung xi
- Schwellenwert
 - AssemblyLine, Tabelle 408
 - Beispiel für 408
 - definieren 408
- SDI
 - Editionen 1
- SDI-Ladeprogramm 54
- Security Directory Server 92
- Security Enhanced 6
- SELinux 6
- Server
 - Befehlszeilenoptionen 234
 - IBM SDI 234
- Server, gesicherter Modus
 - gesichert 148
 - Standard 148
- Server-API 124, 125, 130
 - Authentifizierung 126, 136
 - Beispiele 136
 - Benutzerregistry 140
 - Eigenschaften 120
 - JMX 136
 - JMX-Schicht 136
 - Konfiguration 120
 - Zugriff 123
- Server-API-Authentifizierung
 - ferne Clientsitzung 127
 - JAAS-Authentifizierung 127
 - SSL-basierte Authentifizierung 127
- Server-API-Autorisierung
 - Client-Server-API-Sitzung 138
 - ferne API 138
- Server-Connector 93, 94, 96
 - TCP 91
- Server-ID
 - AMC 211
 - IP-Adresse 211

- Server-RMI
 - SSL-Zugriff 212
- Serverauthentifizierung
 - auf Benutzername/Kennwort basie-rend 135
 - hostbasiert 135
 - LDAP 135
 - SSL-basiert 135
- Serverbenachrichtigungsconnector 421
- ServiceName
 - UNIX 47
- Sicherheit
 - API 103
 - Konfiguration mit SSL 103
 - Konfigurationsdateien 103
 - Schlüssel verwalten 103
 - Verschlüsselung 184
 - Webverwaltungskonsole 103
- Sicherheit für die Webverwaltungskonsole
 - Details 166
- Sicherheit für Systemspeicher
 - benutzerdefinierte Klasse 155
 - externer Verzeichnisservice 155
 - integrierte Derby-Benutzer 155
- Sicherheitsaspekte
 - verschieden 166
- Sicherheitseigenschaften
 - Zusammenfassung 162
- Sicherheitskonzepte 104
- Sicherheitsmodell der Server-API 138
 - Berechtigungsrollen 138
- Sicherheitstools
 - Ikeyman 106
 - JVM 106
 - keytool 106
- Sichern
 - Apache Derby-Datenbanken 230
- Sicherung
 - Upgrade von 6.0 auf 7.1 86
 - Upgrade von 7.0 auf 7.1 86
 - Upgrade von 7.1 auf 7.1.1 87
 - Upgrade von Version 6.1.x auf 7.1 86
 - wichtige Daten 86
- Sicherungstools
 - manuelle Migration
 - backupam/restoram 87
 - backupamc/restoramc 87
 - backupamcdb/restoramcdb 87
- SNMP-Server-Connector
 - EventHandler 92
 - Fertigungslinie 92
- SOAP 96
- solidDB
 - JAR 226
 - JDBC-Verbindungsparameter 226
 - Tabellenanweisungen 226
- Speicherbereich für temporäre Dateien
 - UNIX/LINUX 46
 - Windows 46
- SSL-basierte Authentifizierung 160
- SSL-Clientauthentifizierung 160
- Standardinstallation
 - Position 57
- Standardparameter 260
- starten 269
- Starten des Deinstallationsprogramms
 - Ergebnisse 55

- Starten des Deinstallationsprogramms (Forts.)
 - Verfahren 55
- Stashdatei
 - Schlüsselkennwort 147
 - Schlüsselspeicherkennwort 147
 - Sicherheit für Serverinstanz 147
- Status überwachen
 - Action Manager 303, 304
 - Ergebnis der Überwachung des ord-nungsgemäßen Betriebs 304
 - Überwachung des ordnungsgemäßen Betriebs 304
- Systemspeicher
 - Benutzerauthentifizierung 229
 - Cloudscape 219
 - DDL 224
 - JDBC-Treiber 224
 - JVM 219
 - Lösungsverzeichnisse 219
 - RDBMS 224
- Systemspeicherbedarf
 - Link 3
- Systemwarteschlange
 - Konfiguration, Beispiel 184
 - Parameter für Microbroker 181

T

- Tabelle für "EventNotifier" 412
- Tabellen 412
 - AMC 293
- TCB 159
- TCP
 - Server-Connector 91
- TEP-Agenten 406
- Tivoli Enterprise Portal 406
- Tivoli Monitoring 393
- Tivoli Netcool/OMNIBus 393
- Tombstone
 - Attribute 370
 - Datensätze 370
 - Fertigungslinie 370
 - Konfiguration 370
 - Konfigurationseditor 368
 - Konfigurationsfenster 368
 - Statistik 370
- Tombstone Manager
 - Fertigungslinien 367, 371
 - Konfigurationseigenschaften 371
 - Konfigurationsinstanz 371
- Tombstones
 - Konfiguration 367
 - Konfigurationseditor 367
 - Switches 367
- Touchpoint 335
 - Authentifizierung 361
 - Eigenschaftenblatt
 - Definitionen 357
 - Eintragsobjekte 355
 - HTTP 354, 355, 356
 - HTTP-Inhalt 355
 - HTTP-Server 361
 - Initiator 355
 - Instanz 353
 - Instanzen 335
 - Intermediary 355

- Touchpoint (*Forts.*)
 - Kommunikationsprotokoll 353
 - Konfiguration 335
 - Position 358
 - Provider 354
 - RMI-Server-API 361
 - Schema für Statuseintrag 356
 - Server 335
 - Statuseintrag 356
 - Touchpoint-Instanzen 355
 - XML-Schema 355, 358
- Touchpoint-Instanz 352
 - Beispiel für 362
 - Eintragsressource 363, 364
 - HTTP 338
 - im Lieferumfang enthaltenes Beispiel 362
 - Initiator 338, 362, 364
 - Intermediary 338, 365
 - POST, HTTP-Anforderung 362
 - Provider 338, 362, 363
 - Ressourcen 347
 - Ressourcenpersistenz 347
 - URL 362
- Touchpoint-Instanzen
 - Abläufe bei einem Fehler 358
 - Atom-Dokument 353
 - XML-Dokument 358
 - Zielkonfiguration 353
- Touchpoint-Konfiguration
 - Namensbereich 352
 - Touchpoint-Instanz 352
- Touchpoint-Provider
 - Instanzen 336
 - JVM 336
 - Touchpoint-Server 336
- Touchpoint-Schablone
 - Fertigungslinien 342
 - Initiator 342
 - IntermediaryHandler 342
 - ProviderHandler 342
 - Ressourcen 342
- Touchpoint-Schema
 - HTTP 348
 - Instanzen 348
 - Ressource 348
 - Schema, Baumstruktur 348
 - Server 348
- Touchpoint-Server
 - HTTP-Basisauthentifizierung 360
 - Komponenten 335
 - Konfiguration 360
 - ReSTful, Kommunikationsprotokoll 335
 - Touchpoint-Provider 336
 - Web-Container 360
 - Zugriff 335
- Touchpoint-Typ
 - angepasster Typ 336
 - Standardtyp 336
 - virtueller Typ 336
- Traceeigenschaften
 - dynamisch 265
- Traceerstellung 159
 - JLOG-Protokollebene 264
 - JlogSnapHandler 263
 - Konfiguration 264

- Traceerstellung (*Forts.*)
 - PDLogger, Objekt von JLOG 263
 - SnapMemory 263
- Tracestufen 265

U

- Unbeaufsichtigte Deinstallation
 - Deinstallation über Konsole 56
 - GUI 56
- Unbeaufsichtigte Installation 47
- UNIX
 - ServiceName 47
- Unterstützte Plattformen
 - Link 65
- Unterstützung der LDAP-Authentifizierung 130
- Unterstützung des symmetrischen Chiffrierwerts
 - Verschlüsselung 192
- Unterstützung von LDAP-Gruppen
 - Authentifizierungsprozess 132
 - Benutzer 132
 - Benutzerregistry 132
 - Gruppe 132
- Update Installer 59, 61
 - Fehlerbehebung 64
 - Rollback 64
- Upgrade 9
 - Version 7.1.1 auf 7.2 87

V

- Verschlüsselte Daten
 - Migration 70
- Verschlüsselte Konfigurationen
 - AMC 288
- Verschlüsselung 191
 - geheimer Schlüssel 106
 - global.properties 152
 - Konfigurationsdatei 151
 - Konfigurationsdateien 149
 - Lösungsverzeichnis 160
 - öffentliche/private Schlüssel 149
 - öffentlicher/privater Schlüssel 106
 - RSA 149
 - Server-Hooks 160
 - solution.properties, Datei 152
 - Unterstützung des symmetrischen Chiffrierwerts 192
- Verschlüsselungsartefakte 208
- Verschlüsselungsdienstprogramm 153
- Verschlüsselungsschlüssel 104, 208
 - Hardwareeinheiten 206
 - JRE 206
 - Konfiguration mit SSL 206
 - PKCS 206
 - RSA 206
- Voraussetzungen 6
- Voraussetzungen der Systemplattform
 - Link 3
- VPN
 - Eigenschaften 123

W

- Wichtige Daten
 - Sicherung 86
- Windows-Betriebssystem 9
- Windows-Dienst 375

Z

- Zertifikat
 - Konfiguration mit SSL 202
 - PKI 202
- Zertifikat für öffentlichen Schlüssel 104, 107
- Zertifikate
 - digital 203
 - Konfiguration mit SSL 203
 - PKI 203
 - selbst signiert 205
 - von Zertifizierungsstellen signiert 205
 - Zertifizierungsstelle 203
- Zertifikate konfigurieren
 - Konfiguration mit SSL 205
 - PKI 205
- Zertifikatserstellung 167
- Zugriff
 - Server-API 123



SC12-4443-03

