

Oracle® Fusion Cloud EPM

Mit EPM Automate für Oracle Enterprise
Performance Management Cloud arbeiten



F28917-21



Oracle Fusion Cloud EPM Mit EPM Automate für Oracle Enterprise Performance Management Cloud arbeiten,

F28917-21

Copyright © 2016, 2024, Oracle und/oder verbundene Unternehmen.

Primärer Autor: EPM Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Inhalt

Dokumentation zur Barrierefreiheit

Dokumentationsfeedback

1 Informationen zu EPM Automate

EPM Automate installieren	1-1
Kapazitäts- und Portanforderungen	1-2
Unterstützte Plattformen	1-2
Java Runtime Environment und EPM Automate	1-4
OpenJDK verwenden	1-4
Anweisungen für Windows	1-5
Anweisungen für Linux/UNIX/macOS X	1-5
Serverseitige Ausführung von EPM Automate-Befehlen	1-6
Erläuterungen zur Verschlüsselungsebene für EPM Automate	1-6
Autorisierungsprotokoll OAuth 2.0 bei OCI (Gen2)-Umgebungen verwenden	1-6

2 Befehle

Informationen zur Ausführung von EPM Automate-Befehlen	2-1
Voraussetzungen	2-1
Standarddateispeicherorte	2-3
TLS-Protokoll 1.2 aktivieren	2-4
EPM Automate-Befehle verwenden	2-5
Mehrere Werte für Parameter angeben	2-5
Verhalten während der täglichen Wartung	2-6
EPM Automate ausführen	2-6
Windows	2-6
Linux	2-8
Mehrere Instanzen von EPM Automate ausführen	2-8
Befehle auf einen Blick	2-11
EPM Automate-Befehle	2-16

addUsers	2-16
addUsersToGroup	2-18
addUsersToTeam	2-19
addUserToGroups	2-20
applicationAdminMode	2-21
applyDataGrants	2-21
archiveTmTransactions	2-22
assignRole	2-24
autoPredict	2-26
calculateModel	2-26
clearCube	2-29
clearDataByPointOfView	2-30
clearDataByProfile	2-31
clearPOV	2-32
cloneEnvironment	2-33
copyDataByPointOfView	2-38
copyDataByProfile	2-40
copyFileFromInstance	2-40
copyFromObjectStorage	2-41
copyOwnershipDataToNextYear	2-43
copyPOV	2-43
copySnapshotFromInstance	2-45
copyToObjectStorage	2-46
createGroups	2-47
createNRSnapshot	2-48
createReconciliations	2-49
deleteFile	2-49
deleteGroups	2-51
deletePointOfView	2-51
deletePOV	2-52
deployCube	2-52
deployEJTemplates	2-53
deployFormTemplates	2-54
deployTaskManagerTemplate	2-55
dismissIPMInsights	2-56
downloadFile	2-56
enableApp	2-57
enableQueryTracking	2-58
encrypt	2-58
executeAggregationProcess	2-59
executeBurstDefinition	2-60

executeReportBurstingDefinition	2-61
exportAccessControl	2-61
exportAppAudit	2-62
exportAppSecurity	2-63
exportARApplicationProperties	2-64
exportBackgroundImage	2-65
exportCellLevelSecurity	2-65
exportConsolidationJournals	2-66
exportData	2-66
exportDataManagement	2-67
exportDimension	2-67
exportDimensionMapping	2-68
exportEJournals	2-69
exportEssbaseData	2-70
exportJobConsole	2-71
exportLibraryArtifact	2-74
exportLibraryDocument	2-75
exportLogoImage	2-76
exportMapping	2-76
exportMetadata	2-77
exportOwnershipData	2-78
exportQueryResults	2-78
exportSnapshot	2-81
exportTemplate	2-82
exportTaskManagerAccessControl	2-82
exportValidIntersections	2-83
extractDimension	2-84
feedback	2-85
getApplicationAdminMode	2-86
getDailyMaintenanceStartTime	2-87
getEssbaseQryGovExecTime	2-88
getIdleSessionTimeout	2-88
getIPAllowlist	2-89
getSubstVar	2-90
getVirusScanOnFileUploads	2-91
groupAssignmentAuditReport	2-91
help	2-92
importAppAudit	2-93
importAppSecurity	2-93
importARApplicationProperties	2-94
importBackgroundImage	2-95

importBalances	2-95
importCellLevelSecurity	2-96
importConsolidationJournals	2-97
importData	2-98
importDataManagement	2-98
importDimension	2-99
importJobConsole	2-100
importLibraryArtifact	2-101
importLogoImage	2-103
importMapping	2-103
importMetadata	2-104
importOwnershipData	2-105
importPreMappedBalances	2-106
importPreMappedTransactions	2-107
importProfiles	2-107
importRates	2-108
importRCAttributeValues	2-109
importReconciliationAttributes	2-110
importSnapshot	2-111
importSupplementalCollectionData	2-114
importSupplementalData	2-115
importTemplate	2-116
importTMAttributeValues	2-117
importTmPremappedTransactions	2-118
importValidIntersections	2-119
invalidLoginReport	2-121
listBackups	2-122
listFiles	2-123
loadData	2-124
loadDimData	2-125
loadViewpoint	2-126
login	2-127
logout	2-130
maskData	2-131
mergeDataSlices	2-131
mergeSlices	2-132
optimizeASOCube	2-132
programDocumentationReport	2-133
provisionReport	2-134
purgeArchivedTmTransactions	2-136
purgeTmTransactions	2-136

recomputeOwnershipData	2-138
recreate	2-138
refreshCube	2-142
removeUserFromGroups	2-143
removeUsers	2-144
removeUsersFromGroup	2-145
removeUsersFromTeam	2-146
renameSnapshot	2-147
replay	2-148
resetService	2-149
restoreBackup	2-150
restructureCube	2-151
roleAssignmentAuditReport	2-152
roleAssignmentReport	2-153
runAutomatch	2-155
runBatch	2-155
runBusinessRule	2-156
runCalc	2-157
runComplianceReport	2-159
runDailyMaintenance	2-160
runDataRule	2-161
runDMReport	2-163
runIntegration	2-165
runMatchingReport	2-169
runPlanTypeMap	2-170
runRuleSet	2-171
runSupplementalDataReport	2-171
runTaskManagerReport	2-172
sendMail	2-173
setApplicationAdminMode	2-175
setDailyMaintenanceStartTime	2-175
setDemoDates	2-176
setEJJournalStatus	2-177
setEncryptionKey	2-178
setEssbaseQryGovExecTime	2-179
setIdleSessionTimeout	2-179
setIPAllowlist	2-180
setManualDataAccess	2-181
setPeriodStatus	2-182
setSubstVars	2-183
setVirusScanOnFileUploads	2-183

simulateConcurrentUsage	2-184
skipUpdate	2-188
snapshotCompareReport	2-190
sortMember	2-191
unassignRole	2-192
updateUsers	2-194
upgrade	2-195
uploadFile	2-196
userAuditReport	2-197
userGroupReport	2-198
validateConsolidationMetadata	2-199
validateModel	2-200
Exit-Codes	2-201

3 Beispielszenarios für die Befehlsausführung

Beispielskripte kopieren	3-1
Beispielszenarios für alle Services	3-1
Anwendungs-Snapshots auf einem Computer sichern	3-3
Benutzer über den Abschluss der täglichen Wartung informieren	3-5
Snapshot in oder aus Oracle Object Storage kopieren	3-13
Benutzer erstellen und vordefinierten Rollen zuweisen	3-15
Anzahl der lizenzierten Benutzer zählen (Benutzer, die Rollen zugewiesen sind)	3-19
Auditberichte zu Benutzern erstellen, die Rollen zugewiesen sind	3-21
Auditberichte zu Rollenzuweisungen und zur Aufhebung von Zuweisungen	3-25
Zugriffslogs und Aktivitätsberichte zur Einhaltung von Datenschutzgesetzen maskieren	3-29
Downloads von Aktivitätsberichten auf einen lokalen Computer automatisieren	3-34
Zugriffslogs aus einer Umgebung herunterladen	3-38
Klonen von Umgebungen automatisieren	3-41
Unnötige Dateien aus einer Umgebung entfernen	3-45
Dateien aus einer Umgebung suchen und herunterladen	3-47
Alte EPM Cloud-Umgebungen für Audits neu erstellen	3-49
Audit und Compliance des Datenbankzugriffs automatisieren	3-60
Benutzer und vordefinierte Rollenzuweisungen replizieren	3-71
Benutzer aus einer Identitätsdomain in eine andere replizieren	3-72
Vordefinierte Rollenzuweisungen aus einer Umgebung in eine andere replizieren	3-79
Vierteljährliches Upgradeintervall für EPM Cloud erstellen	3-87
Skript und Anweisungen für Windows	3-88
Skript und Anweisungen für UNIX/Linux	3-91
Groovy-Skript	3-95
Vierteljährliches Upgradeintervall für EPM Cloud mit sechswöchigen Testzyklen erstellen	3-99

Beispielszenarios für Planning, Consolidation, Tax Reporting und Enterprise Profitability and Cost Management	3-113
Export einer großen Anzahl von Zellen aus einem Aggregate Storage-Cube automatisieren	3-114
Metadaten in Anwendungen importieren	3-124
Daten importieren, Berechnungsskripte ausführen und Daten aus einer Block Storage-Datenbank in eine Aggregate Storage-Datenbank kopieren	3-126
Metadaten und Daten exportieren und herunterladen	3-129
Anwendungsdaten exportieren und herunterladen	3-131
Das Archivieren von Auditdatensätzen der Anwendung automatisieren	3-134
Windows-Skript	3-135
Linux-Skript	3-136
Eine Datendatei in eine Umgebung hochladen und eine Dataload-Regel ausführen	3-137
Integration von täglichen Daten automatisieren	3-140
Beispielszenarios für Account Reconciliation	3-143
Vorformatierte Salden in eine Periode laden	3-143
Backup-Snapshots hochladen und importieren	3-146
Alte abgeglichene Transaktionen archivieren und archivierte Transaktionen löschen	3-148
Beispielszenarios für Profitability and Cost Management	3-155
Metadaten in Anwendungen importieren	3-155
Daten importieren und Programmregeln ausführen	3-157
Beispielszenarios für Oracle Enterprise Data Management Cloud	3-160
Dimensionen und Zuordnungen von Oracle Enterprise Data Management Cloud mit EPM Cloud-Anwendungen synchronisieren	3-160
EPM Cloud-Dimensionen mit Oracle Enterprise Data Management Cloud-Anwendungen synchronisieren	3-163
Skriptausführung automatisieren	3-164
EPM Automate-Aktivitäten überwachen	3-164

4 Befehle ohne Installation von EPM Automate ausführen

Umgebungen, die die serverseitige Befehlsausführung unterstützen	4-1
Informationsquellen	4-2
Unterstützte Befehle	4-2
Methoden zur Ausführung von EPM Automate mit serverseitigen Groovy-Skripten	4-3
Umgebungen mit einem serverseitigen Groovy-Skript klonen	4-3
Aktivitätsbericht mit serverseitigem Groovy-Skript per E-Mail versenden	4-5

5 EPM Cloud-Umgebung replizieren

Tägliche Replikation einrichten	5-1
On-Demand-Replikationen einrichten	5-2

A Ausführen des Befehls "simulateConcurrentUsage" vorbereiten

Datei "requirement.csv" erstellen	A-1
Eingabedateien erstellen	A-3
Eingabedatei für Formulare öffnen	A-4
Eingabedatei für Formulare speichern	A-4
Eingabedatei für Geschäftsregeln ausführen	A-5
Eingabedatei für Datenregeln ausführen	A-5
Eingabedatei für Ad-hoc-Raster	A-6
Eingabedatei für Berichte ausführen	A-6
Eingabedatei für Bücher ausführen	A-7
Datei "UserVarMemberMapping.csv" erstellen	A-7
Eingabe-ZIP-Datei erstellen und in die Umgebung hochladen	A-8
Beispiel für einen Bericht zur Simulation der gleichzeitigen Verwendung	A-8

B Ausführen des Wiedergabebefehls vorbereiten

Informationen zum Wiedergabebefehl	B-1
Voraussetzungen	B-1
HAR-Dateien erstellen	B-2
Wiedergabedateien erstellen	B-5
Tracedateien generieren	B-6
Wiedergabesession - Beispiel	B-6

C Sonderzeichen verarbeiten

D Spezifische Befehle für jeden EPM Cloud-Service

Account Reconciliation-Befehle	D-2
Befehle für Financial Consolidation and Close	D-3
Narrative Reporting-Befehle	D-4
Oracle Enterprise Data Management Cloud-Befehle	D-5
Befehle für Planning, Planning Modules, FreeForm, Strategische Personalplanung und Sales Planning	D-6
Profitability and Cost Management-Befehle	D-7
Enterprise Profitability and Cost Management-Befehle	D-8
Tax Reporting-Befehle	D-9

Dokumentation zur Barrierefreiheit

Informationen zu Oracles Verpflichtung zur Barrierefreiheit erhalten Sie über die Website zum Oracle Accessibility Program <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Zugriff auf Oracle Support

Oracle-Kunden mit einem gültigen Oracle-Supportvertrag haben Zugriff auf elektronischen Support über My Oracle Support. Weitere Informationen erhalten Sie unter <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> oder unter <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>, falls Sie eine Hörbehinderung haben.

Dokumentationsfeedback

Um Feedback zu dieser Dokumentation abzugeben, klicken Sie unten auf der Seite eines beliebigen Themas im Oracle Help Center auf die Schaltfläche "Feedback". Sie können auch eine E-Mail an epmdoc_ww@oracle.com senden.

1

Informationen zu EPM Automate

Mit EPM Automate können Sie Aufgaben in Oracle Enterprise Performance Management Cloud-Umgebungen remote durchführen.

EPM Cloud-Serviceadministratoren können viele wiederholbare Aufgaben automatisieren, einschließlich der folgenden:

- Metadaten, Daten, Artefakt- und Anwendungs-Snapshots, Vorlagen und Datenmanagement-Zuordnungen importieren und exportieren
- Dateien in Umgebungen hochladen, Dateien aufführen und Dateien aus dem Service löschen
- Snapshots, Berichte sowie Metadaten- und Datendateien aus dem Service herunterladen
- Geschäftsregeln für Daten ausführen und die Anwendung aktualisieren
- Daten aus einer Datenbank in eine andere kopieren (in der Regel aus einer Block Storage-Datenbank in eine Aggregate Storage-Datenbank oder aus einer Block Storage-Datenbank in eine andere Block Storage-Datenbank)
- Eine Data Management-Batchregel ausführen
- Datenmanagement-Berichte, Provisioning-Bericht und Benutzerauditbericht generieren
- Vorab zugeordnete Saldendaten, Währungskurse, vorab zugeordnete Transaktionen, Saldendaten und Profile in Account Reconciliation importieren
- Profile in eine Periode kopieren, um den Abstimmungsprozess zu initiieren
- Den Berechnungs-Cube von Profitability and Cost Management-Anwendungen bereitstellen
- Point of Views in Enterprise Profitability and Cost Management- und Profitability and Cost Management-Anwendungen löschen und kopieren
- "Oracle Smart View for Office"- oder REST-API-Auslastung in einer Umgebung wiedergeben, um Performancetests unter starker Last zu ermöglichen
- Supplemental Data aus einer Datei in Financial Consolidation and Close importieren

Sie können Skripte zur Durchführung verschiedenster Aufgaben erstellen und die Ausführung der Aufgaben mit einem Scheduler automatisieren. Beispiel: Sie können ein Skript zum Herunterladen des täglichen Wartungsbackups aus Umgebungen erstellen, um lokale Backups Ihrer Artefakte und Daten zu erstellen.



[Tutorial: Informationen zur Ausführung von Planning-Befehlen mit EPM Automate](#)

EPM Automate installieren

Sie installieren EPM Automate, um Befehle auszuführen. Einige Befehle können mit Groovy-Skripten direkt in Oracle Enterprise Performance Management Cloud ausgeführt werden, ohne EPM Automate zu installieren.

Das EPM Automate-Installationsprogramm für Windows, Linux/UNIX und MacOS X steht über die EPM Cloud-Umgebung zur Verfügung.

Da unter Windows 10 und neueren Versionen nur Windows-Administratoren EPM Automate installieren dürfen, kann das Tool nur von diesen installiert und upgegradet werden. Ein Upgrade für EPM Automate kann nur von dem Benutzer, der es installiert hat, oder von einem anderen Windows-Administrator durchgeführt werden.

In diesem Abschnitt finden Sie Informationen zu Folgendem:

- [Kapazitäts- und Portanforderungen](#)
- [Unterstützte Plattformen](#)
- [Java Runtime Environment und EPM Automate](#)
- [OpenJDK verwenden](#)
- [Anweisungen für Windows](#)
- [Anweisungen für Linux/UNIX/MacOS X](#)
- [Serverseitige Ausführung von EPM Automate-Befehlen](#)

Kapazitäts- und Portanforderungen

Da EPM Automate ein Lightweight-Client ist, ist kein großer Client-Footer erforderlich. Die gesamte Verarbeitung findet in Oracle Enterprise Performance Management Cloud statt.

Sie können EPM Automate auf standardmäßigen Clientcomputern, virtuellen Maschinen und Oracle Integration Cloud-Computern installieren, die über eine sichere HTTP-Verbindung auf externe Hosts zugreifen können.

EPM Automate stellt die Verbindung zu EPM Cloud über den standardmäßigen TLS-Port (Port 443) her. Sie müssen keine zusätzlichen ausgehenden Ports für EPM Automate öffnen.

Unterstützte Plattformen

EPM Automate kann auf virtuellen Computern und Oracle Integration Cloud (OIC)-Computern installiert werden, die über eine sichere HTTP-Verbindung auf externe Hosts zugreifen können.

 **Note:**

- EPM Automate darf nur unter 64-Bit-Betriebssystemen verwendet werden, die aktuell vom Betriebssystemhersteller unterstützt werden.
- EPM Automate funktioniert nicht mit SOCKS-Proxy, nur mit HTTP/HTTPS-Proxy.
- EPM Automate unterstützt die Authentifizierungsmechanismen "Basisauthentifizierung", Digest, Kerberos, Negotiate und NTLM für die Verbindung zum Proxyserver.
- EPM Automate kann die Verbindung zu Oracle Enterprise Performance Management Cloud über API-Gateways, z.B. Google APIGEE, IBM Data Power und andere Reverse-Proxy-Server, herstellen. Damit das funktioniert, konfigurieren Sie das Gateway oder den Reverse-Proxy, indem Sie als Ziel die URL Ihrer EPM Cloud-Umgebung ohne einen Kontext wie /epmcloud festlegen. Beispiel: `https://epm-idDomain.epm.dataCenterRegion.oraclecloud.com`. Verwenden Sie dann die Reverse-Proxy-URL anstelle der EPM Cloud-URL im Befehl [login](#). Informationen zur Konfiguration finden Sie in der Dokumentation zu Ihrem Gateway oder Proxyserver.

Stellen Sie bei der Konfiguration der Proxyeinstellungen sicher, dass der Antwortcode bei der Übergabe von EPM Cloud an EPM Automate auf keine Weise verändert wird, damit Antwortcodes wie 200, 206, 400, 404, 500 oder 501 usw. ordnungsgemäß in EPM Automate verarbeitet werden können. Beispiel: Setzen Sie für IBM Datapower den Wert für `proxy HTTP Response` auf ON. Das API-Gateway muss außerdem HTTP-Methoden (GET, POST, PUT, PATCH und DELETE) zulassen.

Auf Linux- und UNIX-Computern sucht EPM Automate nach den folgenden Umgebungsvariablen, um HTTP- oder HTTPS-Proxyeinstellungen zu bestimmen:

- `proxyHost`
- `proxyPort`

Beispiele für HTTP-Proxy-Einstellungen:

```
export proxyHost=host.example.com
export proxyPort=8000
```

Beispiele für HTTPS-Proxy-Einstellungen:

```
export proxyHost=host.example.com
export proxyPort=8080
```

 **Note:**

EPM Automate kann mit dem Authentifizierungsprotokoll OAuth 2.0 auf EPM Cloud-Umgebungen mit OCI (Gen 2) zugreifen (sofern sie für OAuth konfiguriert wurden), um Befehle auszuführen, insbesondere zur Automatisierung der Befehlsausführung.

In Classic-Umgebungen und OCI (Gen 2)-Umgebungen mit Basisauthentifizierung funktioniert EPM Automate nicht mit den SSO-(Identitätsprovider-)Unternehmenszugangsdaten. Da sich Benutzer nicht mit Unternehmenszugangsdaten anmelden können, müssen die Benutzeraccounts für den Zugriff auf EPM Automate in der Identitätsdomain verwaltet werden. Wenn Ihr Abonnement für SSO konfiguriert ist, müssen Sie für EPM Automate-Benutzer auch die Anmeldung mit ihren Zugangsdaten für die Identitätsdomain aktivieren. Informationen hierzu finden Sie unter Anmeldung mit Zugangsdaten für die Identitätsdomain aktivieren in der Dokumentation zum Verwalten von Oracle Cloud Identity Management.

Downloadanweisungen: Clients herunterladen und installieren in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*.

Java Runtime Environment und EPM Automate

Wenn EPM Automate unter Windows installiert wird, wird auch die erforderliche Java Runtime Environment (JRE) installiert. Eine JRE ist jedoch nicht Bestandteil von Installationsprogrammen unter Linux, Unix und macOS X. Für die Nutzung von EPM Automate müssen Sie über Zugriff auf eine JRE-Installation (Version 8 bis 11) verfügen.

Sie sind zur Nutzung von Oracle Java Standard Edition (SE) mit EPM Automate berechtigt, ohne ein Java SE-Abonnement kaufen zu müssen. Weitere Informationen zur Oracle JDK-Lizenzierung mit EPM Automate finden Sie im [Oracle Support-Dokument 1557737.1: "Support Entitlement for Java SE When Used As Part of Another Oracle Product"](#).

OpenJDK verwenden

Sie können auf Linux-, Unix- und MacOS X-Plattformen anstelle von JRE auch OpenJDK Version 14 oder höher verwenden.

OpenJDK, das freie, GPL-lizenzierte, produktionsreife JDK von Oracle, kann unter der folgenden Adresse heruntergeladen werden: <https://openjdk.java.net>. Auf dieser Website finden Sie außerdem Anweisungen zum Installieren von OpenJDK.

Legen Sie vor dem Starten einer EPM Automate-Session die `JAVA_HOME`-Umgebungsvariable so fest, dass sie auf Ihre OpenJDK-Installation verweist:

MacOS X-Beispiel (Bash-Shell angenommen) zur Verwendung von OpenJDK Version 14, installiert in Ihrem Homeverzeichnis.

```
cd ~/
export JAVA_HOME=$(/usr/jdk-14.jdk/Contents/Home)
```

Linux-Beispiel (Bash-Shell angenommen) zur Verwendung von OpenJDK Version 14, installiert in Ihrem Homeverzeichnis.

```
cd ~/
export JAVA_HOME=/openjdk/jdk-14.0.2
```

Anweisungen für Windows

Standardmäßig wird EPM Automate im Verzeichnis `C:/Oracle/EPM Automate` installiert.

So installieren Sie EPM Automate:

1. Greifen Sie auf dem Windows-Computer, auf dem Sie EPM Automate installieren möchten, auf eine Umgebung zu.
2. Greifen Sie auf der Homepage auf **Einstellung und Aktionen** zu, indem Sie auf Ihren Benutzernamen klicken.
3. Klicken Sie auf **Downloads**.
4. Klicken Sie auf der Seite "Downloads" im Abschnitt "EPM Automate" auf **Download für Windows**.
5. Speichern Sie das Installationsprogramm auf Ihrem Computer.
6. Klicken Sie mit der rechten Maustaste auf das Installationsprogramm (`EPM Automate.exe`), und wählen Sie `Als Administrator ausführen aus`.
7. Klicken Sie im Abschnitt für User Account Control auf **Ja**.
8. Befolgen Sie die auf dem Bildschirm angezeigten Prompts, um die Installation abzuschließen.

Anweisungen für Linux/UNIX/MacOS X

Für EPM Automate ist der Zugriff auf ein Deployment einer unterstützten JRE (Version 8 bis 11) erforderlich. Die Umgebungsvariable `JAVA_HOME` muss so festgelegt sein, dass sie auf Ihre JRE-Installation verweist.

So installieren Sie EPM Automate:

1. Greifen Sie auf eine Umgebung zu.
2. Greifen Sie auf der Homepage auf **Einstellung und Aktionen** zu, indem Sie auf Ihren Benutzernamen klicken.
3. Klicken Sie auf **Downloads**.
4. Klicken Sie auf der Seite "Downloads" im Abschnitt "EPM Automate" auf **Download für Linux/macOS X**.
5. Speichern Sie das Installationsprogramm (`EPMAutomate.tar`) in einem Verzeichnis, in dem Sie Berechtigungen zum Lesen, Schreiben und Ausführen haben.

6. Extrahieren Sie die Inhalte des Installationsprogramms, legen Sie die erforderlichen Umgebungsvariablen fest, und führen Sie `epmautomate.sh` aus:

MacOS X-Beispiel (Bash-Shell angenommen) zur Installation und Ausführung aus Ihrem Homeverzeichnis.

```
cd ~/
tar xf path_to_downloaded_EPMAutomate.tar
export JAVA_HOME=$(/usr/libexec/java_home)
export PATH $HOME/epmautomate/bin:$PATH
epmautomate.sh
```

Linux-Beispiel (Bash-Shell angenommen) zur Installation und Ausführung aus Ihrem Homeverzeichnis. Es wird von der JDK-Version 1.8.0_191 ausgegangen.

```
cd ~/
tar xf path_to_downloaded_EPMAutomate.tar
export JAVA_HOME=/opt/jdk1.8.0_191
export PATH ~/Downloads/epmautomate/bin:$PATH
epmautomate.sh
```

Serverseitige Ausführung von EPM Automate-Befehlen

Einige EPM Automate-Befehle können über Groovy direkt in Oracle Enterprise Performance Management Cloud ausgeführt werden. Sie müssen EPM Automate nicht installieren, um Befehle mit Groovy-Skripten auszuführen.

Beachten Sie, dass zum Ausführen von EPM Automate-Befehlen die serverseitige Ausführung von Befehlen nicht mit dem Ausführen von Groovy-Skripten auf einem Clientcomputer identisch ist.

Ausführliche Informationen finden Sie unter [Befehle ohne Installation von EPM Automate ausführen](#).

Erläuterungen zur Verschlüsselungsebene für EPM Automate

Oracle Enterprise Performance Management Cloud verwendet Transport Layer Security (TLS) mit kryptografischem Hash-Algorithmus SHA-2/SHA-256, um die Kommunikation mit EPM Automate zu sichern.

Autorisierungsprotokoll OAuth 2.0 bei OCI (Gen2)-Umgebungen verwenden

EPM Automate kann mit dem Authentifizierungsprotokoll OAuth 2.0 auf Oracle Enterprise Performance Management Cloud-Umgebungen mit OCI (Gen 2) zugreifen, um Befehle auszuführen, insbesondere zur Automatisierung der Befehlsausführung.

Um den Zugriff auf OAuth 2.0 zu aktivieren, muss ein Identitätsdomainadministrator Ihre Anwendung in den Oracle Cloud Identity-Services als öffentlichen Client

registrieren. OAuth wird für die Anwendung durchgesetzt, nicht für Ihr gesamtes Abonnement.

Ausführliche Anweisungen zur Einrichtung von OAuth 2.0 für Ihre OCI (Gen 2)-Umgebungen finden Sie unter Authentifizierung mit OAuth 2 - nur OCI-Umgebungen (2. Generation) in *REST-API für Oracle Enterprise Performance Management Cloud*.



Note:

Die Basisauthentifizierung funktioniert auch, wenn OAuth für eine Umgebung aktiviert ist. Stellen Sie sicher, dass die vorhandene verschlüsselte Kennwortdatei nicht überschrieben wird, wenn Sie sie zu einem späteren Zeitpunkt verwenden möchten.

Verschlüsselte Kennwortdatei mit Aktualisierungstoken und Client-ID erstellen

Serviceadministratoren, die OAuth 2.0 für den EPM Automate-Zugriff auf Umgebungen verwenden möchten, benötigen diese Details für die Erstellung ihrer verschlüsselten Kennwortdatei. Diese wird dann für die Anmeldung bei der Umgebung verwendet:

- **Aktualisierungstoken**
Ausführliche Anweisungen zum Abrufen des Aktualisierungstokens finden Sie in den Schritten im Abschnitt zu den Serviceadministratortasken zum Abrufen von Aktualisierungstoken in EPM Cloud unter Authentifizierung mit OAuth 2 - nur OCI-Umgebungen (2. Generation) in der Dokumentation *REST-API für Oracle Enterprise Performance Management Cloud*.
- **Client-ID**
Die Client-ID wird bei der Konfiguration der Anwendung für OAuth durch den Identitätsdomainadministrator generiert. Sie kann in der Registerkarte "Konfiguration" der Anwendung unter **Allgemeine Informationen** angezeigt werden.

So erstellen Sie die verschlüsselte Kennwortdatei für die OAuth-Authentifizierung:

1. Starten Sie eine EPM Automate-Session.
2. Führen Sie einen Befehl ähnlich dem folgenden aus:

```
epmautomate encrypt REFRESH_TOKEN ENCRYPTION_KEY PASSWORD_FILE
```


ClientID=CLIENT_ID, wobei REFRESH_TOKEN das entschlüsselte Aktualisierungstoken aus dem sicheren Speicher, ENCRYPTION_KEY ein beliebiger persönlicher Schlüssel zur Verschlüsselung des Kennworts und PASSWORD_FILE der Name und der Speicherort der Datei ist, in der das verschlüsselte Aktualisierungstoken gespeichert ist. Die Kennwortdatei muss die Erweiterung .epw aufweisen.

Detaillierte Anweisungen finden Sie unter [encrypt](#).

3. Verwenden Sie die neu generierte Kennwortdatei, um sich mit OAuth anzumelden. Stellen Sie bei automatisierten Skriptausführungen sicher, dass Sie die Skripte so aktualisieren, dass sie auf die neu generierte Kennwortdatei verweisen.

2

Befehle

- [Informationen zur Ausführung von EPM Automate-Befehlen](#)
- [EPM Automate ausführen](#)
- [Befehle auf einen Blick](#)
- [EPM Automate-Befehle](#)
- [Exit-Codes](#)

Einige EPM Automate-Befehle gelten für alle Geschäftsprozesse während andere für eine Gruppe von Geschäftsprozessen gelten. Sofern nichts Gegenteiliges angegeben ist, funktioniert ein Befehl, der für einen bestimmten Geschäftsprozess gilt (z.B. Planning), nicht mit einem anderen Geschäftsprozess (z.B. Financial Consolidation and Close). Versuche, einen Befehl für einen Geschäftsprozess auszuführen, der diesen nicht unterstützt, führen zu einem Fehler. Eine Liste der für die einzelnen Geschäftsprozesse gültigen Befehle finden Sie unter [Spezifische Befehle für die einzelnen EPM Cloud-Service](#).

Informationen zur Ausführung von EPM Automate-Befehlen

Alle Oracle Enterprise Performance Management Cloud-Services verwenden EPM Automate-Befehle zur Remoteverwaltung von Umgebungen.

- [Voraussetzungen](#)
- [Standarddateispeicherorte](#)
- [TLS-Protokoll 1.2 aktivieren](#)
- [EPM Automate-Befehle verwenden](#)
- [Mehrere Werte für Parameter angeben](#)
- [Verhalten während der täglichen Wartung](#)

Voraussetzungen

In diesem Abschnitt werden die Voraussetzung für die Verwendung von EPM Automate aufgelistet, etwa die Verwendung von Oracle Enterprise Performance Management Cloud-Zugangsdaten und -Standardspeicherpositionen für Dateien in Umgebungen.

Allgemein

Alle Benutzer von EPM Cloud können ihre Zugangsdaten für die Identitätsdomain verwenden, um mit EPM Automate eine Verbindung zu einer Umgebung herzustellen. Die vordefinierten Rollen und Anwendungsrollen, die dem Benutzer zugeordnet sind, legen die Befehle fest, die ein Benutzer ausführen kann.

- Außerdem ist die Rolle Identitätsdomainadministrator erforderlich, um Befehle auszuführen, mit denen Benutzer in der Identitätsdomain hinzugefügt oder gelöscht werden.

- Alle Dateien, die zur Ausführung eines Befehls erforderlich sind, müssen innerhalb der Umgebung vorhanden sein. Verwenden Sie den Befehl `uploadFile`, um Dateien hochzuladen.

Weitere Informationen zum Standarddateispeicherort, der von jedem Service verwendet wird, finden Sie unter [Standarddateispeicherorte](#).

- Verwendung von Dateierweiterungen in Befehlen:
 - Geben Sie den vollständigen Dateinamen einschließlich der Dateierweiterung (z.B. `data.csv`) an, um Befehle auszuführen, die Dateivorgänge durchführen. Beispiele für Befehle für Dateivorgänge sind `deletefile`, `listfiles` und `uploadfile`.
 - Verwenden Sie keine Dateierweiterungen, um Befehle auszuführen, die Vorgänge der Migration durchführen. Für Vorgänge der Migration müssen Sie den Namen eines Snapshots angeben.
- Parameterwerte, die ein Leerzeichen enthalten, wie Kommentare, Speicherortnamen und Ordnerpfade müssen in Anführungszeichen stehen.

Planning

- Jobs
Viele der im folgenden Abschnitt beschriebenen Befehle erfordern Jobs. Jobs sind Aktionen, die sofort gestartet oder für einen späteren Zeitpunkt geplant werden können. Beispiel: Daten importieren oder exportieren, Datenbank aktualisieren.
Um die folgenden Vorgänge auszuführen, müssen Sie entsprechende Jobs mit der Jobkonsole erstellen. Ausführliche Anweisungen zum Erstellen von Jobs in Planning finden Sie unter "Jobs verwalten" in der Dokumentation *Planning verwalten*.
 - Daten in eine Anwendung importieren
 - Daten aus einer Anwendung exportieren
 - Metadaten in eine Anwendung importieren
 - Metadaten aus einer Anwendung exportieren
 - Daten aus einer Block Storage-Datenbank in eine Aggregate Storage-Datenbank oder in eine andere Block Storage-Datenbank kopieren
- Geschäftsregeln
Geschäftsregeln, die Sie ausführen möchten, müssen in der Anwendung vorhanden sein.
Mit Calculation Manager können Sie Geschäftsregeln erstellen, die anschließend in der Anwendung bereitgestellt werden. Informationen hierzu finden Sie unter *Mit Calculation Manager für Oracle Enterprise Performance Management Cloud entwerfen*.

Datenmanagement

- Datenregeln
Dataload-Regeln definieren, wie Data Management Daten aus einer Datei lädt. Damit Sie Daten mit EPM Automate laden können, müssen vordefinierte Dataload-Regeln vorhanden sein.
- Batches

Sie können Daten mit in Data Management definierten Batches laden. Mit einem Batch können Benutzer viele Laderegeln in einem Batch kombinieren und diese Regeln seriell oder parallel ausführen.

Standarddateispeicherorte

Standardspeicherort zum Hochladen

Standardmäßig werden alle in Oracle Enterprise Performance Management Cloud hochgeladenen Dateien an einen Standardspeicherort hochgeladen, auf den über Migration zugegriffen werden kann.

Sie müssen Dateien hochladen, die mit Migration verarbeitet werden sollen, z.B. Snapshots, die Sie in den Service importieren und dort am Standardspeicherort ablegen möchten.

Inbox und Outbox

Die Speicherorte für Inbox und Outbox können für verschiedene EPM Cloud-Geschäftsprozesse unterschiedlich sein. Sie verwenden die Inbox, um Dateien hochzuladen, die Sie importieren oder anderweitig mit anderen Geschäftsprozessen als Profitability and Cost Management-Prozessen verarbeiten möchten. Die Komponente Datenmanagement kann Dateien in der Inbox oder in einem darin enthaltenen Verzeichnis verarbeiten.

In der Regel speichert EPM Cloud Dateien, die Sie über die Geschäftsprozesse erstellen, z.B. Exportdateien für Daten oder Metadaten, in der Outbox.

- Der Zugriff auf die Inbox, in die EPM Automate Dateien hochlädt, und die Outbox, in der Dateien für den Download gespeichert werden, erfolgt über diese Anwendungen. Sie müssen an diesen Speicherort Dateien hochladen, wenn Sie diese mit einem Prozess verarbeiten möchten, der zu diesen Anwendungen gehört. Sie können auch Dateien in die Outbox hochladen.
 - Planning
 - Planning Modules
 - Account Reconciliation
 - Financial Consolidation and Close
 - Tax Reporting
 - Narrative Reporting
 - Enterprise Profitability and Cost Management

Sie können den Explorer für Inbox/Outbox verwenden, um die Dateien zu durchsuchen, die am Standardspeicherort gespeichert sind. Anwendungs-Snapshots, die Sie mit EPM Automate erstellen, werden nicht im Explorer für Inbox/Outbox aufgeführt. Sie können sie in der Registerkarte "Snapshots" von Migration anzeigen.

- Dateien, die mit einem Profitability and Cost Management-Prozess verarbeitet werden sollen, müssen in `profitinbox` hochgeladen werden. Sie können auch Dateien in `profitoutbox` hochladen. Dateien, die von Profitability and Cost Management-Prozessen exportiert wurden, werden in `profitinbox` gespeichert. Verwenden Sie den Dateixplorer, um diese Dateien zu durchsuchen.
- Dateien, die mit Data Management verarbeitet werden sollen, müssen in der Inbox oder in einem zugehörigen Unterordner verfügbar sein. Standardmäßig werden Dateien, die mit der Komponente Datenmanagement exportiert wurden, in der Outbox gespeichert. Datenmanagement-Berichtsausgaben werden hingegen in der Komponente

Datenmanagement im Ordner `outbox/report` gespeichert. Verwenden Sie den Dateibrowser von Data Management, um diese Dateien zu durchsuchen.

- Oracle Enterprise Data Management Cloud verwendet den Standardspeicherort für den Import und Export von Dateien, die hochgeladen, kopiert oder heruntergeladen werden. Dateien im Standardspeicherort können mit dem Befehl `ListFiles` angezeigt werden.

Logdateien

Bei jeder Ausführung eines EPM Automate-Befehls wird eine Debuggingdatei erstellt, die automatisch gelöscht wird, wenn der Befehl erfolgreich ist. Wenn bei der Ausführung des Befehls ein Fehler auftritt, wird die Debuggingdatei für den fehlgeschlagenen Befehl in dem Verzeichnis beibehalten, von dem aus EPM Automate ausgeführt wird. Standardmäßig ist dies das Verzeichnis `Oracle/epm_automate/bin` (unter Windows) oder `home/user/epmautomate/bin` (unter Linux/UNIX).

EPM Automate-Debuggingdateien verwenden die folgende Namenskonvention:

`commandname_date_timestamp.log`. Wenn Sie z.B. am 23. November 2020 um 09:28:02 einen `listfiles`-Befehl ausführen, der nicht erfolgreich ist, lautet die Debuggingdatei `listfiles_23_11_2020_09_28_02.log`.

Die Erstellung der Debuggingdatei für einen nicht erfolgreichen Befehl kann nicht deaktiviert werden. Sie können aber Debugginginformationen und die Befehlsausgabe in eine Datei in einem anderen Verzeichnis schreiben, indem Sie am Ende des Befehls `-d` zusammen mit einem Debuggingdateinamen sowie Fehler und Outputstreams anhängen (`-d >> c:\logs\LOG_FILE_NAME.log 2>&1`), wie im folgenden Windows-Beispiel gezeigt:

```
epmautomate listfiles -d >> c:\logs\listfiles.log 2>&1
```

TLS-Protokoll 1.2 aktivieren

EPM Automate muss unter einem Betriebssystem installiert sein, das das TLS- (Transport Layer Security-)Protokoll 1.2 oder höher unterstützt.

Um die höchste Sicherheitsebene bei der Authentifizierung und Datenverschlüsselung sicherzustellen, unterstützt EPM Automate nur TLS 1.2. Wenn TLS 1.2 auf dem Computer, auf dem EPM Automate ausgeführt wird, nicht installiert ist, wird der Fehler `EPMAT-7: Unable to connect. Unsupported Protocol: HTTPS` (Verbindung nicht möglich. Nicht unterstütztes Protokoll: HTTPS) angezeigt. Wenden Sie sich zur Behebung dieses Fehlers an Ihren IT-Administrator, um TLS 1.2 zu aktivieren.

Die Vorgehensweise zur Aktivierung von TLS 1.2 ist vom Betriebssystem abhängig. Verwenden Sie die folgenden Informationsquellen. Ähnliche Webressourcen sind möglicherweise auch für andere unterstützte Betriebssysteme verfügbar:

- [Update to enable TLS 1.1 and TLS 1.2 as default secure protocols in WinHTTP in Windows](#) (Update zum Aktivieren von TLS 1.1 und TLS 1.2 als sichere Standardprotokolle in WinHTTP unter Windows) mit Informationen zum Aktivieren von TLS 1.2 für Windows-Computer
- [Hardening TLS Configuration](#) (Sicherheit der TLS-Konfiguration erhöhen) mit Informationen zum Aktivieren von TLS 1.2 in OpenSSL für Red Hat Enterprise Linux

EPM Automate-Befehle verwenden

Abfolge von Befehlsparametern

Alle erforderlichen Parameter für einen Befehl müssen in der Abfolge gemäß Befehlsverwendung übergeben werden. Erforderliche Parameter und die zugehörigen Werte haben Vorrang vor optionalen Parametern, die in einer beliebigen Abfolge übergeben werden können. Optionale Parameter sind nicht positionsabhängig.

Beispiel: Betrachten Sie die folgende Verwendung des Befehls `login`:

```
epmautomate login USERNAME PASSWORD URL [ProxyServerUserName=PROXY_USERNAME]  
[ProxyServerPassword=PROXY_PASSWORD] [ProxyServerDomain=PROXY_DOMAIN]
```

Dieser Befehl hat drei erforderliche Parameter: `USERNAME`, `PASSWORD` und `URL`. Diese Parameter müssen der Abfolge in der Verwendung entsprechen. Der Befehl gibt einen Fehler zurück, wenn diese Abfolge nicht beibehalten wird. Optionale Parameter `ProxyServerUserName`, `ProxyServerPassword` und `ProxyServerDomain` und die zugehörigen Werte können in beliebiger Abfolge angegeben werden.

Muss bei EPM Automate-Befehlen die Groß-/Kleinschreibung beachtet werden?

Bei EPM Automate-Befehlen muss die Groß-/Kleinschreibung nicht beachtet werden. Wie ein Befehlsname eingegeben wird, hat keine Auswirkung auf die Befehlsausführung. Beispiel: Für den Befehl `addUsers` können Sie `addusers`, `ADDUSERS` oder `AdDuSeRs` eingeben.

Muss bei EPM Automate-Befehlsparametern die Groß-/Kleinschreibung beachtet werden?

Bei EPM Automate-Befehlsparametern muss die Groß-/Kleinschreibung nicht beachtet werden. Wie ein Befehlsparametername eingegeben wird, hat keine Auswirkung auf die Befehlsausführung. Beispiel: Sie können den Parameter `fileName` als `filename`, `fileName` oder `fileName` eingeben, ohne dass dies Auswirkungen auf die Befehlsausführung hat.

Mehrere Werte für Parameter angeben

Einige EPM Automate-Befehle akzeptieren mehrere durch Komma getrennte Parameterwerte, z.B. einen Runtime Prompt vom Typ "Elemente" in Geschäftsregeln, Regelsets und Vorlagen in einer Planning-Anwendung.

Um mehrere Elemente für einen Runtime Prompt vom Typ "Elemente" mit dem Namen `Entities` in einem EPM Automate-Befehl festzulegen, verwenden Sie ein `,` (Komma), wie im folgenden Beispiel zur Ausführung des Befehls `runbusinessrule` veranschaulicht.

```
epmautomate runbusinessrule clearDistData TargetYear=FY19  
TargetMonth=Feb Entities=District1,District2
```

Elementnamen, die Sonderzeichen wie Leerzeichen und Komma enthalten, müssen in Anführungszeichen eingeschlossen und mit `\` (umgekehrter Schrägstrich) maskiert werden, wie im folgenden Beispiel gezeigt:

```
epmautomate runbusinessrule clearDistData TargetYear=FY19  
TargetMonth=Feb Entities="\\"District 1\\",\\"entity_name, with  
comma\\""
```


Verhalten während der täglichen Wartung

Führen Sie keine EPM Automate-Befehle aus, während die tägliche Wartung einer Umgebung läuft.

Während der täglichen Wartung sind keine Benutzeraktivitäten zulässig. Versuche, EPM Automate-Befehle direkt oder mit Skripten auszuführen, während die tägliche Wartung läuft, führen zur Anzeige des folgenden Fehlers:

```
EPMAT-11:Internal server error. Due to the daily maintenance,
your Oracle EPM Cloud Service environment is currently
unavailable. (EPMAT-11: Interner Serverfehler. Ihre Oracle EPM Cloud Service-
Umgebung ist aufgrund der täglichen Wartung derzeit nicht verfügbar.)
```

EPM Automate ausführen

Mit Ihren Oracle Enterprise Performance Management Cloud-Zugangsdaten können Sie sich mit EPM Automate anmelden. Sie können Ihre SSO-Zugangsdaten nicht für die Anmeldung verwenden.

Alle Benutzer von EPM Cloud können ihre Zugangsdaten für die Identitätsdomain verwenden, um mit EPM Automate eine Verbindung zu einer Umgebung herzustellen. Die vordefinierten Rollen und Anwendungsrollen, die dem Benutzer zugeordnet sind, legen die Befehle fest, die ein Benutzer ausführen kann.

Darüber hinaus können nur Serviceadministratoren einige Befehle ausführen, während für die Ausführung anderer Befehle auch die Rolle Identitätsdomainadministrator erforderlich sein kann.

Debugginglogdatei generieren

Oracle Support bittet Sie um eine Debugginglogdatei der Session, um Probleme zu beheben, die beim Ausführen von EPM Automate aufgetreten sind. EPM Automate unterstützt die Option `-d`, um Debuggingmeldungen zu generieren, die dann anhand der Anweisung `>` zu einer Datei umgeleitet werden können. Sie können eine Debuggingdatei für einen Befehl, eine Batchausführungsdatei oder ein Skript mit mehreren Befehlen erstellen.

Verwendung: `epmautomate command [command_parameters] -d > log_file 2>&1`

Windows-Beispiel: `epmautomate downloadfile "Artifact Snapshot" -d > C:\logs\download_log.txt 2>&1`

Linux-Beispiel: `epmautomate.sh downloadfile "Artifact Snapshot" -d > ./logs/download_log 2>&1`

Windows

Stellen Sie vor dem Ausführen von EPM Automate sicher, dass Sie von dem Computer, auf dem Sie EPM Automate ausführen, auf Ihre Umgebung zugreifen können.

EPM Automate erstellt eine `.prefs`-Datei, die Benutzerinformationen enthält, sowie Logdateien im aktuellen Verzeichnis. Auf Windows-Computern wird der Inhalt der `.prefs`-Datei nur dem Benutzer, der die Datei erstellt hat, sowie Windows-

Administratoren angezeigt. In Linux-, UNIX- und macOSX-Umgebungen wird die `.prefs`-Datei mit der Berechtigung 600 generiert, d.h., nur der Eigentümer kann diese Datei lesen und darin schreiben.

EPM Automate zeigt den Fehler `FileNotFoundException: .prefs (Access is denied)` in Windows-Umgebungen an, wenn Sie in dem Windows-Verzeichnis, in dem Sie EPM Automate ausführen, nicht über die Schreibberechtigung verfügen. Stellen Sie zum Beheben dieses Fehlers sicher, dass das Windows-Konto des aktuellen Benutzers über Lese-/Schreibzugriff auf das Verzeichnis verfügt, über das EPM Automate ausgeführt wird. Darüber hinaus muss dieser Benutzer über den entsprechenden Zugriff auf jedes andere Verzeichnis verfügen, in dem auf eine Datei zugegriffen wird (z.B. während der Ausführung des Befehls `uploadFile`) oder in das geschrieben wird (z.B. während der Ausführung des Befehls `downloadFile`).

 **Hinweis:**

EPM Automate kann von keinem Ordner aus ausgeführt werden, dessen Name ein `&` enthält. Beispiel: `C:\Oracle\A&B`.

So führen Sie EPM Automate auf einem Windows-Client aus:

1. Klicken Sie auf **Start, Alle Programme, EPM Automate, EPM Automate starten**. Der Befehls-Prompt für EPM Automate wird angezeigt.
2. **Optional:** Navigieren Sie zu dem Verzeichnis, aus dem Sie Vorgänge mit EPM Automate ausführen möchten.
3. **Optional:** Generieren Sie eine Kennwortverschlüsselungsdatei. Mit der Kennwortverschlüsselungsdatei können Sie ein verschlüsseltes Kennwort übergeben, um eine Session zu initiieren.

```
epmautomate encrypt P@ssword1 myKey C:/mySecuredir/password.epw
```

4. Starten Sie eine Session als Serviceadministrator. Verwenden Sie z.B. folgenden Befehl:

- Unverschlüsseltes Kennwort verwenden:

```
epmautomate login serviceAdmin P@ssword1  
https://test-cloudpln.pbcs_us1.oraclecloud.com
```

- Verschlüsseltes Kennwort verwenden:

```
epmautomate login serviceAdmin C:\mySecuredir\password.epw  
https://test-cloudpln.pbcs_us1.oraclecloud.com
```

5. Geben Sie Befehle ein, mit denen die abzuschließenden Aufgaben ausgeführt werden sollen.

Informationen zum Ausführungsstatus von Befehlen finden Sie unter [Exit-Codes](#).

6. Melden Sie sich von der Umgebung ab. Verwenden Sie den folgenden Befehl:

```
epmautomate logout
```

Linux

Hinweis:

Stellen Sie sicher, dass `JAVA_HOME` in der Variablen `PATH` für die `PROFILE`-Datei oder als Shellumgebungsvariable eingestellt ist. Eine unterstützte JRE (Version 8 bis 11) ist erforderlich.

So führen Sie EPM Automate auf einem Linux-Client aus:

1. Öffnen Sie ein Terminalfenster, und navigieren Sie zu dem Verzeichnis, in dem EPM Automate installiert ist.
2. **Optional:** Generieren Sie eine Kennwortverschlüsselungsdatei. Mit der Kennwortverschlüsselungsdatei können Sie anstelle eines unverschlüsselten Kennworts ein verschlüsseltes Kennwort übergeben, um eine Session zu initiieren.

```
epmautomate encrypt P@ssword1 myKey ../misc/encrypt/password.epw
```

3. Starten Sie eine Session als Serviceadministrator. Verwenden Sie z.B. folgenden Befehl:

- Unverschlüsseltes Kennwort verwenden:

```
./bin/epmautomate.sh login serviceAdmin P@ssword1  
https://test-cloudpln.pbcs_us1.oraclecloud.com
```

- Verschlüsseltes Kennwort verwenden:

```
./bin/epmautomate.sh login serviceAdmin ../misc/encrypt/  
password.epw  
https://test-cloudpln.pbcs_us1.oraclecloud.com
```

4. Geben Sie Befehle ein, mit denen die abzuschließenden Aufgaben ausgeführt werden sollen.

Informationen zum Ausführungsstatus von Befehlen finden Sie unter [Exit-Codes](#).

5. Melden Sie sich von der Umgebung ab. Verwenden Sie den folgenden Befehl:

```
./bin/epmautomate.sh logout
```

Mehrere Instanzen von EPM Automate ausführen

Sie können mehrere Instanzen von EPM Automate in einer Umgebung aus demselben Verzeichnis ausführen. Ebenso können Sie mehrere Instanzen von EPM Automate in verschiedenen Umgebungen aus demselben oder verschiedenen Verzeichnissen ausführen.

Beispiel: Sie müssen den Cube der Planning-Anwendung unter `https://cloudpln.pbcs.us1.oraclecloud.com` und `https://`

testcloudpln.pbcsl.us1.oraclecloud.com möglicherweise gleichzeitig aktualisieren. In diesem Szenario haben Sie zwei Möglichkeiten:

- Zwei Instanzen von EPM Automate aus demselben Verzeichnis ausführen, um Anwendungs-Cubes in verschiedenen Umgebungen zu aktualisieren
- EPM Automate aus separaten Verzeichnissen ausführen, um eine Verbindung zu den Umgebungen herzustellen und dann die Anwendungs-Cubes zu aktualisieren

In beiden Szenarios arbeiten die Instanzen von EPM Automate unabhängig voneinander. Beim Abmelden von einer Instanz werden Sie nicht von anderen Instanzen abgemeldet. Aktivitäten, die mit EPM Automate initiiert wurden, werden in der Umgebung weiterhin bis zum Abschluss ausgeführt, auch wenn Sie sich von der anderen Instanz abmelden.

Dieser Abschnitt enthält Beispielskripte für Windows und Unix/Linux (`caller` und `multisession`), mit denen zwei EPM Automate-Sessions zum Ausführen von Aufgaben erstellt werden können. Um mehrere Sessions gleichzeitig auszuführen, müssen Sie folgende Verbindungsinformationen im `caller`-Skript hinzufügen. Damit wird das `multisession`-Skript zum Ausführen der Befehle `login`, `uploadfile`, `listfiles` und `logout` aufgerufen. Sie können das `multisession`-Skript ändern, um damit andere Aufgaben als diese auszuführen. Stellen Sie sicher, dass beide Skripte in demselben Verzeichnis gespeichert sind.

- EPM Automate verwendet die Umgebungsvariable `EPM_SID`, um mehrere Sessions zu unterscheiden. Diese Variable muss im Aufruferskript für jede Session auf einen eindeutigen Wert gesetzt sein. In den Beispielskripten ist sie wie folgt auf eindeutige Werte gesetzt:
 - In `caller.BAT` ist `EPM_SID` auf `!RANDOM!` gesetzt. Dadurch wird der ID eine eindeutige systemgenerierte Zahl zugewiesen. Diese Zahl wird auch zum Generieren der Logdateien für jede Session verwendet. Wenn Sie die Logdatei für jede Session verfolgen möchten, können Sie eine eindeutige Zahl anstelle von `!RANDOM!` angeben.
 - In `caller.sh`, ist `EPM_SID` auf die Prozess-ID gesetzt, die eindeutig ist. Wenn Sie die Logdatei für jede Session verfolgen möchten, können Sie einen eindeutigen Wert für `EPM_SID` angeben, indem Sie die Anweisung `export EPM_SID=$$` im `multisession`-Skript ändern, sodass der übergebene Wert verwendet wird, und dann einen eindeutigen Wert für diesen Parameter in das `caller`-Skript für jede Session übergeben, indem Sie z.B. wie folgt den Wert von `EPM_SID` in `caller.sh` angeben:

```

$SCRIPT_DIR/multisession.sh EPM_SID "USERNAME" "PASSWORD" "URL" "/
home/user/Snapshot1.zip" &
$SCRIPT_DIR/multisession.sh EPM_SID "USERNAME" "PASSWORD" "URL" "/
home/user/Snapshot2.zip" &

```

- `USERNAME`: Anmelde-ID eines Serviceadministrators
- `PASSWORD`: Kennwort des Serviceadministrators
- `URL`: Verbindungs-URL der Umgebung

Windows-Beispielskripte

caller.BAT

```

@echo off
setlocal EnableExtensions EnableDelayedExpansion

```

```

REM syntax: start /B multisession.bat EPM_SID "USERNAME" "PASSWORD"
"URL" "SNAPSHOTPATH"
start /B multisession.bat !RANDOM! "USERNAME" "PASSWORD" "URL"
"C:\Snapshot1.zip"
start /B multisession.bat !RANDOM! "USERNAME" "PASSWORD" "URL"
"C:\Snapshot2.zip"

endlocal

```

multisession.BAT

```

@echo off

set EPM_SID=%1
set USERNAME=%2
set PASSWORD=%3
set URL=%4
set SNAPSHOTNAME=%5

echo User: %USERNAME% > %EPM_SID%.log
echo Cloud Instance: %URL% >> %EPM_SID%.log

call epmautomate login %USERNAME% %PASSWORD% %URL% >> %EPM_SID%.log
call epmautomate uploadfile %SNAPSHOTNAME% >> %EPM_SID%.log
call epmautomate listfiles >> %EPM_SID%.log
call epmautomate logout

```

Bourne-Shell-Beispielskript

caller.sh

```

#!/bin/sh

set +x
SCRIPT_DIR=`dirname "${0}"`

# syntax: /home/user/multisession.sh "USERNAME" "PASSWORD" "URL"
"SNAPSHOTPATH" &
$SCRIPT_DIR/multisession.sh "USERNAME" "PASSWORD" "URL" "/home/user/
Snapshot1.zip" &
$SCRIPT_DIR/multisession.sh "USERNAME" "PASSWORD" "URL" "/home/user/
Snapshot2.zip" &

```

multisession.sh

```

#!/bin/sh

set +x

EPM_AUTOMATE_HOME=/home/user/epmautomate

export JAVA_HOME=/home/user/jre
export EPM_SID=$$

```

```

USERNAME=$1
PASSWORD=$2
URL=$3
SNAPSHOTNAME=$4

echo User: $USERNAME > $EPM_SID.log
echo Cloud Instance: $URL >> $EPM_SID.log

$EPM_AUTOMATE_HOME/bin/epmautomate.sh login $USERNAME $PASSWORD $URL
>> $EPM_SID.log
$EPM_AUTOMATE_HOME/bin/epmautomate.sh uploadfile $SNAPSHOTNAME
>> $EPM_SID.log
$EPM_AUTOMATE_HOME/bin/epmautomate.sh listfiles >> $EPM_SID.log
$EPM_AUTOMATE_HOME/bin/epmautomate.sh logout

```

Befehle auf einen Blick

Dies ist eine alphabetische Liste aller EPM Automate-Befehle.

Tabelle 2-1 Alle EPM Automate-Befehle

Befehlsname	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
addUsers	✓	✓	✓	✓	✓	✓	✓	✓
addUsersToGroup	✓	✓	✓	✓	✓	✓	✓	✓
addUsersToTeam		✓	✓			✓		
addUserToGroups	✓	✓	✓	✓	✓	✓	✓	✓
applicationAdminMode	✓	✓	✓		✓			
applyDataGrants				✓				
archiveTmTransactions						✓		
assignRole	✓	✓	✓	✓	✓	✓	✓	✓
autoPredict* siehe Fußnote	✓							
calculateModel					✓			
clearCube	✓				✓			
clearDataByPointOfView					✓			
clearDataByProfile		✓	✓					
clearPOV				✓				
cloneEnvironment	✓	✓	✓	✓	✓	✓	✓	✓
copyDataByPointOfView					✓			
copyDataByProfile		✓	✓					
copyFileFromInstance	✓	✓	✓	✓	✓	✓	✓	✓
copyFromObjectStorage	✓	✓	✓	✓	✓	✓	✓	✓
copyOwnershipDataToNextYear		✓	✓					
copyPOV				✓				
copySnapshotFromInstance	✓	✓	✓	✓	✓	✓	✓	
copyToObjectStorage	✓	✓	✓	✓	✓	✓	✓	✓

Tabelle 2-1 (Fortsetzung) Alle EPM Automate-Befehle

Befehlsname	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
createGroups	✓	✓	✓	✓	✓	✓	✓	✓
createNRSnapshot								✓
createReconciliations						✓		
deleteFile	✓	✓	✓	✓	✓	✓	✓	✓
deleteGroups	✓	✓	✓	✓	✓	✓	✓	✓
deletePointOfView					✓			
deletePOV				✓				
deployCube				✓				
deployEJTemplates		✓						
deployFormTemplates		✓	✓					
deployTaskManagerTemplate		✓						
dismissIPMInsights**	✓							
downloadFile	✓	✓	✓	✓	✓	✓	✓	✓
enableApp				✓				
enableQueryTracking	✓				✓			
encrypt	✓	✓	✓	✓	✓	✓	✓	✓
executeAggregationProcess	✓				✓			
executeBurstDefinition								✓
executeReportBurstingDefinition	✓	✓	✓		✓			
exportAccessControl						✓		
exportAppAudit	✓	✓	✓		✓			
exportAppSecurity	✓	✓	✓		✓			
exportARApplicationProperties						✓		
exportBackgroundImage						✓		
exportCellLevelSecurity	✓		✓		✓			
exportData	✓	✓	✓		✓			
exportConsolidationJournals		✓						
exportDataManagement	✓	✓	✓	✓	✓			
exportDimension							✓	
exportDimensionMapping							✓	
exportEJJournals		✓						
exportEssbaseData	✓	✓	✓		✓			
exportJobConsole	✓	✓	✓		✓			
exportLibraryArtifact								✓
exportLibraryDocument	✓	✓	✓		✓			
exportLogoImage						✓		
exportMapping	✓	✓	✓	✓	✓	✓		
exportMetadata	✓	✓	✓		✓			
exportOwnershipData		✓	✓					

Tabelle 2-1 (Fortsetzung) Alle EPM Automate-Befehle

Befehlsname	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
exportQueryResults				✓				
exportSnapshot	✓	✓	✓	✓	✓	✓	✓	
exportTemplate				✓				
exportTaskManagerAccessControl		✓	✓					
exportValidIntersections	✓	✓	✓		✓			
extractDimension							✓	
feedback	✓	✓	✓	✓	✓	✓	✓	✓
getApplicationAdminMode	✓	✓	✓		✓	✓		
getDailyMaintenanceStartTime	✓	✓	✓	✓	✓	✓	✓	✓
getEssbaseQryGovExecTime	✓	✓	✓	✓	✓			
getIdleSessionTimeout	✓	✓	✓	✓	✓	✓	✓	✓
getIPAllowlist	✓	✓	✓	✓	✓	✓	✓	✓
getSubstVar	✓	✓	✓		✓			
getVirusScanOnFileUploads	✓	✓	✓	✓	✓	✓	✓	✓
groupAssignmentAuditReport	✓	✓	✓	✓	✓	✓	✓	✓
help	✓	✓	✓	✓	✓	✓	✓	✓
importAppAudit	✓				✓			
importAppSecurity	✓	✓	✓		✓			
importARApplicationProperties						✓		
importBackgroundImage						✓		
importBalances						✓		
importCellLevelSecurity	✓		✓		✓			
importConsolidationJournals		✓						
importData	✓	✓	✓	✓	✓			
importDataManagement	✓	✓	✓	✓	✓			
importDimension							✓	
importJobConsole	✓	✓	✓		✓			
importLibraryArtifact								✓
importLogoImage						✓		
importMapping	✓	✓	✓	✓	✓	✓		
importMetadata	✓	✓	✓		✓			
importOwnershipData		✓	✓					
importPreMappedBalances						✓		
importPreMappedTransactions						✓		
importProfiles						✓		
importRates						✓		
importRCAAttributeValues						✓		
importReconciliationAttributes						✓		
importSnapshot	✓	✓	✓	✓	✓	✓	✓	

Tabelle 2-1 (Fortsetzung) Alle EPM Automate-Befehle

Befehlsname	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
importSupplementalCollectionData		✓	✓					
importSupplementalData		✓	✓					
importTemplate				✓				
importTMAtributeValues						✓		
importValidIntersections	✓	✓	✓		✓			
invalidLoginReport	✓	✓	✓	✓	✓	✓	✓	✓
listBackups	✓	✓	✓	✓	✓	✓	✓	✓
listFiles	✓	✓	✓	✓	✓	✓	✓	✓
loadData				✓				
loadDimData				✓				
loadViewpoint							✓	
login	✓	✓	✓	✓	✓	✓	✓	✓
logout	✓	✓	✓	✓	✓	✓	✓	✓
maskData	✓	✓	✓		✓			
mergeDataSlices	✓				✓			
mergeSlices				✓				
optimizeASOCube				✓				
programDocumentationReport				✓				
provisionReport	✓	✓	✓	✓	✓	✓	✓	✓
purgeArchivedTmTransactions						✓		
purgeTmTransactions						✓		
recomputeOwnershipData		✓	✓					
recreate	✓	✓	✓	✓	✓	✓	✓	✓
refreshCube	✓	✓	✓		✓			
removeUserFromGroups	✓	✓	✓	✓	✓	✓	✓	✓
removeUsers	✓	✓	✓	✓	✓	✓	✓	✓
removeUsersFromGroup	✓	✓	✓	✓	✓	✓	✓	✓
removeUsersFromTeam		✓	✓			✓		
renameSnapshot	✓	✓	✓	✓	✓	✓	✓	
replay	✓	✓	✓	✓	✓	✓	✓	✓
resetService	✓	✓	✓	✓	✓	✓	✓	✓
restoreBackup	✓	✓	✓	✓	✓	✓	✓	✓
restructureCube	✓	✓	✓					
roleAssignmentAuditReport	✓	✓	✓	✓	✓	✓	✓	✓
roleAssignmentReport	✓	✓	✓	✓	✓	✓	✓	✓
runAutomatch						✓		
runBatch	✓	✓	✓	✓	✓	✓		
runBusinessRule	✓	✓	✓					
runCalc				✓				

Tabelle 2-1 (Fortsetzung) Alle EPM Automate-Befehle

Befehlsname	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
runComplianceReport						✓		
runDailyMaintenance	✓	✓	✓	✓	✓	✓	✓	✓
runDataRule	✓	✓	✓	✓	✓	✓		
runDMReport	✓	✓	✓	✓	✓	✓		
runIntegration	✓	✓	✓	✓	✓	✓		
runMatchingReport						✓		
runPlanTypeMap	✓							
runRuleSet	✓	✓	✓					
runSupplementalDataReport		✓	✓					
runTaskManagerReport		✓	✓					
sendMail	✓	✓	✓	✓	✓	✓	✓	✓
setApplicationAdminMode	✓	✓	✓		✓	✓		
setDailyMaintenanceStartTime	✓	✓	✓	✓	✓	✓	✓	✓
setDemoDates		✓	✓			✓		
setEJJournalStatus		✓						
setEncryptionKey	✓	✓	✓	✓	✓	✓	✓	✓
setEssbaseQryGovExecTime	✓	✓	✓	✓	✓			
setIdleSessionTimeout	✓	✓	✓	✓	✓	✓	✓	✓
setIPAllowlist	✓	✓	✓	✓	✓	✓	✓	✓
setManualDataAccess	✓	✓	✓	✓	✓	✓	✓	✓
setPeriodStatus						✓		
setSubstVars	✓	✓	✓		✓			
setVirusScanOnFileUploads	✓	✓	✓	✓	✓	✓	✓	✓
simulateConcurrentUsage	✓	✓	✓					
skipUpdate	✓	✓	✓	✓	✓	✓	✓	✓
snapshotCompareReport	✓	✓	✓		✓			
sortMember	✓				✓			
unassignRole	✓	✓	✓	✓	✓	✓	✓	✓
updateUsers	✓	✓	✓	✓	✓	✓	✓	✓
upgrade	✓	✓	✓	✓	✓	✓	✓	✓
uploadFile	✓	✓	✓	✓	✓	✓	✓	✓
userAuditReport	✓	✓	✓	✓	✓	✓	✓	✓
userGroupReport	✓	✓	✓	✓	✓	✓	✓	✓
validateConsolidationMetadata		✓						
validateModel					✓			

- * Dieser Befehl wird nur unterstützt, wenn hybridfähige Oracle Essbase-Cubes in der Anwendung aktiviert sind. Strategische Personalplanung und Sales Planning unterstützen kein hybridfähiges Essbase. Dieser Befehl wird nicht für FreeForm unterstützt.

- ** Dieser Befehl wird nicht für FreeForm unterstützt.

Abkürzungen

- PLN: Planning (einschließlich Planning Modules)
- FF: FreeForm
- SWP: Strategische Personalplanung
- SP: Sales Planning
- FCC: Financial Consolidation and Close
- TR: Tax Reporting
- PCM: Profitability and Cost Management
- EPCM: Enterprise Profitability and Cost Management
- AR: Account Reconciliation
- EDM: Oracle Enterprise Data Management Cloud
- NR: Narrative Reporting

EPM Automate-Befehle

In diesem Abschnitt werden alle EPM Automate-Befehle erläutert. Zu dem jeweiligen Befehl sind die folgenden Informationen verfügbar: die Services, die den Befehl verwenden können, die Verwendung des Befehls und ein Beispiel.

addUsers

Erstellt einen Batch von Benutzern in einer Identitätsdomain mit einer ANSI- oder UTF-8-kodierten CSV-Datei (Comma Separated Value), die in die Umgebung hochgeladen wurde. Informiert neue Benutzer zudem über ihren Benutzernamen und ihr temporäres Kennwort.

Verwenden Sie den Befehl [uploadFile](#), um Dateien in eine Umgebung hochzuladen. Alle Spalten in der CSV-Datei sind obligatorisch. Dieser Befehl validiert die Werte in jeder Spalte einer Definition und zeigt Fehlermeldungen an, die jeden fehlenden oder ungültigen Wert angeben. Die CSV-Datei weist das folgende Format auf:

```
First Name,Last Name,Email,User Login  
Jane,Doe,jane.doe@example.com,jdoe  
John,Doe, john.doe@example.com, john.doe@example.com
```

Eine ausführliche Beschreibung des CSV-Dateiformats finden Sie unter Batch mit Benutzeraccounts importieren in der Dokumentation *Erste Schritte mit Oracle Cloud*.

Beim Wert von `User Login`, der in der Importdatei angegeben wird, wird nicht zwischen Groß- und Kleinschreibung unterschieden. Beispiel: Der Wert `John.doe@example.com` wird genauso wie `John.Doe@example.com` oder jede andere Variation der Groß-/Kleinschreibung behandelt.

Wenn eine Benutzerdefinition in der CSV-Datei mit einem Benutzeraccount in der Identitätsdomain übereinstimmt, werden keine Änderungen am vorhandenen Benutzeraccount vorgenommen. Dieser Befehl erstellt Accounts nur für neue Benutzer, deren Accountinformationen in der Datei enthalten sind. Da

Benutzeraccounts in allen Umgebungen vorkommen, die eine Identitätsdomain unterstützt, stehen neue Benutzer in allen Umgebungen zur Verfügung, die dieselbe Identitätsdomain verwenden.

Wenn die Befehlsausführung abgeschlossen ist, gibt EPM Automate Informationen zu jedem nicht erfolgreichen Eintrag an die Konsole aus. Lesen Sie diese Informationen, um zu verstehen, warum die Befehlsausführung für einige Einträge in der CSV-Datei nicht erfolgreich war.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Identitätsdomainadministrator und alle vordefinierten Rollen (Serviceadministrator, Poweruser, Benutzer oder Leseberechtigter)

Verwendung

```
epmautomate addUsers FILE_NAME [userPassword=PASSWORD] [resetPassword=true|false], wobei Folgendes gilt:
```

- *FILE_NAME* ist der Name einer CSV-Datei mit Benutzerinformationen. Die Eingabedatei mit Multibyte-Zeichen muss die UTF-8-Zeichenkodierung verwenden. Die Verwendung der ANSI-Codierung verursacht Probleme bei der Anzeige von Benutzerinformationen in den Fenstern "Meine Services".
- *userPassword* (optional) gibt das Standardkennwort für alle neuen Benutzer an, die in der Identitätsdomain erstellt werden. Wenn dieses Kennwort angegeben wird, muss es die Mindestanforderungen an Kennwörter der Identitätsdomain erfüllen. Wenn der Parameter nicht angegeben wird, wird jedem Benutzer ein eindeutiges temporäres Kennwort zugewiesen.
Wenn der Parameter angegeben wird, wird der Wert von *userPassword* für alle in der CSV-Datei angegebenen Benutzer als Kennwort verwendet. Es kann von Vorteil sein, dasselbe Kennwort allen Benutzern zuzuweisen, wenn Sie Benutzer ausschließlich zu Testzwecken erstellen. Wenn Sie reelle Oracle Enterprise Performance Management Cloud-Benutzer erstellen und jedem Benutzer ein spezifisches Kennwort zuweisen möchten, können Sie diesen Befehl verwenden. Geben Sie dabei keinen Wert für den optionalen Parameter *userPassword* an.
- *resetPassword* (optional) gibt an, ob der neue Benutzer das Kennwort bei der ersten Anmeldung ändern muss. Der Standardwert ist `true`. Sofern dieser Parameter nicht auf `false` gesetzt ist, wird erzwungen, dass neue Benutzer das Kennwort bei der ersten Anmeldung ändern.
Dieser Befehl sendet jedem neuen Benutzer eine E-Mail mit Details zu den zugehörigen Accounts (Benutzername und Kennwort), wenn *resetPassword* auf `true` gesetzt ist. Wenn *resetPassword* auf `false` gesetzt ist, wird keine E-Mail gesendet. Wenn Sie *resetPassword* auf "false" setzen, müssen Sie *userPassword* angeben. Anderenfalls wird jedem Benutzer ein eindeutiges temporäres Kennwort zugewiesen. Da jedoch keine E-Mail gesendet wird, kennen die Benutzer die Kennwörter nicht und können sich nicht anmelden.

Beispiele

- So fügen Sie Testbenutzer in der Identitätsdomain mit demselben Kennwort hinzu und erzwingen nicht, dass sie ihr Kennwort ändern:

```
epmautomate addUsers user_file.CSV userPassword=Example@PwD12  
resetPassword=false
```
- So fügen Sie Benutzer mit einem temporären Kennwort zur Identitätsdomain hinzu und fordern sie auf, dieses zu ändern:

```
epmautomate addUsers user_file.CSV
```

addUsersToGroup

Fügt einer vorhandenen Gruppe in der Zugriffskontrolle einen Batch von Benutzern hinzu. Hierzu wird eine ANSI- oder UTF-8-codierte CSV-Datei verwendet, die in die Umgebung hochgeladen wurde.

Verwenden Sie den Befehl [uploadFile](#), um Dateien in eine Umgebung hochzuladen. Beim Benutzeranmeldewert wird nicht zwischen Groß- und Kleinschreibung unterschieden. Das Dateiformat lautet wie folgt:

```
User Login  
jdoe  
john.doe@example.com
```



Hinweis:

Ein Benutzer wird einer Gruppe nur hinzugefügt, wenn die beiden folgenden Bedingungen erfüllt sind:

- Der in der Datei enthaltenen Benutzeranmeldewert ist in der Identitätsdomain für die Umgebung vorhanden. Bei Benutzeranmeldewerten wird nicht zwischen Groß- und Kleinschreibung unterschieden.
- Der Benutzer ist einer vordefinierten Rolle in der Identitätsdomain zugewiesen.

Wenn die Befehlsausführung abgeschlossen ist, gibt EPM Automate Informationen zu jedem nicht erfolgreichen Eintrag an die Konsole aus. Lesen Sie diese Informationen, um zu verstehen, warum die Befehlsausführung für einige Einträge in der CSV-Datei nicht erfolgreich war.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Enterprise Profitability and Cost Management, Narrative Reporting, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator oder Zugriffskontrollemanager

Verwendung

epmautomate addUsersToGroup *FILE_NAME* *GROUP_NAME* wobei:

- *FILE_NAME* der Name einer CSV-Datei mit den Anmeldenamen der Benutzer ist, die Sie einer Gruppe in der Zugriffskontrolle zuweisen möchten.
- *GROUP_NAME* der Name einer in der Zugriffskontrolle vorhandenen Gruppe ist. Bei diesem Wert wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Beispiel

```
epmautomate addUsersToGroup user_file.CSV example_group
```

addUsersToTeam

Fügt die in einer CSV-Datei enthaltenen Benutzer von Oracle Enterprise Performance Management Cloud einem vorhandenen Team hinzu.

Wenn ein Benutzer in der CSV-Datei bereits Mitglied des Teams ist, ignoriert dieser Befehl den entsprechenden Benutzer. Bei den Werten in dieser Datei wird nicht zwischen Groß- und Kleinschreibung unterschieden. Die CSV-Datei weist das folgende Format auf:

```
User Login, primary_user  
jdoe, yes  
jane.doe@example.com, no
```

Hinweis:

Ein primärer Benutzer ist standardmäßig zum Ausführen der Aufgaben vorgesehen, die dem Team zugewiesen sind.

Gilt für

Financial Consolidation and Close, Tax Reporting und Account Reconciliation.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

epmautomate addUsersToTeam *FILE* *TEAM_NAME* wobei Folgendes gilt:

- *FILE* gibt eine CSV-Datei im UTF8-Format mit den Anmelde-IDs der Benutzer an, die dem Team hinzugefügt werden sollen. Bevor Sie diesen Befehl ausführen, müssen Sie mit dem Befehl [uploadFile](#) Dateien in eine Umgebung hochladen.
- *TEAM_NAME* gibt einen in der Zugriffskontrolle definierten Teamnamen an. Bei diesem Wert wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Beispiel

```
epmautomate addUsersToTeam example_users.csv example_team
```

addUserToGroups

Fügt einen Benutzer als Mitglied der Zugriffskontrollgruppen hinzu, die in einer mit ANSI oder UTF-8 codierten CSV-Datei identifiziert sind.

Verwenden Sie den Befehl [uploadFile](#), um Dateien in eine Umgebung hochzuladen. Das Dateiformat lautet wie folgt:

```
Group Name  
Group1  
Group2
```

Bei Gruppennamenswerten wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Wenn die Befehlsausführung abgeschlossen ist, gibt EPM Automate Informationen zu jedem nicht erfolgreichen Eintrag an die Konsole aus. Lesen Sie diese Informationen, um zu verstehen, warum die Befehlsausführung für einige Einträge in der CSV-Datei nicht erfolgreich war.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator oder Zugriffskontrollemanager

Verwendung

`epmautomate addUserToGroups FILE_NAME User_Login`, wobei Folgendes gilt:

- `FILE_NAME` ist der Name einer CSV-Datei, die die Zugriffskontrollgruppennamen enthält, denen Sie den Benutzer zuweisen möchten.
- `User_Login` ist die Anmelde-ID eines Oracle Enterprise Performance Management Cloud-Benutzers, der Gruppen in der Zugriffskontrolle zugewiesen werden soll. Diese Benutzeranmelde-ID, bei der die Groß-/Kleinschreibung nicht beachtet wird, muss in der Identitätsdomain für die Umgebung vorhanden sein und einer vordefinierten Rolle zugewiesen werden.

Beispiel

```
epmautomate addUserToGroups groups.csv jdoe@example.com
```

applicationAdminMode

Versetzt die Anwendung in den Administratormodus, sodass der Zugriff auf die Anwendung auf Serviceadministratoren beschränkt ist.

Dieser Befehl ist nützlich, um zu verhindern, dass Benutzer an der Anwendung arbeiten, wenn Serviceadministratoren administrative Vorgänge durchführen. Die Anwendung bleibt im Administrationsmodus, bis Sie sie wieder zurücksetzen, damit alle Benutzer darauf zugreifen können.



Hinweis:

Dieser Befehl ist veraltet, wurde aber nicht aus EPM Automate entfernt. Oracle empfiehlt, stattdessen den Befehl [setApplicationAdminMode](#) zu verwenden.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate applicationAdminMode VALUE`, wobei `VALUE` angibt, ob die Anwendung in den Administrationsmodus versetzt werden soll. Folgende Werte werden akzeptiert:

- `true`, um die Anwendung in den Administrationsmodus zu versetzen.
- `false`, um die Anwendung in den Normalmodus zurückzusetzen, damit alle Benutzer darauf zugreifen können.

Beispiele

- Anwendung in Administrationsmodus setzen: `epmautomate applicationAdminMode true`
- Anwendung in Normalmodus zurücksetzen: `epmautomate applicationAdminMode false`

applyDataGrants

Aktualisiert die Datenzugriffsberechtigungen, mit denen der Zugriff auf Oracle Essbase-Datensegmente gesteuert wird, sodass sie mit den Datenzugriffsberechtigungen übereinstimmen, die in einer Profitability and Cost Management-Anwendung definiert sind.

Datenzugriffsberechtigungen auf Benutzer- und Gruppenebene, die Sie in der Profitability and Cost Management-Anwendung gewähren, werden automatisch in Essbase synchronisiert. Verwenden Sie diesen Befehl, um den Zugriff auf Essbase-Daten zu synchronisieren, wenn Sie eine Abweichung zwischen den Datenzugriffsberechtigungen in der Anwendung und den Filtern in Essbase vermuten.

Die Zeit, die für die Ausführung dieses Vorgangs benötigt wird, hängt von der Größe der Anwendung ab. Stellen Sie sicher, dass der Aktualisierungsvorgang der

Datenzugriffsberechtigung abgeschlossen ist, bevor die Anwendung während des nächsten Wartungsfensters gesichert wird. Da während der Ausführung dieses Vorgangs die Anwendung nicht verwendet werden darf, empfiehlt Oracle, diesen Vorgang für einen Zeitpunkt zu planen, zu dem Benutzer die Anwendung nicht verwenden.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate applyDataGrants APPLICATION_NAME wobei APPLICATION_NAME` der Name der Profitability and Cost Management-Anwendung ist, für die Datenzugriffsberechtigungen neu erstellt werden sollen.

Beispiel

```
epmautomate applyDataGrants BksML12
```

archiveTmTransactions

Archiviert abgeglichene Transaktionen, einschließlich der Bestätigungs- und Anpassungsdetails, die mindestens dem angegebenen Alter entsprechen. Die abgeglichenen Transaktionen werden in einer ZIP-Datei erfasst.

Verwenden Sie diesen Befehl, um die Anwendung von Account Reconciliation in einer optimalen Größe zu halten, indem Sie alte abgeglichene Transaktionen archivieren und dann auf Basis der Policies Ihres Unternehmens zur Aufbewahrung von Transaktionen löschen.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

```
epmautomate archiveTmTransactions matchType age [filterOperator=VALUE]  
[filterValue=VALUE] [logFilename=FILE_NAME] [filename=FILE_NAME], wobei  
Folgendes gilt:
```

- `matchType` ist die ID (TextID) des Abgleichstyps, aus dem die abgeglichenen Transaktionen archiviert werden müssen.
- `age` gibt die Anzahl Tage seit dem letzten Abgleich der Transaktion an. Abgeglichene Transaktionen, die diesem Wert entsprechen oder älter sind, werden archiviert.

- `filterOperator` (optional) stellt eine der folgenden Filterbedingungen zum Identifizieren der Konten dar, die abgeglichene Transaktionen für die Archivierung enthalten. Dieser Wert wird mit `filterValue` kombiniert, um die Konten zu ermitteln, aus denen abgeglichene Transaktionen archiviert werden müssen:
 - `equals`
 - `not_equals`
 - `starts_with`
 - `ends_with`
 - `contains`
 - `not_contains`
- `filterValue` (optional) ist ein Filterwert zum Ermitteln der Transaktionen, die archiviert werden sollen. Wenn `filterOperator` den Wert `equals` oder `not_equals` aufweist, können Sie eine durch Leerzeichen getrennte Liste verwenden, um mehrere Werte anzugeben. Beispiel: `filterValue=101-120 filterValue=102-202`. Wenn mehrere Werte angegeben werden, werden Transaktionen aus Konten zum Archivieren ausgewählt, die zu einer beliebigen Kombination aus Filteroperator und Filterwert passen.

 **Note:**

Wenn `filterOperator` und `filterValue` nicht angegeben sind, werden alle abgeglichenen Transaktionen, die dem Wert `age` entsprechen oder älter sind, aus allen Konten für den angegebenen `matchType` archiviert.

- `logFilename` (optional) ist der Name einer Logdatei zum Aufzeichnen von Informationen über die Befehlsaktivität. Wenn kein Dateiname angegeben ist, wird automatisch eine Logdatei namens `Archive_Transactions_matchType_JOBID.log` generiert.
- `filename` (optional) ist der Name einer ZIP-Datei, die die archivierten Transaktionen enthalten soll. Wenn kein Name angegeben ist, erstellt der Befehl standardmäßig `Archived_Transactions_matchType_JOBID.zip`. Verwenden Sie den Befehl [downloadFile](#), um diese Datei auf einen lokalen Computer herunterzuladen.

 **Note:**

Dieser Befehl führt den Job zum Archivieren von TM-Transaktionen aus, wobei die Parameter verwendet werden, die Sie festlegen. Die Job-ID wird in der Befehlsausgabe zurückgegeben, damit sie leichter mit dem Befehl [purgeArchivedTmTransactions](#) verwendet werden kann. Sie können den Job von der Jobkonsole aus überwachen.

Beispiele

- Archivieren Sie alte abgeglichene Transaktionen ohne Verwendung von Filtern, sondern mit benutzerdefinierter Logdatei und ZIP-Dateinamen:


```
epmautomate archiveTmTransactions cashrecon 180 logFile=tmlogs.log
filename=trans.zip
```

- Archivieren Sie alte abgeglichene Transaktionen mit Filtern:
 - `epmautomate archiveTmTransactions cashrecon 180
filterOperator>equals filterValue=101-120 FilterValue=102-202`
 - `epmautomate archiveTmTransactions cashrecon 180
filterOperator=contains filterValue=11`

assignRole

Weist Benutzern (einschließlich dem Benutzer, der diesen Befehl ausführt), deren Anmelde-IDs in einer ANSI- oder UTF-8-codierten CSV-Datei enthalten sind, eine Rolle zu. Mit diesem Befehl weisen Sie Benutzer einer vordefinierten Rolle oder einer Anwendungsrolle zu, die zu einer Anwendung von Planning, Financial Consolidation and Close, Tax Reporting, Account Reconciliation oder Oracle Enterprise Data Management Cloud gehört.

Hinweis:

Sie können diesen Befehl nicht verwenden, um Anwendungsrollen für Profitability and Cost Management- und Narrative Reporting-Anwendungen zuzuweisen, da diese Anwendungen die Zuweisung von Anwendungsrollen zu Benutzern nicht unterstützen.

Bevor Sie diesen Befehl verwenden, müssen Sie mit dem Befehl [uploadFile](#) Dateien in eine Umgebung hochladen. Das Dateiformat lautet wie folgt:

```
User Login  
jane.doe@example.com  
jdoe
```

Informationen hierzu finden Sie unter *Vielen Benutzern eine Rolle zuweisen in Erste Schritte mit Oracle Cloud*.

Hinweis:

- Bei Benutzeranmeldewerten in der Datei wird nicht zwischen Groß- und Kleinschreibung unterschieden.
- Setzen Sie Rollennamen, die Leerzeichen enthalten, in doppelte Anführungszeichen.

Wenn die Befehlsausführung abgeschlossen ist, gibt EPM Automate Informationen zu jedem nicht erfolgreichen Eintrag an die Konsole aus. Lesen Sie diese Informationen, um zu verstehen, warum die Befehlsausführung für einige Einträge in der CSV-Datei nicht erfolgreich war.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise

Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

So weisen Sie vordefinierte Rollen zu:

- Classic-Umgebungen: Identitätsdomainadministrator und alle vordefinierten Rollen (Serviceadministrator, Poweruser, Benutzer oder Leseberechtigter)
- OCI-Umgebungen: Serviceadministrator oder Identitätsdomainadministrator und alle vordefinierten Rollen (Serviceadministrator, Poweruser, Benutzer oder Leseberechtigter)

So weisen Sie Anwendungsrollen zu: Serviceadministrator oder Zugriffskontrollemanager

Verwendung

`epmautomate assignRole FILE_NAME ROLE` wobei:

- `FILE_NAME` der Name einer CSV-Datei mit Benutzeranmelde-IDs ist. Geben Sie die Erweiterung "CSV" in Kleinbuchstaben an.
- `ROLE` enthält einen der folgenden Werte. Bei diesem Wert wird nicht zwischen Groß- und Kleinschreibung unterschieden:
 - Wenn Sie Benutzer vordefinierten Rollen für die Identitätsdomain zuweisen, muss `ROLE` eine vordefinierte Rolle angeben, die für den Service gültig ist. Informationen hierzu finden Sie unter Erläuterungen zu vordefinierten Rollen in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*. Eine Beschreibung dieser Rollen finden Sie unter Rollenzuweisungen auf Anwendungsebene verwalten in der Dokumentation *Zugriffskontrolle für Oracle Enterprise Performance Management Cloud verwalten*
 - Wenn Sie Anwendungsrollen Benutzer zuweisen, muss `ROLE` eine Rolle angeben, die zur Anwendung in der aktuellen Umgebung gehört. Anwendungsrollen sind auf der Registerkarte **Rollen** unter Zugriffskontrolle aufgelistet. Beschreibungen der Anwendungsrollen für jeden Geschäftsprozess finden Sie in diesen Themen in der Dokumentation *Zugriffskontrolle für Oracle Enterprise Performance Management Cloud verwalten*:
 - * Planning, Financial Consolidation and Close und Tax Reporting
 - * Enterprise Profitability and Cost Management
 - * Oracle Enterprise Data Management
 - * Profitability and Cost Management
 - * Account Reconciliation

Beispiele

- So weisen Sie Benutzer einer vordefinierten Rolle für eine Identitätsdomain zu:
`epmautomate assignRole admin_role_file.csv "Service Administrator"`
- So weisen Sie Benutzer einer Anwendungsrolle zu:
`epmautomate assignRole example_file.csv "Task List Access Manager"`

autoPredict

Erstellt Vorhersagen zur zukünftigen Performance auf der Basis einer vorhandenen Definition für die automatische Vorhersage in Planning oder Planning Modules.

Mit diesem Befehl wird ein Job eingeleitet, der die historischen Daten für jedes Element verwendet, das in der in der Anwendung angegebenen Definition für die automatische Vorhersage angegeben ist. Ausführliche Informationen zu den Anwendungen, die die Funktion der automatischen Vorhersage verwenden, sowie zum Einrichten von Vorhersagen finden Sie unter Einrichten von Vorhersagen zur automatischen Ausführung mit automatischer Vorhersage in der Dokumentation *Planning verwalten*.

Gilt für

Planning, Planning Modules, wenn Oracle Essbase-Hybrid-Cubes in der Anwendung aktiviert sind.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate autoPredict PREDICTION_DEFINITION [forceRun=true|false]
[paginatedDim=DIMENSION_NAME] wobei Folgendes gilt:
```

- *PREDICTION_DEFINITION* ist der Name einer in der Anwendung verfügbaren Definition für die automatische Vorhersage.
- *forceRun* (optional) gibt an, ob die Vorhersage ausgeführt werden soll, wenn die zugrundeliegende Definition seit der ersten Ausführung nicht geändert wurde. Der Standardwert ist *false*.
Legen Sie den Wert dieses Parameters auf *true* fest, um den Job der automatischen Vorhersage auszuführen, auch wenn die Jobdefinition nicht geändert wurde. Verwenden Sie den Standardwert (*false*), um die Vorhersage nur einmal bei der ersten Ausführung des Jobs auszuführen.
- *paginatedDim* (optional) gibt eine Dimension an, die verwendet wird, um den Job der automatischen Vorhersage zu beschleunigen, indem Vorhersagen in getrennten Threads parallel ausgeführt werden. Damit diese parallelen Threads effizient ausgeführt werden, geben Sie eine Dimension an, die zu einer gleichmäßigen Verteilung der Daten auf die einzelnen Vorhersage-Threads führt.

Beispiel

```
epmautomate autoPredict AS0toBS0 forceRun=true paginatedDim=Entity
```

calculateModel

Führt den Berechnungsprozess in Enterprise Profitability and Cost Management-Anwendungen aus.

Gilt für

Enterprise Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate calculateModel POV_NAME MODEL_NAME EXECUTION_TYPE
[povDelimiter=DELIMITER] [optimizeForReporting=true|false]
[captureDebugScripts=true|false] [comment=COMMENT] [PARAMETER=VALUE]. Dabei gilt:
```

- *POV_NAME* ist der Name des zu berechnenden Daten-POV. Um mehrere POVs zu berechnen, listen Sie POV-Namen durch Komma als Trennzeichen getrennt auf. Verwenden Sie keine anderen Trennzeichen zum Trennen von POV-Namen. Setzen Sie die Liste der POV-Namen in doppelte Anführungszeichen, wenn Leerzeichen in Elementnamen vorkommen.
- *MODEL_NAME* ist der Name des zu berechnenden Modells. Setzen Sie den Modellnamen in doppelte Anführungszeichen, wenn im Namen Leerzeichen vorkommen.
- *EXECUTION_TYPE* ist einer der folgenden Werte zur Angabe des Regelausführungstyps.
 - *ALL_RULES* gibt an, dass alle Regeln zum Berechnen des POV verwendet werden. Wenn Sie diesen Wert angeben, dürfen Sie keine Laufzeitparameter für Regelteilmengen oder Einzelregeln angeben, wie *rulesetSeqNumStart*, *rulesetSeqNumEnd* oder *ruleName*.
 - *RULESET_SUBSET* gibt an, dass der POV mit einer Teilmenge eines Regelsets berechnet wird. Wenn Sie diesen Wert verwenden, müssen Sie Werte für *rulesetSeqNumStart* und *rulesetSeqNumEnd* als Laufzeitparameter angeben.
 - *SINGLE_RULE* gibt an, dass der POV mit einer bestimmten Regel berechnet wird. Wenn Sie diesen Wert verwenden, dürfen Sie nur *ruleName* als Laufzeitparameter angeben.
 - *RUN_FROM_RULE* gibt an, dass Berechnungen für einen POV beginnend mit einer bestimmten Regel ausgeführt werden. Wenn Sie diesen Wert verwenden, dürfen Sie nur *ruleName* als Laufzeitparameter angeben.
 - *STOP_AFTER_RULE* gibt an, dass die POV-Berechnung gestoppt werden soll, wenn eine bestimmte Regel abgeschlossen ist. Wenn Sie diesen Wert verwenden, dürfen Sie nur *ruleName* als Laufzeitparameter angeben.
- *povDelimiter* ist das optionale Trennzeichen in POV-Werten. Das Standardtrennzeichen ist ein "_" (Unterstrich). Trennzeichen müssen in doppelte Anführungszeichen gesetzt werden. Es werden nur die folgenden Trennzeichen unterstützt:
 - _ (Unterstrich)
 - # (Rautezeichen)
 - & (Et-Zeichen)
 - ~ (Tilde)
 - % (Prozentsatz)
 - ; (Semikolon)
 - : (Doppelpunkt)

- - (Strich)
- `optimizeForReporting=true|false` gibt optional an, ob Berechnungen mit oder ohne Optimierung für das Reporting ausgeführt werden sollen. Der Standardwert ist `false`.
Setzen Sie diesen Wert auf `false`, um Verarbeitungszeit zu sparen, indem der Schritt zur Aggregationserstellung übersprungen wird, z.B. wenn Sie eine einzelne Regel oder eine sequentielle Reihe von POVs ausführen. Wenn Sie mehrere nebenläufige Berechnungsjobs ausführen, legen Sie `optimizeForReporting=true` für alle Jobs fest. So führt nur der letzte Job, der abgeschlossen wird, die Aggregation durch, und Sie vermeiden redundante Verarbeitung und verhindern, dass sich laufende Jobs verlangsamen.
- `captureDebugScripts=true|false` gibt optional an, ob Debug-Skripte in der Inbox generiert werden sollen. Diese Skripte müssen möglicherweise zur Behebung von Berechnungsfehlern an Oracle weitergegeben werden. Der Standardwert ist `false`.
- `comment="COMMENT"` gibt optional einen Kommentar zum Prozess in doppelten Anführungszeichen an.
- `PARAMETER=VALUE` gibt optional Laufzeitparameter und deren Werte zum Ausführen der Berechnung an. Geben Sie so viele Parameter/Wert-Paare an, wie der Prozess erfordert. Gültige Parameter und deren Werte:
 - `rulesetSeqNumStart`: Die Folgenummer der ersten auszuführenden Regel im Regelset. Nur gültig, wenn `EXECUTION_TYPE=RULESET_SUBSET` verwendet wird.
 - `rulesetSeqNumEnd` gibt die Folgenummer der letzten auszuführenden Regel im Regelset an. Nur gültig, wenn `EXECUTION_TYPE=RULESET_SUBSET` verwendet wird.
 - `ruleName`: Name der auszuführenden Regel. Setzen Sie den Wert in doppelte Anführungszeichen, wenn Leerzeichen darin vorkommen. Nur gültig, wenn der Wert von `EXECUTION_TYPE` auf `SINGLE_RULE`, `RUN_FROM_RULE` oder `STOP_AFTER_RULE` gesetzt ist.
 - `clearCalculatedData=true|false` gibt an, ob vorhandene Berechnungen gelöscht werden sollen. Der Standardwert ist `false`.
 - `executeCalculations=true|false` gibt an, ob Berechnungen ausgeführt werden sollen. Der Standardwert ist `false`.

 **Note:**

Parameterwerte (`true` und `false`) müssen in Kleinbuchstaben angegeben werden.

Beispiele

- Alle Regeln ausführen, um einen einzelnen POV zu berechnen:
`epmautomate calculateModel FY21_Jan_Actual_Working ForecastingModel ALL_RULES clearCalculatedData=true executeCalculations=true optimizeForReporting=true comment="Running all rules to calculate a POV"`
- Alle Regeln ausführen, um mehrere POVs zu berechnen:

```
epmautomate calculateModel
"FY21:Jan:Actual:Working,FY21:Feb:Actual:Working,FY21:Mar:Actual:Working"
"10 Actuals Allocation Process" ALL_RULES clearCalculatedData=true
executeCalculations=true optimizeForReporting=true captureDebugScripts=true
comment="Test calculation of many POVs" povDelimiter=":"
```

- Teilmenge eines Regelsets ausführen, um den POV zu berechnen:

```
epmautomate calculateModel FY21_Jan_Actual_Working ForecastingModel
RULESET_SUBSET rulesetSeqNumStart=10 rulesetSeqNumEnd=20
clearCalculatedData=true executeCalculations=true comment="Running a subset
of rules to calculate a POV"
```

- Bestimmte Regel ausführen, um den POV zu berechnen:

```
epmautomate calculateModel FY21_Jan_Actual_Working ForecastingModel
SINGLE_RULE ruleName="Occupancy Expense Allocations"
clearCalculatedData=true executeCalculations=true comment="Running a
specific rule to calculate a POV"
```

- Alle Regeln ausführen, um einen einzelnen POV mit einem benutzerdefinierten POV-Trennzeichen zu berechnen:

```
epmautomate calculateModel FY21:Jan:Actual_Working ForecastingModel
ALL_RULES clearCalculatedData=true executeCalculations=true
optimizeForReporting=true comment="Running all rules to calculate a POV"
povDelimiter=":"
```

- Alle Regeln ausführen, um POVs und Modelle mit einem Leerzeichen im Namen zu berechnen:

```
epmautomate calculateModel "FY21_Jan_New
Actual_Working,FY21:Feb:Actual:Working" "Forecasting Model" ALL_RULES
clearCalculatedData=true executeCalculations=true optimizeForReporting=true
comment="Running all rules to calculate a POV"
```

clearCube

Löscht bestimmte Daten aus Eingabe- und Reporting-Cubes anhand der in einem Job vom Typ `clear cube` angegebenen Einstellungen.

Dieser Befehl löscht nicht die Anwendungsdefinition in den relationalen Tabellen der Anwendung. Informationen hierzu finden Sie unter *Cubes löschen* in der Dokumentation *Planning verwalten*.

Gilt für

Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate clearCube JOB_NAME` wobei `JOB_NAME` der Name eines in der Anwendung definierten Jobs ist.

Beispiel

```
epmautomate clearCube ClearPlan1
```

clearDataByPointOfView

Löscht Daten für einen bestimmten POV für einen Enterprise Profitability and Cost Management-Cube.

Gilt für

Enterprise Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmAutomate clearDataByPointOfView POV_NAME [cubeName=CUBE_NAME]  
[PARAMETER=VALUE]. Dabei gilt:
```

- *POV_NAME* ist der Name eines POV in der Anwendung.
- *cubeName* ist optional der Name des Cubes, in dem Daten gelöscht werden sollen. Der Standardwert lautet *PCM_CLC*.
- *PARAMETER=VALUE* gibt optionale Laufzeitparameter und deren Werte an. Geben Sie so viele Parameter- und Wertpaarungen an, wie der Prozess erfordert. Gültige Parameter und deren Werte:
 - *povDelimiter* gibt das in POV-Werten verwendete Trennzeichen an. Der Standardwert lautet *::* (doppelter Doppelpunkt). Dieser Wert muss in doppelten Anführungszeichen angegeben werden. Beispiel:
povDelimiter="_".
Neben dem Standardtrennzeichen werden nur die folgenden Trennzeichen unterstützt: *_* (Unterstrich), *#* (Rautezeichen), *&* (Et-Zeichen), *~* (Tilde), *%* (Prozentsatz), *;* (Semikolon), *:* (Doppelpunkt), *-* (Strich).
 - *clearInput=true|false* gibt an, ob Eingabedaten gelöscht werden sollen. Der Standardwert ist *false*.
 - *clearAllocatedValues=true|false* gibt an, ob umgelegte Werte gelöscht werden sollen. Der Standardwert ist *false*.
 - *clearAdjustmentValues=true|false* gibt an, ob Anpassungswerte gelöscht werden sollen. Der Standardwert ist *false*.

 **Note:**

- * Parameterwerte (`true` oder `false`) müssen in Kleinbuchstaben angegeben werden.
- * Sie müssen mindestens einen der Parameter `clearInput`, `clearAllocatedValues` und `clearAdjustmentValues` auf `true` setzen.

Beispiele

- Daten aus einem POV im PCM_CLC-Standard-Cube mit dem standardmäßigen POV-Trennzeichen löschen:

```
epmAutomate clearDataByPointOfView FY21::Jan::Actual::Working
clearInput=true clearAllocatedValues=true clearAdjustmentValues=true
```
- Eingabedaten und umgelegte Werte aus einem POV in einem bestimmten Cube mit einem benutzerdefinierten POV-Trennzeichen löschen:

```
epmAutomate clearDataByPointOfView FY21_Jan_Actual_Working cubeName=PCM_REP
povDelimiter="_" clearInput=true clearAllocatedValues=true
```
- Eingabedaten aus einem POV in einem bestimmten Cube mit einem benutzerdefinierten POV-Trennzeichen löschen:

```
epmAutomate clearDataByPointOfView FY21:Jan:Actual:Working cubeName=PCM_REP
povDelimiter=":" clearInput=true
```

clearDataByProfile

Löscht Daten aus den Elementen (z.B. Regionen), die in einem Profil zum Löschen von Daten identifiziert werden, das in Financial Consolidation and Close und Tax Reporting definiert ist.

Gilt für

Financial Consolidation and Close, Tax Reporting

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate clearDataByProfile PROFILE_NAME wobei PROFILE_NAME` der Name des Profils zum Löschen von Daten ist.

Beispiel

```
epmautomate clearDataByProfile clearDataProfile_01
```

clearPOV

Löscht Modellartefakte und -daten aus einer POV-Kombination oder einem Datenbereich innerhalb des POV in einer Profitability and Cost Management-Anwendung.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

```
epmautomate clearPOV APPLICATION_NAME POV_NAME [QUERY_NAME]
PARAMETER=VALUE stringDelimiter="DELIMITER", wobei Folgendes gilt:
```

- *APPLICATION_NAME* ist der Name einer Profitability and Cost Management-Anwendung.
- *POV_NAME* ist ein POV in der Anwendung. Dieser Wert ist erforderlich.
- *QUERY_NAME* (optional) ist der Name einer Abfrage genau gemäß Definition in Profitability and Cost Management. Wenn angegeben, wird diese Abfrage verwendet, um den Datenbereich innerhalb des POV zu löschen.

Hinweis:

Wenn Sie einen Abfragenamen angeben, müssen Sie den Wert aller Laufzeitparameter (siehe unten) auf "false" setzen.

- *PARAMETER=VALUE* gibt Laufzeitparameter und deren Werte zum Leeren des POV an. Geben Sie so viele Parameter- und Wertpaarungen an, wie der Prozess erfordert. Gültige Parameter, von denen mindestens einer erforderlich ist, und ihre Werte:
 - *isManageRule=true|false* gibt an, ob Regeln gelöscht werden sollen.
 - *isInputData=true|false* gibt an, ob Eingabedaten gelöscht werden sollen.
 - *isAllocatedValues=true|false* gibt an, ob Umlagewerte gelöscht werden sollen.
 - *isAdjustmentValues=true|false* gibt an, ob Anpassungswerte gelöscht werden sollen.

 **Hinweis:**

Parameterwerte (`true` oder `false`) müssen in Kleinbuchstaben angegeben werden.

Um Datenbereiche in einem POV zu löschen (wenn ein Wert für `QUERY_NAME` angegeben ist), müssen Sie den Wert der Laufzeitparameter (`isManageRule`, `isInputData`, `isAllocatedValues` und `isAdjustmentValues`) auf `false` setzen.

- `stringDelimiter="DELIMITER"` gibt das in POV-Werten verwendete Trennzeichen an. Trennzeichen müssen in doppelte Anführungszeichen gesetzt werden.

Beispiele

- **Alle Modellartefakte und Daten aus einem POV löschen:** `epmautomate clearPOV BksML12 2012_Jan_Actual isManageRule=true isInputData=true isAllocatedValues=true isAdjustmentValues=true stringDelimiter="_"`
- **Datenbereich innerhalb eines POV löschen:** `epmautomate clearPOV BksML12 2012_Jan_Actual queryName=BksML12_2012_Jan_clear_query isManageRule=false isInputData=false isAllocatedValues=false isAdjustmentValues=false stringDelimiter="_"`

cloneEnvironment

Klont die aktuelle Umgebung und (optional) Artefakte der Identitätsdomain (Benutzer und vordefinierte Rollenzuweisungen), Datensätze des Datenmanagement, Auditdatensätze, Jobkonsolendatensätze, Inhalte von Inbox und Outbox sowie gespeicherte Snapshots. Dieser Befehl stellt eine Alternative zur Verwendung der Funktion "Umgebung klonen" in Migration dar.

 **Note:**

- **Account Reconciliation:** Nach dem Klonen werden die Zielanwendungseinstellungen für Account Reconciliation auf die Standardwerte zurückgesetzt. Wenn Sie die Zielanwendungseinstellungen beibehalten möchten, exportieren Sie sie mit dem Befehl `exportARApplicationProperties` aus der Quellumgebung. Importieren Sie die Anwendungseigenschaften dann nach Abschluss des Klonverfahrens mit dem Befehl `importARApplicationProperties` in die Zielumgebung.
- **Datenmanagement:** Das Klonen von Datensätzen der Komponente Datenmanagement kann sehr lange dauern, wenn die Staging-Tabellen sehr viele Datensätze enthalten. Auch das Klonen der Inhalte von Inbox und Outbox sowie von gespeicherten Snapshots kann beträchtliche Zeit in Anspruch nehmen, insbesondere wenn große Datenmengen enthalten sind.
- **Legacy-Umgebungen:** Beim Klonen wird die Oracle Essbase-Version wie in den folgenden Szenarios beschrieben beibehalten:
 - Szenario 1: Sie klonen eine Legacy-Quellumgebung, die eine Essbase-Version verwendet, die keine Hybrid-Cubes unterstützt, in eine Legacy-Zielumgebung, die eine Essbase-Version verwendet, die Hybrid-Cubes unterstützt. In diesem Szenario wird für die Essbase-Version in der Zielumgebung ein Downgrade durchgeführt, sodass sie mit der Version in der Quellumgebung übereinstimmt.
 - Szenario 2: Sie klonen eine Legacy-Quellumgebung, die eine Essbase-Version verwendet, die Hybrid-Cubes unterstützt, in eine Legacy-Zielumgebung, die eine Essbase-Version verwendet, die keine Hybrid-Cubes unterstützt. In diesem Szenario wird für die Essbase-Version in der Zielumgebung ein Upgrade durchgeführt, sodass sie mit der Version in der Quellumgebung übereinstimmt.
 - Szenario 3: Sie klonen eine Legacy-Quellumgebung, die eine Essbase-Version verwendet, die keine Hybrid-Cubes unterstützt, in eine EPM Standard Cloud Service- oder EPM Enterprise Cloud Service-Zielumgebung, die standardmäßig eine Essbase-Version verwendet, die Hybrid-Cubes unterstützt. In diesem Szenario wird für die Essbase-Version in der Zielumgebung kein Downgrade durchgeführt, sodass sie mit der Version in der Quellumgebung übereinstimmt.
- **Planning:** Das Klonen kann unter Umständen nicht erfolgreich verlaufen, wenn der Planning-Geschäftsprozess ein umbenanntes vordefiniertes Periodenelement enthält, das durch ein benutzerdefiniertes Periodenelement ersetzt wurde. Beispiel: Sie haben das vordefinierte Periodenelement *YearTotal* in *unused_YearTotal* umbenannt und dann ein Periodenelement mit einem anderen Typ mit dem ursprünglichen vordefinierten Elementnamen (in diesem Beispiel *YearTotal*) hinzugefügt. In diesem Szenario kann es vorkommen, dass das Klonen der Umgebung nicht erfolgreich ist.
- **Zeitpunkt zum Ausführen dieses Befehls:** Führen Sie diesen Befehl nach der geplanten täglichen Wartung der Quell- und der Zielumgebung aus.

Die tägliche Wartung kann während des Klonprozesses weder für die Quell- noch für die Zielumgebung gestartet werden. Sie wird zur nächsten geplanten Zeit nach dem Ende des Klonprozesses oder der automatischen Beendigung nach 36 Stunden ausgeführt.

Wenn dieser Befehl ausgeführt wird, während der Snapshot der Quellumgebung generiert wird (z.B. während der täglichen Wartung), tritt der Fehler `Datei nicht gefunden auf`.

Ausführliche Informationen zu diesen Themen finden Sie unter EPM Cloud-Umgebungen klonen in der Dokumentation *Migration für Oracle Enterprise Performance Management Cloud verwalten*.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Die Rolle Identitätsdomainadministrator ist zum Klonen von Benutzern und vordefinierten Rollen erforderlich.

Verwendung

```
epmAutomate cloneEnvironment TARGET_USERNAME TARGET_PASSWORD TARGET_URL
[SnapshotName=NAME] [UsersAndPreDefinedRoles=true|false] [DataManagement=true|
false] [appAudit=true|false] [jobConsole=true|false]
[storedSnapshotsAndFiles=true|false] [DailyMaintenanceStartTime=true|false],
wobei Folgendes gilt:
```

 **Note:**

- Der Parameter `dataManagement` gilt nicht für Oracle Enterprise Data Management Cloud- und Narrative Reporting-Umgebungen. Klonen Sie Datenmanagement-Datensätze nur, wenn für die Quell- und Zielumgebung dieselbe monatliche Aktualisierung gilt oder wenn die Zielumgebung genau eine neuere Aktualisierung aufweist als die Quellumgebung. Beispiel: Sie können die Datenmanagement-Datensätze für 22.01 nur in eine andere 22.01-Umgebung oder in eine 22.02-Umgebung klonen.
- Der Parameter `jobConsole` gilt nur für Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategic Workforce Planning.
- Der Parameter `appAudit` gilt nur für Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Sales Planning und Strategic Workforce Planning. Auditinformationen für Financial Consolidation and Close und Tax Reporting sind standardmäßig im Snapshot enthalten.
- Wenn der Parameter `dataManagement`, `jobConsole` oder `appAudit` für eine Umgebung nicht anwendbar ist, ignoriert EPM Automate den angegebenen Wert.

- `TARGET_USERNAME` entspricht der ID eines Serviceadministrators in der Zielumgebung. Sie müssen den Zielbenutzernamen der Identitätsdomain verwenden (nicht den SSO-Benutzernamen). Wenn Sie planen, Benutzer- und Rollenzuweisungen in der Zielumgebung zu klonen, muss dieser Benutzer auch über die Rolle Identitätsdomainadministrator verfügen.
- `TARGET_PASSWORD` ist der Speicherort der verschlüsselten Kennwortdatei des Benutzers, der durch `TARGET_USERNAME` identifiziert wird.
- `TARGET_URL` ist die URL der Umgebung, die zur geklonten Umgebung wird.
- `SnapshotName` entspricht optional dem Namen eines Snapshots, der für das Klonen verwendet werden muss. Dieser Snapshot muss in der Quellumgebung vorhanden sein. Die Standardeinstellung lautet `Artifact Snapshot`. Dabei wird der letzte Wartungs-Snapshot zum Klonen der Umgebung verwendet.
- `UsersAndPreDefinedRoles` (optional) gibt an, ob Benutzer und ihre vordefinierten Rollenzuweisungen geklont werden sollen (Gruppen für die Zugriffskontrolle werden immer geklont). Der Standardwert ist `false`. Damit diese Option funktioniert, muss der Benutzer, der durch `TARGET_USER_NAME` identifiziert wird, die Rolle Identitätsdomainadministrator in der Zielumgebung aufweisen.

Der Import von Benutzern und ihren vordefinierten Rollen verläuft nicht erfolgreich, wenn ein Benutzer, der kein Identitätsdomainadministrator ist, dieses Kontrollkästchen aktiviert und anschließend eine Umgebung klonet. Ein Fehler wie der Folgende wird im Migrationsstatusbericht aufgezeichnet: `Externes Verzeichnisartefakt <artifact_name> konnte nicht exportiert werden. Benutzer <user_name> ist nicht zur Ausführung dieses`

Vorgangs autorisiert. Der Benutzer muss über die Rolle "Identitätsdomainadministrator" verfügen, um diesen Vorgang auszuführen.

- Wenn Sie keine Benutzer importieren, und ein Benutzer im Quell-Snapshot nicht zu einer vordefinierten Rolle in der Zielumgebung zugewiesen ist, wird ein Fehler (EPMIE-00070: Benutzer konnte beim Import von zugewiesenen Rollen nicht gefunden werden) angezeigt.
 - Die Rollenzuweisung Identitätsdomainadministrator wird nicht geklont. Benutzer, denen nur die Rolle Identitätsdomainadministrator zugewiesen ist, werden nicht in die Zielumgebung geklont.
Benutzer, die einer Kombination aus der Rolle Identitätsdomainadministrator und vordefinierten Rollen in der Quellumgebung zugewiesen sind, werden geklont, aber nur den entsprechenden vordefinierten Rollen in der Zielumgebung zugewiesen. Diese Benutzer erhalten in der Zielumgebung nicht die Rolle Identitätsdomainadministrator.
 - Änderungen an den vordefinierten Rollen des Benutzers werden basierend auf den im Quell-Snapshot zugewiesenen Rollen aktualisiert. Rollenzuweisungen im Ziel werden jedoch nicht entfernt, damit sie denen im Quell-Snapshot entsprechen.
Beispiel: `jdoe` ist der vordefinierten Rolle Poweruser in der Zielumgebung zugewiesen, hat aber nur die Rolle Benutzer im Quell-Snapshot. In dieser Situation weist dieser Befehl `jdoe` der Rolle Benutzer zu und entfernt nicht die Zuweisung der Rolle Poweruser in der Zielumgebung.
 - Mit diesem Befehl werden Benutzer, die nicht im Quell-Snapshot vorhanden sind, nicht aus der Zielumgebung gelöscht. Beispiel: `jdoe` hat einen Account in der Zielumgebung, der aber nicht im Quell-Snapshot vorhanden ist. In dieser Situation wird der Account von `jdoe` nicht aus der Zielumgebung gelöscht.
 - Mit diesem Befehl werden Benutzer hinzugefügt, die nicht in der Zielumgebung vorhanden sind. Aktuelle Benutzereigenschaften werden nicht in der Zielumgebung aktualisiert, selbst wenn sie sich von denen im Quell-Snapshot unterscheiden.
Beispiel: Der Nachname von `jdoe` ist im Quell-Snapshot anders geschrieben als in der Zielumgebung. Diese Änderung wird nicht in der Zielumgebung vorgenommen. Neuen Benutzern in der Zielumgebung wird ein Zufallskennwort zugewiesen. Die neuen Benutzer erhalten E-Mails zur Accountaktivierung, in denen sie aufgefordert werden, das Kennwort zu ändern.
 - Mit diesem Befehl werden Benutzerkennwörter in der Zielumgebung nicht geändert, selbst wenn sie im Quell-Snapshot anders lauten.
- `dataManagement=true|false` (optional) klonet Datenmanagement-Datensätze von der Quellumgebung in die Zielumgebung. Die Standardeinstellung lautet `true`. Dabei werden Datenmanagement-Datensätze geklont. Setzen Sie diesen Wert auf `false`, wenn Sie keine Datenmanagement-Datensätze klonen möchten.
 - `appAudit=true|false` (optional) klonet die Auditdatensätze von der Quellumgebung in die Zielumgebung. Die Standardeinstellung lautet `true`. Dabei werden Auditdaten der Anwendung geklont. Setzen Sie diesen Wert auf `false`, wenn Sie keine Auditdaten der Anwendung in die Zielumgebung klonen möchten.
 - `jobConsole=true|false` (optional) klonet die Jobkonsolendatensätze von der Quellumgebung in die Zielumgebung. Der Standardwert ist `true`. Setzen Sie diesen Wert auf `false`, wenn Sie keine Jobkonsolendatensätze klonen möchten.
 - `storedSnapshotsAndFiles` (optional) gibt an, ob der Befehl die Inhalte von Inbox und Outbox sowie gespeicherte Snapshots klonen soll. Der Standardwert ist `false`.

 **Note:**

Nur die Ordner der obersten Ebene der Inbox und Outbox werden geklont, Unterordner werden nicht geklont. Wenn Sie die Inhalte der Unterordner behalten möchten, sichern Sie diese auf einem lokalen Computer, und laden Sie sie in die Zielumgebung hoch.

- *DailyMaintenanceStartTime* (optional) setzt die Startzeit der Wartung der geklonten Zielumgebung auf die Startzeit der Quellumgebung zurück. Der Standardwert ist `true`. Um die aktuelle Startzeit der Wartung der Zielumgebung beizubehalten, setzen Sie diesen Wert auf `false`.

Beispiele

- Klonen Sie die Umgebung, Benutzer und vordefinierte Rollenzuweisungen, Auditdaten, Jobkonsolendatensätze und Datenmanagement-Datensätze. Ändern Sie zudem die Startzeit der Wartung der Zielumgebung in die Startzeit der Quellumgebung:

```
epmAutomate cloneEnvironment serviceAdmin Password.epw https://test-cloudpln.pbc.us1.oraclecloud.com UsersAndPreDefinedRoles=true
```
- Klonen Sie die Umgebung, einschließlich der Inhalte von Inbox und Outbox sowie gespeicherter Snapshots, jedoch nicht die Benutzer und vordefinierten Rollenzuweisungen, Datenmanagement-Datensätze, Auditdaten und Jobkonsolendatensätze, ohne die Startzeit der Wartung der Zielumgebung zu ändern:

```
epmAutomate cloneEnvironment serviceAdmin Password.epw https://test-cloudpln.pbc.us1.oraclecloud.com DataManagement=false appAudit=false jobConsole=false storedSnapshotsAndFiles=true DailyMaintenanceStartTime=false
```
- Klonen Sie die gesamte Umgebung (Benutzer und vordefinierte Rollenzuweisungen, Auditdaten, Jobkonsolendatensätze, Inhalte von Inbox und Outbox, gespeicherte Snapshots und Datenmanagement-Datensätze) mit einem benutzerdefinierten Snapshot. Ändern Sie zudem die Startzeit der Wartung der Zielumgebung in die Startzeit der Quellumgebung:

```
epmAutomate cloneEnvironment serviceAdmin Password.epw https://test-cloudpln.pbc.us1.oraclecloud.com UsersAndPreDefinedRoles=true storedSnapshotsAndFiles=true SnapshotName=SampleSnapshot
```

copyDataByPointOfView

Kopiert Daten aus einem Quell-POV in einem Cube in einen Ziel-POV in demselben oder einem anderen Enterprise Profitability and Cost Management-Cube.

Gilt für

Enterprise Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmAutomate copyDataByPointOfView SOURCE_POV_NAME TARGET_POV_NAME
copyType=ALL_DATA|INPUT SOURCE_CUBE_NAME TARGET_CUBE_NAME [PARAMETER=VALUE].
```

Dabei gilt:

- *SOURCE_POV_NAME* ist der Name des Quell-POV, aus dem Daten kopiert werden sollen.
- *TARGET_POV_NAME* ist der Name eines gültigen Ziel-POV, in den die Daten aus der Quelle kopiert werden sollen.
- *copyType* gibt die Daten an, die aus dem Quell-POV kopiert werden sollen. Gültige Werte sind:
 - *ALL_DATA* gibt an, dass alle Eingabedaten und berechneten Daten in den Ziel-POV kopiert werden sollen.
 - *INPUT* gibt an, dass alle Eingabedaten, einschließlich Treiberdaten, in den Ziel-POV kopiert werden sollen.
- *SOURCE_CUBE_NAME* ist der Name des Cubes mit dem Quell-POV.
- *TARGET_CUBE_NAME* ist der Name des Cubes mit dem Ziel-POV.
- *PARAMETER=VALUE* gibt optionale Laufzeitparameter und deren Werte an. Geben Sie so viele Parameter- und Wertpaarungen an, wie der Prozess erfordert. Gültige Parameter und deren Werte:
 - *povDelimiter* ist das optionale Trennzeichen in POV-Werten. Der Standardwert lautet `::` (doppelter Doppelpunkt). Dieser Wert muss in doppelten Anführungszeichen angegeben werden. Beispiel: `povDelimiter="_"`. Neben dem Standardtrennzeichen werden nur die folgenden Trennzeichen unterstützt: `_` (Unterstrich), `#` (Rautezeichen), `&` (Et-Zeichen), `~` (Tilde), `%` (Prozentsatz), `;` (Semikolon), `:` (Doppelpunkt), `-` (Strich).
 - *createDestPOV=true|false* gibt an, ob der Ziel-POV erstellt werden soll, wenn er noch nicht vorhanden ist. Der Standardwert ist `false`. Sie müssen diesen Parameterwert auf `true` setzen, wenn der Ziel-POV nicht vorhanden ist.

Beispiele

- Alle Daten in einen anderen POV im selben Cube kopieren:


```
epmAutomate copyDataByPointOfView FY21_Jan_Actual_Working
FY22_Jan_Actual_Working ALL_DATA PCM_CLC PCM_CLC povDelimiter="_"
createDestPOV=true
```
- Alle Daten in einen anderen POV in einem anderen Cube kopieren:


```
epmAutomate copyDataByPointOfView FY21_Jan_Actual_Working
FY22_Jan_Actual_Working ALL_DATA PCM_CLC PCM_REP povDelimiter="_"
createDestPOV=true
```
- Eingabedaten in einen anderen POV im selben Cube kopieren:


```
epmAutomate copyDataByPointOfView FY21_Jan_Actual_Working
FY22_Jan_Actual_Working INPUT PCM_CLC PCM_CLC povDelimiter="_"
createDestPOV=true
```
- Eingabedaten in einen anderen POV in einem anderen Cube kopieren:


```
epmAutomate copyDataByPointOfView FY21_Jan_Actual_Working
FY22_Jan_Actual_Working INPUT PCM_CLC PCM_REP povDelimiter="_"
createDestPOV=true
```

copyDataByProfile

Kopiert Daten für die Elemente (z.B. Bereiche), die in einem Profil zum Kopieren von Daten identifiziert werden.

Gilt für

Financial Consolidation and Close, Tax Reporting

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate copyDataByProfile PROFILE_NAME`, wobei `PROFILE_NAME` der Name eines Profils zum Kopieren von Daten ist, das in Financial Consolidation and Close und Tax Reporting definiert ist.

Beispiel

```
epmautomate copyDataByProfile copyDataProfile_01
```

copyFileFromInstance

Kopiert eine Datei aus einer Quellumgebung in die Umgebung, von der aus Sie diesen Befehl ausführen.

Melden Sie sich vor dem Ausführen dieses Befehls mit EPM Automate bei der Umgebung an, in die Sie die Datei kopieren möchten.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator, Poweruser, die der Anwendungsrolle Migrationsadministrator zugewiesen sind

Verwendung

```
epmautomate copyFileFromInstance SOURCE_FILE_NAME USERNAME PASSWORD_FILE  
URL TARGET_FILE_NAME wobei Folgendes gilt:
```

- `SOURCE_FILE_NAME` ist der Name der Datei (einschließlich Erweiterung), die Sie aus der Quellumgebung kopieren möchten.
- `USERNAME` ist der Benutzername für einen Serviceadministrator der Quellumgebung.
- `PASSWORD_FILE` ist der Name und der Speicherort der Datei, in der das verschlüsselte Kennwort für den Serviceadministrator der Quellumgebung enthalten ist.

- *URL* ist die URL der Quellumgebung.
- *TARGET_FILE_NAME* ist ein eindeutiger Name für die Datei (einschließlich Erweiterung) in der Umgebung, in der Sie diesen Befehl ausführen.

Beispiel

```
epmautomate copyFileFromInstance "my data file.zip" serviceAdmin
C:\mySecuredir\password.epw https://test-cloud-pln.pbcs.us1.oraclecloud.com "my
target data file.zip"
```

copyFromObjectStorage

Kopiert eine Datei oder einen Backup-Snapshot aus einer Oracle Object Storage-Gruppe in die aktuelle Umgebung.

Wenn Sie einen Backup-Snapshot kopieren, wird er mit diesem Befehl aus der Object Storage-Gruppe kopiert, und sein Inhalt wird in Oracle Enterprise Performance Management Cloud extrahiert.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate copyFromObjectStorage USERNAME PASSWORD URL TARGET_FILE_NAME wobei
Folgendes gilt:
```

- *USERNAME* ist die ID eines Benutzers, der in Oracle Object Storage Cloud über die erforderlichen Zugriffsberechtigungen verfügt.
Bei im föderierten Identitätsprovider erstellten Benutzern geben Sie den vollqualifizierten Namen des Benutzers an (Beispiel: `exampleIdP/jdoe` oder `exampleIdP/john.doe@example.com`, wobei `exampleIdP` der Name des föderierten Identitätsproviders ist). Geben Sie bei anderen Benutzern die Benutzer-ID an.
- *PASSWORD* ist das Swift-Kennwort oder das Authentifizierungstoken, das mit dem Benutzer verknüpft ist. Dieses Kennwort stimmt nicht mit dem Kennwort überein, das Sie für die Anmeldung bei der Object Storage-Konsole verwenden. Das Authentifizierungstoken ist ein von Oracle generiertes Token, mit dem Sie Drittanbieter-APIs authentifizieren können, beispielsweise für die Authentifizierung mit einem Swift-Client. Anweisungen zum Erstellen dieses Token finden Sie unter [So erstellen Sie ein Authentifizierungstoken](#): in der Dokumentation *Oracle Cloud Infrastructure-Dokumentation*.
- *URL* ist die URL der Oracle Object Storage Cloud-Gruppe, einschließlich des Gruppennamens und des Namens des zu kopierenden Objekts.
Das URL-Format:

```
https://swiftobjectstorage.region_identifier.oraclecloud.com/v1/namespace/
bucket_name/object_name
```

Komponenten dieser URL:

- `region_identifier` ist eine Region für das Hosting von Oracle Cloud Infrastructure.
- `namespace` ist der Container der obersten Ebene für alle Gruppen und Objekte. Beim Erstellen des Accounts wird jedem Mandanten von Oracle Cloud Infrastructure ein eindeutiger, vom System generierter und unveränderbarer Object Storage-Namespace-Name zugewiesen. Der Namespace-Name Ihres Mandanten, zum Beispiel `axaxnpcrorw5`, gilt für alle Regionen.
- `bucket_name` ist der Name eines logischen Containers, in dem Sie Ihre Daten und Dateien speichern. Gruppen werden in Compartments organisiert und verwaltet. Ein vom System generierter Gruppenname, zum Beispiel `bucket-20210301-1359`, enthält das aktuelle Jahr, den Monat, den Tag und die Uhrzeit.
- `object_name` ist der Name des Snapshots oder der Datei, die Sie aus Oracle Object Storage Cloud kopieren möchten. Dieser Wert muss exakt mit dem vollständigen Namen des Objekts in Object Storage Cloud übereinstimmen. Verwenden Sie Erweiterungen wie `.zip` nur, wenn diese auch im Objektnamen enthalten sind.

Weitere Informationen finden Sie unter den folgenden Themen in der *Oracle Cloud Infrastructure-Dokumentation*

- [Regionen und Availability-Domains](#)
 - [Erläuterungen zu Object Storage-Namespace](#)
 - [Gruppen verwalten](#)
- `TARGET_FILE_NAME` ist ein eindeutiger Name für die Datei oder den Snapshot in der EPM Cloud-Umgebung. Geben Sie beim Kopieren von Snapshots nicht die ZIP-Erweiterung an, damit dieser Dateiname mit dem Befehl `importSnapshot` verwendet werden kann.
Dateien, die größer als 100 MB sind, werden in Oracle Object Storage zusammen mit der Manifestdatei, die die entsprechenden Segmente angibt, in einem logischen Verzeichnis gespeichert. Geben Sie den Namen des logischen Verzeichnisses als `TARGET_FILE_NAME` an.

Beispiele

Ersetzen Sie in diesen Beispielen `URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET` mit einer aktiven URL in diesem Format: `https://`

```
swiftobjectstorage.region_identifier.oraclecloud.com/v1/namespace/
bucket_name/.
```

- Kopieren Sie einen Snapshot mit der Bezeichnung `backup_Snapshot_12_05_20.zip` aus einer Oracle Object Storage-Gruppe in EPM Cloud, und benennen Sie ihn um:

```
epmautomate copyFromObjectStorage oracleidentitycloudservice/jDoe
example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/
backup_Snapshot_12_05_20.zip snapshot_from_osc
```
- Kopieren Sie einen Snapshot mit der Bezeichnung `backup_Snapshot_12_05_20` aus einer Oracle Object Storage-Gruppe in EPM Cloud, und benennen Sie ihn um:

```
epmautomate copyFromObjectStorage oracleidentitycloudservice/jDoe
example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/
backup_Snapshot_12_05_20 snapshot_from_osc
```

- Kopieren Sie einen Snapshot mit der Bezeichnung `backup_snapshot_12_05_20` aus einer Oracle Object Storage-Gruppe in EPM Cloud, ohne ihn umzubenennen:

```
epmautomate copyFromObjectStorage oracleidentitycloudservice/jDoe  
example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/backup_snapshot_12_05_20  
backup_snapshot_12_05_20
```
- So kopieren Sie eine Datei aus einer Oracle Object Storage-Gruppe in EPM Cloud:

```
epmautomate copyFromObjectStorage oracleidentitycloudservice/jDoe  
example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/example_file.txt  
copied_from_osc.txt
```

copyOwnershipDataToNextYear

Kopiert die Anteilsdaten aus der letzten Periode eines Jahres in die erste Periode des nächsten Jahres.

Die initialen Standard- und Überschreibungseinstellungen für den Anteil werden automatisch von Periode zu Periode innerhalb desselben Jahres, jedoch nicht auf die Perioden in nachfolgenden Jahren übertragen. Um die aktuellsten Anteilseinstellungen aus der letzten Periode in einem Jahr in die erste Periode des nächsten Jahres zu übertragen, müssen Sie die Anteilseinstellungen aus der letzten Periode des Jahres im POV auf die erste Periode des nächsten Jahres übertragen.

Gilt für

Financial Consolidation and Close und Tax Reporting.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer

Verwendung

`epmautomate copyOwnershipDataToNextYear Scenario Year` wobei:

- `Scenario` der Name des Szenarios ist, aus dem Anteilsdaten kopiert werden sollen.
- `Year` das Jahr ist, aus dem Anteilsdaten in die erste Periode des nächsten Jahres kopiert werden sollen.

Beispiel

```
epmautomate copyOwnershipDataToNextYear FCCS_total_Actual FY18
```

copyPOV

Kopiert die Modellartefakte und Oracle Essbase-Cube-Daten von einem Quell-POV zu einem Ziel-POV.

Gilt für

Profitability and Cost Management.

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate copyPOV APPLICATION_NAME SOURCE_POV_NAME TARGET_POV_NAME
PARAMETER=VALUE stringDelimiter="DELIMITER" [isInputData=true|false
isAllInputData=true|false]`, wobei:

- `APPLICATION_NAME` der Name der Profitability and Cost Management-Anwendung ist, die den Quell-POV enthält.
- `SOURCE_POV_NAME` der Name des Quell-POV in der angegebenen Anwendung ist.
- `TARGET_POV_NAME` der Name eines gültigen Ziel-POVs im Status Draft (Entwurf) ist.
- `PARAMETER=VALUE` Laufzeitparameter und deren Werte zum Kopieren des POV angibt. Geben Sie so viele Parameter- und Wertpaarungen an, wie der Prozess erfordert. Gültige Parameter und deren Werte:
 - `isManageRule=true|false` gibt an, ob Regeln kopiert werden sollen.
 - `isInputData=true | isAllData=true | isAllInputData=true` (optional) gibt an, wie Daten kopiert werden sollen. Für diese Parameter ist der Standardwert "false". Geben Sie nur für einen davon "true" an:
 - * Geben Sie `isInputData=true` an, um Eingabedaten an den Ziel-POV zu kopieren.
 - * Geben Sie `isAllData=true` an, um alle Eingabedaten und berechneten Daten an den Ziel-POV zu kopieren.
 - * Geben Sie `AllInputData=true` an, um alle Eingabedaten, einschließlich Treiberdaten, an den Ziel-POV zu kopieren.
 - `modelViewName=NAME` gibt den Namen des Datensegments an, das aus dem Quell-POV in den Ziel-POV kopiert werden soll.
 - `createDestPOV=true|false` gibt an, ob der Ziel-POV erstellt werden soll, wenn er noch nicht vorhanden ist.
 - `nonEmptyTupleEnabled=true|false` gibt an, ob nicht leere Tupel (NET, Non-Empty Tuple) aktiviert werden sollen, sodass der Befehl nur die Schnittmengen berücksichtigt, die Daten aufweisen. Der Standardwert ist `true`, was in seltenen Fällen dazu führen kann, dass der Befehl beim Kopieren von Essbase-Daten nicht ordnungsgemäß ausgeführt wird. Überschreiben Sie den Standardwert in diesen Fällen mit `nonEmptyTupleEnabled=false`, um die Performance zu verbessern.

Hinweis:

Parameterwerte (`true` oder `false`) müssen in Kleinbuchstaben angegeben werden.

- `stringDelimiter="DELIMITER"` gibt das in POV-Werten verwendete Trennzeichen an. Trennzeichen müssen in doppelte Anführungszeichen gesetzt werden.

Beispiele

- `epmautomate copyPOV BksML12 2012_Jan_Actual 2012_Feb_Actual isManageRule=true isInputData=true modelViewName="Balancing - 5 Customer Costs" createDestPOV=true stringDelimiter="_"`
- `epmautomate copyPOV BksML12 2012_Jan_Actual 2012_Feb_Actual isManageRule=true isAllInputData=true createDestPOV=true stringDelimiter="_"`
- `epmautomate copyPOV BksML12 2012_Jan_Actual 2012_Feb_Actual isManageRule=true isAllData=true createDestPOV=true stringDelimiter="_"`

copySnapshotFromInstance

Kopiert den aktuellen Snapshot aus einer Quellumgebung in die Umgebung (Ziel), von der aus Sie diesen Befehl ausführen.

Dieser Befehl wird hauptsächlich als erster Schritt für die Migration einer Umgebung verwendet, indem der aktuelle Snapshot aus einer anderen Umgebung kopiert wird, z.B. aus einer Testumgebung in eine Produktionsumgebung. Sie verwenden den Befehl [importSnapshot](#), um den Migrationsprozess abzuschließen.

Bevor Sie diesen Befehl ausführen, starten Sie eine EPM Automate-Session, und melden Sie sich bei der Zielumgebung an.

Wenn dieser Befehl zum Kopieren des aktuellen Snapshots ausgeführt wird, während der Snapshot der Quellumgebung generiert wird (z.B. während der täglichen Wartung), tritt der Fehler `Datei nicht gefunden auf`.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator, Poweruser, die der Anwendungsrolle Migrationsadministrator zugewiesen sind

Verwendung

`epmautomate copySnapshotFromInstance SNAPSHOT_NAME USERNAME PASSWORD_FILE URL`
wobei Folgendes gilt:

- `SNAPSHOT_NAME` ist der Name eines vorhandenen Snapshots in der Quellumgebung.
- `USERNAME` ist der Benutzername für einen Serviceadministrator der Quellumgebung.
- `PASSWORD_FILE` ist der Name und der Speicherort der Datei, in der das verschlüsselte Kennwort für den Serviceadministrator der Quellumgebung enthalten ist.
- `URL` ist die URL der Quellumgebung.

Beispiel

```
epmautomate copySnapshotFromInstance "Artifact Snapshot" serviceAdmin  
C:\mySecuredir\password.epw https://test-cloud-pln.pbcs.us1.oraclecloud.com
```


copyToObjectStorage

Kopiert eine Datei oder einen Snapshot aus der aktuellen Umgebung in eine Oracle Object Storage Cloud-Gruppe.

Wenn Sie einen Snapshot kopieren, komprimiert dieser Befehl den Inhalt des Snapshots vor dem Kopieren in Oracle Object Storage.

Damit Sie Dateien schneller kopieren können, teilt dieser Befehl große Dateien (größer als 100 MB) in Segmente von je 10 MB auf (*FILE_NAME/FILE_NAME_object_store_bytes_seg_0* bis *FILE_NAME/FILE_NAME_object_store_bytes_seg_n*) und erstellt eine Manifestdatei (*FILE_NAME/FILE_NAME.manifest*). Dateisegmente werden gemeinsam mit der Manifestdatei in Oracle Object Storage gespeichert. In der Object Storage-Konsole wird die Datei als logisches Verzeichnis angezeigt, das die Dateisegmente und die Manifestdatei enthält.

Dateien, die kleiner als 100 MB sind, werden nicht in Segmente aufgeteilt und mit dem ursprünglichen Dateinamen gespeichert.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate copyToObjectStorage SOURCE_FILE_NAME USERNAME PASSWORD URL`
wobei Folgendes gilt:

- *SOURCE_FILE_NAME* ist der Name der Datei oder des Snapshots in der Oracle Enterprise Performance Management Cloud. Geben Sie beim Kopieren eines Snapshots nicht die ZIP-Erweiterung an.
- *USERNAME* ist die ID eines Benutzers, der in Oracle Object Storage Cloud über die erforderlichen Zugriffsberechtigungen zum Schreiben verfügt. Bei im föderierten Identitätsprovider erstellten Benutzern geben Sie den vollqualifizierten Namen des Benutzers an (Beispiel: `exampleIdP/jdoe` oder `exampleIdP/john.doe@example.com`, wobei `exampleIdP` der Name des föderierten Identitätsproviders ist). Geben Sie bei anderen Benutzern die Benutzer-ID an.
- *PASSWORD* ist das Swift-Kennwort oder das Authentifizierungstoken, das mit dem Benutzer verknüpft ist. Dieses Kennwort stimmt nicht mit dem Kennwort überein, das Sie für die Anmeldung bei der Object Storage-Konsole verwenden. Das Authentifizierungstoken ist ein von Oracle generiertes Token, mit dem Sie Drittanbieter-APIs authentifizieren können, beispielsweise für die Authentifizierung mit einem Swift-Client. Anweisungen zum Erstellen dieses Token finden Sie unter [So erstellen Sie ein Authentifizierungstoken](#): in der Dokumentation *Oracle Cloud Infrastructure-Dokumentation*.
- *URL* ist die URL der Oracle Object Storage Cloud-Gruppe, der ein optionaler Objektname angehängt wurde.

Das URL-Format ohne Objektname lautet folgendermaßen:

```
https://swiftobjectstorage.region_identifizier.oraclecloud.com/v1/namespace/  
bucket_name
```

Das URL-Format mit Objektname lautet folgendermaßen:

```
https://swiftobjectstorage.region_identifizier.oraclecloud.com/v1/namespace/  
bucket_name/object_name
```

Komponenten dieser URL:

- `region_identifizier` ist eine Region für das Hosting von Oracle Cloud Infrastructure.
- `namespace` ist der Container der obersten Ebene für alle Gruppen und Objekte. Beim Erstellen des Accounts wird jedem Mandanten von Oracle Cloud Infrastructure ein eindeutiger, vom System generierter und unveränderbarer Object Storage-Namespace-Name zugewiesen. Der Namespace-Name Ihres Mandanten, zum Beispiel `axaxnpcrorw5`, gilt für alle Regionen.
- `bucket_name` ist der Name eines logischen Containers, in dem Sie Ihre Daten und Dateien speichern. Gruppen werden in Compartments organisiert und verwaltet. Ein vom System generierter Gruppenname, zum Beispiel `bucket-20210301-1359`, enthält das aktuelle Jahr, den Monat, den Tag und die Uhrzeit.
- `object_name` (optional) ist der Name, den Sie für die Datei in Oracle Object Storage Cloud verwenden können. Wenn kein Objektname angegeben ist, wird die Datei mit ihrem Originalnamen kopiert.

Weitere Informationen finden Sie unter den folgenden Themen in der *Oracle Cloud Infrastructure-Dokumentation*

- [Regionen und Availability-Domains](#)
- [Erläuterungen zu Object Storage-Namespace](#)
- [Gruppen verwalten](#)

Beispiele

Ersetzen Sie in diesen Beispielen `URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET` mit einer aktiven URL in diesem Format: `https://`

```
swiftobjectstorage.region_identifizier.oraclecloud.com/v1/namespace/bucket_name/.
```

- So kopieren Sie einen Snapshot in eine Object Storage-Gruppe und benennen ihn um:

```
epmautomate copyToObjectStorage "Artifact Snapshot"  
oracleidentitycloudservice/jDoe example_pwd  
URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/Snapshot_04_30_21
```
- So kopieren Sie eine Datei in eine Object Storage-Gruppe:

```
epmautomate copyToObjectStorage example_file.txt oracleidentitycloudservice/  
jDoe example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET
```
- So kopieren Sie eine Datei in eine Object Storage-Gruppe und benennen sie um:

```
epmautomate copyToObjectStorage example_file.txt  
eoracleidentitycloudservice/jDoe example_pwd  
URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/epm_text_file.txt
```

createGroups

Fügt Gruppen mit einer ANSI- oder UTF-8-codierten CSV-Datei, die in die Umgebung hochgeladen wurde, zur Zugriffskontrolle hinzu.

Verwenden Sie den Befehl [uploadFile](#), um Dateien in eine Umgebung hochzuladen. Das Dateiformat lautet wie folgt:

```
Group Name,Description  
Example_grp1,My test group  
Example_grp2,My other test group
```

Bei Gruppennamen wird nicht zwischen Groß- und Kleinschreibung unterschieden. Wenn die Befehlsausführung abgeschlossen ist, gibt EPM Automate Informationen zu jedem nicht erfolgreichen Eintrag an die Konsole aus. Lesen Sie diese Informationen, um zu verstehen, warum die Befehlsausführung für einige Einträge in der CSV-Datei nicht erfolgreich war.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator oder Zugriffskontrollemanager

Verwendung

`epmautomate createGroups FILE_NAME` wobei `FILE_NAME` der Name einer CSV-Datei ist, die Gruppennamen und Beschreibungen enthält.

Beispiel

```
epmautomate createGroups group_file.CSV
```

createNRSnapshot

Erstellen Sie einen On-Demand-Snapshot mit dem Namen `EPRCS_Backup.tar.gz` für eine Narrative Reporting-Umgebung.

Sie können die Datei `EPRCS_Backup.tar.gz` und die Fehlerdatei mit dem Befehl [downloadFile](#) auf einen lokalen Computer herunterladen oder mit dem Befehl [copyFileFromInstance](#) in eine andere Umgebung kopieren.

Die Anwendungsdaten in `EPRCS_Backup.tar.gz` entsprechen der letzten täglichen Wartung. Wenn Sie ein Backup neuerer Daten erstellen möchten, verwenden Sie die Narrative Reporting-Funktion zum Exportieren von Daten.

Gilt für

Narrative Reporting

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate createNRSnapshot [errorFile=Error_File.txt]` wobei `errorFile` (optional) den Namen einer eindeutigen Textdatei für die Aufzeichnung von Fehlern angibt, falls der Befehl Fehler findet.

Beispiel

```
epmautomate createNRSnapshot errorFile=EPRCS_backup_Error.txt
```

createReconciliations

Kopiert die Profile in eine angegebene Periode.

Gilt für

Account Reconciliation.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

`epmautomate createreconciliations PERIOD SAVED_FILTER` wobei Folgendes gilt:

- `PERIOD` ist der Name einer Periode.
- `SAVED_FILTER` ist der Name eines gespeicherten öffentlichen Filters. Wenn Sie keinen gespeicherten Filter angeben, kopiert EPM Automate alle passenden Profile.

Beispiele

- Alle Profile für den folgenden Zeitraum kopieren: `epmautomate createReconciliations "January 2015"`
- Profile mit einem bestimmten Filter kopieren: `epmautomate createReconciliations "January 2015" "Corporate Recs"`

deleteFile

Löscht eine Datei oder einen Snapshot am Standardspeicherort zum Hochladen, in der Inbox oder Outbox, in einem Datenmanagement-Ordner oder in "profitinbox/profitoutbox".

Um eine Datei zu löschen, die sich an einem anderen Speicherort als dem Standardspeicherort zum Hochladen befindet, müssen Sie den Speicherort der Datei angeben.

Wenn dieser Befehl zum Löschen eines Snapshots ausgeführt wird, der gerade generiert oder archiviert wird, tritt einer der folgenden Fehler auf:

- Datei nicht gefunden, wenn der Snapshot gerade generiert wird
- Archivprozess wird verarbeitet. Element kann nicht umbenannt oder gelöscht werden, wenn der Snapshot gerade archiviert wird

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, die der Anwendungsrolle Migrationsadministrator zugewiesen sind

Verwendung

```
epmautomate deleteFile FILE_NAME
```

Hinweis:

Sie müssen den Dateinamen mit der Erweiterung angeben, z.B. `data.csv`, `data.zip`, falls vorhanden. Sie können einen Snapshot löschen, ohne seine Dateiendung (ZIP) anzugeben. Diese Verwendung ist jedoch veraltet. Sie müssen den Speicherort für die Datei angeben, falls nicht der Standardspeicherort verwendet werden soll. Ausführliche Informationen finden Sie unter [Standarddateispeicherorte](#). Unterstützte Speicherorte: `inbox`, `profitinbox`, `outbox`, `profitoutbox`, `to_be_imported` und `inbox/Verzeichnisname`.

Beispiele

- Eine Datei aus dem Standardspeicherort zum Hochladen löschen:
`epmautomate deleteFile data.csv`
- Eine Datei aus der Inbox löschen:
`epmautomate deleteFile inbox/data.csv`
- Aus der Outbox löschen:
`epmautomate deleteFile outbox/data.csv`
- Einen Snapshot löschen, den Sie mit der Option Migration erstellt haben:
 - `epmautomate deleteFile "Backup 18-06-12.zip"` oder
 - `epmautomate deleteFile "Backup 18-06-12"` (veraltet)
- Aus "profitinbox" (Profitability and Cost Management) löschen:
`epmautomate deleteFile profitinbox/data.csv`
- Aus "profitoutbox" (Profitability and Cost Management) löschen:
`epmautomate deleteFile profitoutbox/data.csv`
- Aus einem Datenmanagement-Ordner zum Hochladen löschen:
`epmautomate deleteFile inbox/dm_data/data.csv`
- Aus einem Datenmanagement-Ordner löschen:
`epmautomate deleteFile outbox/dm_data/data.csv`

deleteGroups

Entfernt Gruppen basierend auf den Informationen, die in einer ANSI- oder UTF-8-codierten CSV-Datei verfügbar sind, die in die Umgebung hochgeladen wurde, aus der Zugriffskontrolle.

Verwenden Sie den Befehl [uploadFile](#), um Dateien in eine Umgebung hochzuladen. Das Dateiformat lautet wie folgt:

```
Group Name  
Example_grp1  
Example_grp2
```

Bei Gruppennamenswerten in der Datei wird nicht zwischen Groß- und Kleinschreibung unterschieden. Wenn die Befehlsausführung abgeschlossen ist, gibt EPM Automate Informationen zu jedem nicht erfolgreichen Eintrag an die Konsole aus. Lesen Sie diese Informationen, um zu verstehen, warum die Befehlsausführung für einige Einträge in der CSV-Datei nicht erfolgreich war.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator oder Zugriffskontrollemanager

Verwendung

`epmautomate deleteGroups FILE_NAME` wobei `FILE_NAME` der Name einer CSV-Datei ist, die die Namen der Gruppen enthält, die aus der Zugriffskontrolle entfernt werden sollen.

Beispiel

```
epmautomate deleteGroups group_file.CSV
```

deletePointOfView

Löscht Artefakte und Oracle Essbase-Cube-Daten aus einem POV in einer Enterprise Profitability and Cost Management-Anwendung.

Gilt für

Enterprise Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate deletePointOfView POV_NAME [povDelimiter="DELIMITER"]`. Dabei gilt:

- *POV_NAME* gibt den Namen des zu löschenden POV an.
- *povDelimiter* gibt das in POV-Werten verwendete Trennzeichen an. Der Standardwert lautet : : (doppelter Doppelpunkt). Dieser Wert muss in doppelten Anführungszeichen angegeben werden. Beispiel: `povDelimiter="_"`. Neben dem Standardtrennzeichen werden nur die folgenden Trennzeichen unterstützt: _ (Unterstrich), # (Rautezeichen), & (Et-Zeichen), ~ (Tilde), % (Prozentsatz), ; (Semikolon), : (Doppelpunkt), - (Strich).

Beispiel

- POV löschen, der ein benutzerdefiniertes POV-Trennzeichen verwendet
`epmAutomate deletePointOfView FY21_Jan_Actual_Working povDelimiter="_"`
- POV löschen, der das POV-Standardtrennzeichen verwendet
`epmAutomate deletePointOfView FY21::Jan::Actual::Working`

deletePOV

Löscht Modellartefakte und Oracle Essbase-Cube-Daten aus einem POV in Profitability and Cost Management.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate deletePOV APPLICATION_NAME POV_NAME`
`stringDelimiter="DELIMITER"`, wobei Folgendes gilt:

- *APPLICATION_NAME* ist der Name der Profitability and Cost Management-Anwendung, die den zu löschenden POV enthält.
- *POV_NAME* ist der Name des POV, der gelöscht werden soll. Dieser Wert ist erforderlich.
- `stringDelimiter="DELIMITER"` gibt das in POV-Werten verwendete Trennzeichen an. Trennzeichen müssen in doppelte Anführungszeichen gesetzt werden.

Beispiel

`epmautomate deletePOV BksML12 2012_Jan_Actual stringDelimiter="_"`

deployCube

Stellt den Berechnungs-Cube einer Profitability and Cost Management-Anwendung bereit oder erneut bereit.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate deployCube APPLICATION_NAME PARAMETER=VALUE comment="comment"`, wobei Folgendes gilt:

- `APPLICATION_NAME` ist der Name einer Profitability and Cost Management-Anwendung.
- `PARAMETER=VALUE` gibt Laufzeitparameter und deren Werte an, um den Cube bereitzustellen. Geben Sie so viele Parameter- und Wertpaarungen an, wie der Prozess erfordert. Gültige Parameter und deren Werte:

Hinweis:

Parameterwerte (`true` oder `false`) müssen in Kleinbuchstaben angegeben werden.

- `isKeepData=true|false`
gibt an, ob vorhandene Daten beibehalten werden sollen, sofern vorhanden.
- `isReplaceCube=true|false` gibt an, ob der vorhandene Cube ersetzt werden soll.

Hinweis:

Die Werte von `isKeepData` und `isReplaceCube` können nicht beide auf `true` gesetzt werden.

- `isRunNow=true|false` gibt an, ob der Prozess sofort durchgeführt werden soll.
- `comment` stellt einen optionalen Kommentar dar, der in doppelte Anführungszeichen gesetzt werden muss.

Beispiel

```
epmautomate deployCube BksML12 isKeepData=true isReplaceCube=false isRunNow=true
comment="Test cube deployment"
```

deployEJTemplates

Stellt abgeschlossene Enterprise-Journalvorlagen für offene Perioden in Financial Consolidation and Close bereit. Beim Deployment von Enterprise-Journalvorlagen werden wiederkehrende Journale erstellt, die mit der Vorlage für die ausgewählte Periode verknüpft sind. Zudem können Sie mit den bereitgestellten Vorlagen Ad-hoc-Journale erstellen.

Dieser Befehl bietet eine Alternative zur Verwendung von Financial Consolidation and Close-Fenstern für das Deployment neuer Enterprise-Journalvorlagen am Monatsanfang.

Gilt für

Financial Consolidation and Close

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate deployEJTemplates YEAR PERIOD [Template=TEMPLATE_NAME]
[ResetJournals=true|false]`, wobei Folgendes gilt:

- `Year` ist das Journaljahr.
- `Period` ist die Journalperiode. Dieser Wert kann nur angegeben werden, wenn das Jahr angegeben wurde.
- `Template=TEMPLATE_NAME` gibt die Namen der Journale an, die bereitgestellt werden sollen. Wenn Sie mehrere Journale bereitstellen möchten, vergeben Sie jeweils eindeutige Vorlagennamen im Format `Template=TEMPLATE_NAME`, z.B. `Template="Loan Details" Template="Housing Details" Template="Repayment Details"`.
Wenn dieser Parameterwert nicht angegeben ist, werden mit dem Befehl alle Vorlagen für die angegebene Kombination aus Jahr und Periode bereitgestellt.
- `ResetJournals` gibt optional an, ob alle Journale auf die erste Stufe nach dem erneuten Deployment der Vorlagen zurückgesetzt werden müssen. Der Standardwert ist `false`.
Financial Consolidation and Close validiert diesen Wert intern basierend auf Änderungen an Vorlagen und kann bei Bedarf den angegebenen Wert überschreiben.

Beispiel

```
epmautomate deployEJTemplates 2021 May Template="Loan Details"  
Template="Housing Details" ResetJournals=true
```

deployFormTemplates

Stellt abgeschlossene Formularvorlagen für neue Datenerfassungsperioden bereit, um Supplemental Data-Formulare zu erstellen, wodurch der Datenerfassungsprozess konsistent und wiederholbar wird.

Gilt für

Financial Consolidation and Close, Tax Reporting

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate deployFormTemplates COLLECTION_INTERVAL [DIMENSION] [Template]
[resetWorkFlows=true|false]` wobei Folgendes gilt:

- `COLLECTION_INTERVAL` ist der Name des Collection-Intervalls, für den die Vorlage bereitgestellt werden soll.
- `DIMENSION` (optional) gibt die Frequency-Dimensionen des Datenerfassungsprozesses im Format `DIMENSION=MEMBER_NAME` an. Geben Sie so viele Dimensionen an, wie im Collection-Intervall definiert sind (maximal vier, einschließlich Year und Period). Beispiel: `"Year=2020" "Period=July" "Product=Oracle EPM" "Consolidation=entity Input"`. Wenn dieser Parameterwert nicht angegeben ist, wird kein Standardwert verwendet.
- `Template` (optional) gibt eindeutige Namen für die bereitzustellenden Formatvorlagen im Format `Template=TEMPLATE_NAME` an. Sie können eine beliebige Anzahl an eindeutigen Namen (so viele wie erforderlich) in diesem Format angeben. Beispiel: `Template="Loan Details Template" Template="Housing Details Template" Template="Repayment Details Template"`. Wenn dieser Eigenschaftswert nicht angegeben ist, werden mit dem Befehl alle Vorlagen für das angegebene Intervall bereitgestellt.
- `resetWorkFlows` (optional) gibt an, ob nach der Bereitstellung der Formatvorlagen sämtliche Vorlagen auf die erste Phase zurückgesetzt werden sollen. Der Standardwert ist `false`.

Beispiel

```
epmautomate deployFormTemplates "Journal Collection Interval" "Year=2020"
"Period=July" "Product=Oracle EPM" "Consolidation=entity Input" Template="Loan
Details Template" Template="Housing Details Template" resetWorkFlows=true
```

deployTaskManagerTemplate

Führt das Deployment von Aufgaben aus einer Task Manager-Vorlage in einen Aufgabenplan durch. Dabei wird die konsistente Ausführung sich wiederholender Geschäftsprozesse sichergestellt.

Gilt für

Financial Consolidation and Close, Tax Reporting

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmAutomate deployTaskManagerTemplate TEMPLATE_NAME SCHEDULE_NAME YEAR PERIOD
DAY_ZERO_DATE [dateFormat=DATE_FORMAT] [orgUnit=ORGANIZATION UNIT], wobei
Folgendes gilt:
```

- `TEMPLATE_NAME` ist der Name der Task Manager-Vorlage für das Deployment.
- `SCHEDULE_NAME` ist der Zeitplanname für die Erstellung aus der Vorlage.
- `YEAR` ist das Element der Dimension "Year", in dem die Vorlage bereitgestellt werden soll.
- `PERIOD` ist das Element der Dimension "Period", in dem die Vorlage bereitgestellt werden soll.
- `DAY_ZERO_DATE` ist das Datum von Tag Null im gültigen Format, das zum Erstellen des Zeitplans verwendet werden soll.

- `dateFormat` (optional) ist das Datumsformat für das Datum von Tag Null. Das Standardformat lautet JJJJ-MM-TT.
- `orgUnit` (optional) ist der Name der Organisationseinheit. Wenn ein Wert nicht angegeben ist, wird der Zeitplan mit der standardmäßigen Datumszuordnung erstellt. Es werden keine Feiertagsregeln verwendet.

Beispiel

- Task Manager-Vorlagen für die Organisationseinheit `Ind` mit dem Standarddatumsformat (JJJJ-MM-TT) für das Datum für Tag Null bereitstellen:
`epmautomate deployTaskManagerTemplate "Vision Monthly Close" "Qtr 2 Close" 2021 July 2021-07-10 orgUnit=Ind`
- Task Manager-Vorlagen für die Organisationseinheit "Ind" mit dem Datumsformat `tt/mm/jjjj` für das Datum für Tag Null bereitstellen:
`epmautomate deployTaskManagerTemplate "Vision Monthly Close" "Qtr 2 Close" 2021 July 02/07/2021 dateFormat=dd/MM/yyyy orgUnit=Ind`

dismissIPMInsights

Automatisiert das Schließen von Daten in Intelligent Performance Management (IPM) Insight vor dem Ausführen neuer IPM Insight-Jobs. Durch das Schließen von Daten werden alle offenen Insights geschlossen, für die Sie keine Aktionen durchführen möchten. Dieser Befehl ist eine Alternative zum manuellen Schließen der Daten über das IPM Insight-Dashboard.

Gilt für

Planning, Planning Modules, Strategische Personalplanung, Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate dismissIPMInsights [comment="comment"]`, wobei "comment" (optional) eine Begründung für das Schließen offener Insights enthält.

Beispiel

`epmautomate dismissIPMInsights comment="dismissing unusable insights"`

downloadFile

Lädt eine Datei aus einer Umgebung auf den lokalen Computer herunter.

Verwenden Sie diesen Befehl, um Daten, Metadaten und Backup-Snapshots herunterzuladen und lokal zu speichern. Die Datei wird in den Ordner heruntergeladen, aus dem Sie EPM Automate ausführen.

Wenn dieser Befehl zum Herunterladen des aktuellen Snapshots ausgeführt wird, während der Snapshot der Umgebung generiert wird (z.B. während der täglichen Wartung), tritt der Fehler `Datei nicht gefunden auf`.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, die der Anwendungsrolle Migrationsadministrator zugewiesen sind

Verwendung

```
epmautomate downloadFile "[FILE_PATH]/FILE_NAME"
```

Beispiele

- Download des täglichen Wartungs-Snapshots: `epmautomate downloadFile "Artifact Snapshot"`
- Download eines benutzerdefinierten Snapshots: `epmautomate downloadFile "mySnapshot.zip"`
- Download eines Narrative Reporting-Wartungs-Snapshots: `epmautomate downloadFile "EPRCS_Backup.tar.gz"`
- Download einer Datei aus dem standardmäßigen Downloadspeicherort: `epmautomate downloadFile data.csv`
- Download von einem Data Management-Ordner: `epmautomate downloadfile outbox/dm_data/data.csv`
- Download von profitoutbox: `epmautomate downloadFile profitOutbox/data.csv`

enableApp

Aktiviert eine Anwendung.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

```
epmautomate enableapp APPLICATION_NAME wobei APPLICATION_NAME der Name der Profitability and Cost Management-Anwendung ist, die aktiviert werden soll.
```

Beispiel

```
epmautomate enableApp BksML12
```

enableQueryTracking

Aktiviert das Abfrage-Tracking bei ASO-Cubes, um mit dem Erfassen von Mustern beim Abruf von Benutzerdaten (Abfragen) zu beginnen.

Mit den erfassten Mustern beim Datenabruf wird die Aggregation von ASO-Cubes optimiert. Dieser Vorgang wird mit dem Befehl [executeAggregationProcess](#) gestartet.

Gilt für

Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate enableQueryTracking ASO_CUBE_NAME`, wobei `ASO_CUBE_NAME` dem Namen des ASO-Cubes entspricht, in dem das Abfrage-Tracking aktiviert werden soll.

Beispiel

```
epmautomate enableQueryTracking VISION_ASO
```

encrypt

Verwendet Advanced Encryption Standard (AES/CBC/PKCS5Padding(128)), um das Kennwort von Oracle Enterprise Performance Management Cloud (oder das OAuth 2.0-Aktualisierungstoken und die Client-ID für den Zugriff auf OCI-Umgebungen (2. Generation)) sowie optional das Kennwort für den Internetproxyserver für die Anmeldung bei Oracle Fusion Cloud EPM-Umgebungen zu verschlüsseln, und speichert es in einer Kennwortdatei.

Durch Verschlüsseln des Secrets können Serviceadministratoren ihre verschlüsselte Kennwortdatei mit Entwicklern teilen, die EPM Automate-Skripte schreiben, damit diese die Skripte ausführen können. Dadurch wird verhindert, dass das Serviceadministrator-Kennwort geteilt werden oder ein generisches, gemeinsames EPM Cloud-Konto erstellt werden muss, das speziell zum Ausführen von Skripten verwendet wird.

Das Verschlüsseln des Kennworts ist ein einmaliger Prozess.



Hinweis:

Informationen zum Verschlüsseln von Kennwörtern mit Sonderzeichen finden Sie unter [Sonderzeichen verarbeiten](#).

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise

Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Verwendung

`epmautomate encrypt PASSWORD|REFRESH_TOKEN KEY PASSWORD_FILE`
`[ClientID=CLIENT_ID] [ProxyServerPassword=PROXY_PASSWORD]`, wobei Folgendes gilt:

- `PASSWORD|REFRESH_TOKEN PASSWORD` ist das Kennwort oder das OAuth-Aktualisierungstoken, das Sie verschlüsseln möchten. Sie können für EPM Automate nicht die Zugangsdaten des Unternehmens verwenden.
- `KEY` ist der private Schlüssel, der zum Verschlüsseln des Kennworts verwendet wird.
- `PASSWORD_FILE` ist der Name und der Speicherort der Datei, in der das verschlüsselte Kennwort oder das Aktualisierungstoken gespeichert ist. Die Kennwortdatei muss die Erweiterung `EPW` aufweisen.
- `ClientID` (optional) ist die Client-ID, die während des OAuth 2.0-Setups erstellt wird. Dieser Wert muss bei der Verschlüsselung eines OAuth 2.0-Aktualisierungstokens angegeben werden. Geben Sie diesen Wert nicht bei der Verschlüsselung eines Kennworts an.
- `ProxyServerPassword` ist das Kennwort zum Authentifizieren des Benutzers beim HTTP-Proxyserver. Nur erforderlich, wenn die Authentifizierung auf dem Proxyserver für Ihr Netzwerk aktiviert ist.

Beispiele

- **Nur das EPM Cloud-Kennwort verschlüsseln:** `epmautomate encrypt P@ssword1 myKey C:\mySecuredir\password.epw`
- **Kennwort für EPM Cloud und für den Internetproxyserver verschlüsseln:** `epmautomate encrypt E@xample1 myKey C:\mySecuredir\password.epw ProxyServerPassword=Proxy_Pwd1`
- **Aktualisierungstoken und Client-ID verschlüsseln:** `epmautomate encrypt AAyyilyBAWD4...FVxkxfd8kjoJr6HJPA= myEncyprtion42Key C:\mySecuredir\oauthfile1.epw ClientID=6fdf2e72fd343430ABR22394C`

executeAggregationProcess

Startet den Aggregationsprozess. Dabei werden optional Statistiken für das Abfrage-Tracking verwendet, um die Performance von ASO-Cubes zu verbessern. Dies stellt einen wichtigen Schritt bei der Optimierung von ASO-Cubes dar.

Vor dem Ausführen dieses Befehls:

- Verwenden Sie den Befehl [enableQueryTracking](#), um Statistiken zum Datenabruf für die Optimierung der ASO-Aggregation zu erfassen.
- Gewähren Sie dem Geschäftsprozess ausreichend Zeit, um Muster beim Abruf von Benutzerdaten (Abfragen) zu erfassen, die wiederum zum Erstellen von Aggregatsichten verwendet werden können.

Gilt für

Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate executeAggregationProcess ASO_CUBE_NAME [useQueryData=true|false] [includeAlternateRollups=disable|enable] [growthSizeRatio=VALUE]`, wobei Folgendes gilt:

- `useQueryData`, um erfasste Abfragedaten zu verwenden, die mit dem Abfrage-Tracking erfasst wurden, um den am besten geeigneten Satz von Aggregatsichten auszuwählen. Der Standardwert ist `false`.
- `includeAlternateRollups`, um sekundäre Hierarchien (mit Verwendung der Standardebene) in den Prozess der Ansichtenauswahl einzuschließen. Die Standardeinstellung lautet `disable`.
- `growthSizeRatio` (optional) ist das Verhältnis für das maximale Cube-Wachstum, um die vom Server ausgewählten Ansichten zu aggregieren. Das Cube-Wachstum wird gestoppt, wenn das maximale Wachstum das angegebene Verhältnis erreicht. In der Standardeinstellung ist ein Cube-Wachstum ohne Begrenzung des Wachstumsverhältnisses zulässig.



Note:

Um standardmäßige Aggregatsichten zu erstellen, führen Sie diesen Befehl ohne optionale Parameter aus.

Beispiele

- Aggregatsichten basierend auf den erfassten Abfragedaten mit dem Befehl [enableQueryTracking](#) erstellen:

```
epmautomate executeAggregationProcess VISION_ASO useQueryData=true includeAlternateRollups=enable
```
- Standardmäßige Aggregatsichten erstellen:

```
epmautomate executeAggregationProcess Vis1ASO
```

executeBurstDefinition

Führt eine Bursting-Definition aus, mit der Artefakte, POVs und andere Einstellungen angegeben werden, die zum Ausführen von Berichten oder Büchern für mehrere Elemente einer einzelnen Dimension für eine Datenquelle erforderlich sind.

Gilt für

Narrative Reporting

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzern mit den Rollen Poweruser, Benutzer und Viewer muss über die ACL zusätzliche Sicherheit zugewiesen werden.

Verwendung

`epmAutomate executeBurstDefinition ARTIFACT_NAME`, wobei `ARTIFACT_NAME` dem Pfad und Namen der Bursting-Definition entspricht.

Beispiel

```
epmAutomate executeBurstDefinition "library/Reports/Example BurstDef1"
```

executeReportBurstingDefinition

Führt mit einer Bursting-Definition Bursting für einen einzelnen Bericht oder ein Buch für mehrere Elemente einer einzelnen Dimension aus und veröffentlicht eine PDF- oder statische (in Oracle Smart View for Office nicht aktualisierbare) Excel-Ausgabe für jedes Element.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmAutomate executeReportBurstingDefinition BURST_DEFINITION_NAME`
`[jobName=JOB_NAME]`. Dabei gilt:

- `BURST_DEFINITION_NAME` ist der Pfad und Name einer Bursting-Definition.
- `JOB_NAME` (optional) ist der Name des Jobs, der zum Ausführen der Bursting-Definition verwendet werden soll. Der Standardwert lautet `Execute Bursting Definition`.

Beispiel

```
epmAutomate executeReportBurstingDefinition /Library/MonthlySalesBurstDef
```

exportAccessControl

Exportiert den Benutzerdetailbericht, der Informationen zu den Benutzern mit vordefinierten Rollen in der Umgebung enthält und Attribute der einzelnen Benutzer (z.B. Name und E-Mail-Adresse) sowie Informationen zu ihrem Zugriff (z.B. Zuweisung zu Gruppen, Teams und Organisationen) in einer CSV- oder XLS-Datei auflistet.

Beispiel für einen Bericht:

	A	B	C	D	E	F	G	H	I	J	K	L
	Name	User Login	Status	Teams	Email	Role	Workflow Roles	Preparer	Reviewer	Organizations	Power User Filter	Last Login
1	John Doe	john.doe@example.com	Available	AP Preparers	john.doe@example.com	Administrator		Yes	Yes			
2	Jane Doe	jane.doe@example.com	Available	AR Preparers	jane.doe@example.com	Power User		No	No			
3				AP Reviewers								
4	ats_power_user4	ats_power_user4	Available	AR Reviewers	example1@example.com	User		No	No			
5	ats_user1	ats_user1	Available		example2@example.com	User		No	No			
6	ats_view_user1	ats_view_user1	Available		view.user@exaple.com	Viewer		No	No			

Sie können diesen Bericht mit dem Befehl [downloadFile](#) herunterladen.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmAutomate exportAccessControl REPORT_NAME [reportFormat=XLS|CSV]`, wobei Folgendes gilt:

- `REPORT_NAME` ist der Name der Exportdatei, die den Bericht enthält.
- `reportFormat(optional)` ist das Dateiformat. Gültige Werte sind "XLS" und "CSV" (Standard).

Beispiel

```
epmAutomate exportAccessControl aclreport.xls reportFormat=XLS
```

exportAppAudit

Exportiert Datensätze für den Datenaudit in eine ZIP-Datei, die Sie herunterladen und auf einem lokalen Computer archivieren können. Auditinformationen sind in der Umgebung bis zu 365 Tage lang verfügbar.

Das erste Zeichen in der Ausgabe-CSV-Datei ist das BOM-Zeichen (Byte Order Mark, Markierung für Bytereihenfolge) `\ufeff` gefolgt von einer verschlüsselten Anwendungs-ID in doppelten Anführungszeichen. Der CSV-Dateiheader steht nach der Anwendungs-ID.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate exportAppAudit EXPORT_FILE_NAME [userNames=USER_NAMES] [nDays=Number_of_Days] [excludeApplicationId=true|false]`, wobei Folgendes gilt:

- `EXPORT_FILE_NAME` ist der Name der ZIP-Datei, in der die exportierten Auditdaten gespeichert werden. Verwenden Sie den Befehl [downloadFile](#), um Dateien aus einer Umgebung herunterzuladen.
- `userNames` ist eine durch ein Komma getrennte optionale Liste von Benutzeranmeldungen. Wenn der Wert angegeben ist, werden nur die von diesen Benutzern erstellten Auditdaten exportiert. Geben Sie diesen Wert nicht an, wenn Sie die Auditdaten für alle Benutzer exportieren möchten.
- `nDays` gibt optional die Anzahl der Tage an, für die Auditdatensätze exportiert werden sollen. Der Standardwert ist sieben Tage. Mögliche Werte sind: `all` zum Exportieren aller verfügbaren Auditdaten der letzten 365 Tage, `1`, `2`, `7`, `30`, `60` und `180`.
- `excludeApplicationId` (optional) gibt an, ob die Anwendungs-ID in die Exportdatei geschrieben werden soll. Der Standardwert ist `false`.

 **Hinweis:**

Daten aus exportierten Dateien, die keine Anwendungs-ID enthalten, können nicht in Oracle Enterprise Performance Management Cloud-Umgebungen importiert werden.

Beispiele

- Auditdaten mit der Anwendungs-ID exportieren:

```
epmautomate exportAppAudit auditData userNames=johnDoe,jane.doe@example.com ndays=30
```
- Auditdaten ohne Anwendungs-ID exportieren:

```
epmautomate exportAppAudit auditData userNames=johnDoe,jane.doe@example.com ndays=30 excludeApplicationId=true
```

exportAppSecurity

Exportiert die Zuweisungen für den Zugriff auf Artefaktebene (ACLs) in eine CSV-Datei, die Sie herunterladen und lokal speichern können.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate exportAppSecurity EXPORT_FILE_NAME.CSV wobei EXPORT_FILE_NAME` der Name der Datei ist, in der die exportierten Sicherheitsdaten gespeichert werden. Diese Datei wird in der Outbox erstellt, über die Sie die Datei auf Ihren Computer herunterladen können.

Beispiel

```
epmautomate exportAppSecurity app_security.CSV
```

exportARApplicationProperties

Exportiert die Anwendungseinstellungen von Account Reconciliation (bezüglich Redwood-Design, Theme, E-Mail-Benachrichtigung und Geschäftsprozessname), das Hintergrundbild und das Bild für das Logo in eine JSON-Datei, sodass Sie sie in dieselbe oder eine andere Umgebung importieren können.

Dieser Befehl ist nützlich, wenn Sie eine Anwendung aus einer Produktionsumgebung in eine Testumgebung importieren möchten. Wenn Ihre Anwendungseinstellungen in der Produktionsumgebung und der Testumgebung unterschiedlich sind, können Sie sie aus der Testumgebung exportieren, bevor Sie die Anwendung aus der Produktionsumgebung importieren, und anschließend die Einstellungen in die Testumgebung importieren, um die ursprünglichen Einstellungen beizubehalten.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate exportARApplicationProperties FILE_NAME  
[Properties=PROPERTIES_TO_EXPORT]
```

- *FILE_NAME* ist der Name der JSON-Datei, in der die exportierten Eigenschaftswerte gespeichert werden. Sie können die Exportdatei mit dem Befehl [downloadFile](#) herunterladen. Mit dem Befehl [uploadFile](#) können Sie sie in die Zielumgebung hochladen und dann den Befehl [importARApplicationProperties](#) ausführen, um diese Einstellungen in der Zielumgebung wiederherzustellen.
- *Properties* (optional) ist eine durch Komma getrennte Liste der zu exportierenden Eigenschaften. Sie können einige oder alle der folgenden Eigenschaften exportieren. Wird diese Eigenschaft weggelassen, werden alle folgenden Eigenschaften exportiert:
 - *Theme*: exportiert das in der Umgebung verwendete Theme.
 - *EmailNotification*: exportiert die Einstellungen für E-Mail-Benachrichtigungen, die in der Umgebung definiert sind.
 - *DisplayBusinessProcessName*: exportiert die Einstellung, die angibt, ob der Geschäftsprozessname auf der Seite in der Umgebung angezeigt werden soll.
 - *RedwoodExperience*: exportiert die Einstellungen für das Redwood-Design der Umgebung.
 - *BackgroundImage*: exportiert das in der Umgebung verwendete Hintergrundbild.
 - *LogoImage*: exportiert das in der Umgebung verwendete Bild für das Logo.

Beispiel

Nur die E-Mail-Benachrichtigung, die Einstellungen für das Redwood-Design und das Bild für das Logo aus einer Umgebung exportieren:

```
epmautomate exportARApplicationProperties myProp.JSON
Properties=EmailNotification,RedwoodExperience,LogoImage
```

exportBackgroundImage

Exportiert das Hintergrundbild, das in einer Account Reconciliation-Umgebung verwendet wird, in eine JPG-Datei, sodass Sie es in eine andere Umgebung importieren können.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate exportBackgroundImage IMAGE_NAME.jpg`, wobei `IMAGE_NAME` der Name der Datei für das Hintergrundbild ist.

Sie können die Bilddatei mit dem Befehl [downloadFile](#) herunterladen. Mit dem Befehl [uploadFile](#) können Sie sie in die Zielumgebung hochladen und dann den Befehl [importBackgroundImage](#) ausführen, um sie zu importieren.

Beispiel

```
epmautomate exportBackgroundImage corpImage.jpg
```

exportCellLevelSecurity

Exportiert Sicherheitseinstellungen auf Zellenebene aus dem Geschäftsprozess in eine ZIP-Datei, die Sie mit dem Befehl [downloadFile](#) auf einen lokalen Computer herunterladen können.

Gilt für

Planning, Planning Modules, FreeForm, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate exportCellLevelSecurity FILE_NAME.ZIP [names=SECURITY_RECORD_NAMES]`, wobei Folgendes gilt:

- `FILE_NAME` ist der Name für die ZIP-Datei, die zum Speichern der Excel-Datei mit den Informationen zur Sicherheit auf Zellenebene erstellt werden soll.
- `names` (optional) gibt eine durch Komma getrennte Liste der Sicherheitsdefinitionen auf Zellenebene in der Anwendung an. Wenn diese Option nicht bereitgestellt wird, werden alle Sicherheitsdefinitionen auf Zellenebene in der Anwendung exportiert.

Beispiele

- **So exportieren Sie spezifische Sicherheitsdefinitionen auf Zellenebene:**

```
epmautomate exportCellLevelSecurity ExportCLSDRecordsFile.zip  
names=CLSDAccountPeriod,CLSDEntityPeriod,CLSDProductPeriod
```

- **So exportieren Sie alle Sicherheitsdefinitionen auf Zellebene:**
epmautomate exportCellLevelSecurity ExportCLSDRecordsFile.zip

exportConsolidationJournals

Exportiert Konsolidierungsjournale mit einem in Financial Consolidation and Close definierten Job.

Gilt für

Financial Consolidation and Close

Erforderliche Rollen

Serviceadministrator

Verwendung

epmautomate exportConsolidationJournals jobName [fileName=FILE_NAME], wobei Folgendes gilt

- jobName ist der Name eines Journalexportjobs, der in Financial Consolidation and Close erstellt wurde.
- fileName (optional) ist der Name einer JLF-Datei, in die die Journaleinträge exportiert werden sollen. Laden Sie diese Datei mit dem Befehl [downloadFile](#) auf einen lokalen Computer herunter.

Beispiel

```
epmautomate exportConsolidationJournals "JEXPORT1"  
fileName=Export_Test.jlf
```

exportData

Die Anwendungsdaten werden mit den Einstellungen, die in einem Job vom Typ `export data` (Daten exportieren) festgelegt sind, einschließlich Dateinamen, in eine ZIP-Datei exportiert.

Die exportierte Datendatei wird am standardmäßigen Downloadspeicherort gespeichert, von dem er auf Ihren Computer heruntergeladen werden kann. Verwenden Sie den Explorer für die Inbox/Outbox, um Details zur exportierten Datei anzuzeigen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate exportData JOB_NAME [FILE_NAME]`, wobei `JOB_NAME` der Name eines in der Anwendung definierten Jobs ist. `FILE_NAME` ist der Name der ZIP-Datei (optional), in die die Daten exportiert werden sollen.

Beispiel

```
epmautomate exportData dailydataexport dailyData.zip
```

exportDataManagement

Exportiert Datenmanagement-Datensätze aus einer Umgebung in eine ZIP-Datei.

Mit diesem Befehl wird ein vollständiger Satz aus Setup- und Staging-Tabellendaten, einschließlich der ID-Spalten, in eine ZIP-Datei exportiert, damit die Daten importiert werden können, ohne dass die referenzielle Integrität verloren geht.

Die exportierte Datei, z.B. `dataFile.zip`, wird in der Outbox gespeichert. Sie können die exportierte Datei mit dem Befehl [downloadFile](#) herunterladen, z.B. `epmAutomate downloadFile outbox/dataFile.zip`. Sie können diese ZIP-Datei verwenden, um die Daten mit dem Befehl [importDataManagement](#) zu importieren.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate exportDataManagement FILE_NAME.zip`, wobei `FILE_NAME` der Name der ZIP-Datei ist, in die die Daten exportiert werden sollen.

Beispiel

```
epmautomate exportDataManagement dataFile.zip
```

exportDimension

Exportiert eine Dimension aus einer Oracle Enterprise Data Management Cloud-Anwendung in eine Datei in der Staging Area oder optional in eine in der Verbindung definierte Zielumgebung.

Gilt für

Oracle Enterprise Data Management Cloud

Erforderliche Rollen

Serviceadministrator, Benutzer (mit der Berechtigung "Datenmanager")

Verwendung

`epmautomate exportDimension APPLICATION DIMENSION FILE_NAME`
[`connection=NAME`] wobei Folgendes gilt:

- `APPLICATION` ist der Name einer Oracle Enterprise Data Management Cloud-Anwendung.
- `DIMENSION` ist der Name einer Anwendungsdimension.
- `FILE_NAME` ist der Name der Datei (CSV für den Export in eine Datei oder ZIP für den Export in Oracle Financials Cloud), in der die exportierten Daten gespeichert werden sollen. Wenn der Wert für den Parameter `connection` nicht festgelegt ist, wird diese Datei in der Staging Area erstellt. Sie können sie mit dem Befehl [downloadFile](#) auf einen lokalen Computer herunterladen oder mit dem Befehl [copyFileFromInstance](#) in eine andere Oracle Enterprise Performance Management Cloud-Umgebung kopieren.
- `connection=NAME` (optional) gibt einen Verbindungsnamen (Speicherort einer Instanz) an, der in Oracle Enterprise Data Management Cloud definiert ist. Wenn der Wert angegeben ist, wird die exportierte Datei in die Zielumgebung hochgeladen (Inbox für EPM Cloud und Standardspeicherort zum Hochladen für Oracle Financials Cloud).

Hinweis:

Die in der Verbindungsdefinition angegebenen Zugangsdaten müssen über die Zugriffsberechtigungen zum Schreiben in der Zielumgebung verfügen.

Beispiele

- So exportieren Sie in die Staging Area von Oracle Enterprise Data Management Cloud: `epmautomate exportDimension USOperations Entity EntityData.CSV`
- In Oracle Financials Cloud exportieren und hochladen: `epmautomate exportDimension USOperations Entity EntityData.zip Connection=ora_fusion_gl`
- In die EPM Cloud-Zielinbox exportieren und hochladen: `epmautomate exportDimension USOperations Entity EntityData.CSV Connection=EPM_cloud_pln`

exportDimensionMapping

Exportiert Zuordnungsregeln einer bestimmten Oracle Enterprise Data Management Cloud-Dimension für einen Speicherort, um eine Zuordnungsregeldatei zu erstellen, und lädt (optional) die exportierte Datei in die Datenmanagement-Inbox einer anderen Oracle Enterprise Performance Management Cloud-Umgebung hoch.

Gilt für

Oracle Enterprise Data Management Cloud

Erforderliche Rollen

Serviceadministrator, Benutzer (mit der Berechtigung "Datenmanager")

Verwendung

`epmautomate exportDimensionMapping APPLICATION DIMENSION LOCATION FILE_NAME [connection=NAME]` wobei Folgendes gilt:

- `APPLICATION` ist der Name einer Oracle Enterprise Data Management Cloud-Anwendung.
- `DIMENSION` ist der Name einer Anwendungsdimension.
- `LOCATION` ist der spezifische Speicherort, für den Zuordnungsregeln exportiert werden sollen.
- `FILE_NAME` ist der Name der CSV-Datei, in der die exportierten Zuordnungen gespeichert werden sollen. Diese Datei wird in der Staging Area erstellt, wenn der Parameter `connection` nicht festgelegt ist. Mit dem Befehl [downloadFile](#) können Sie sie auf einen lokalen Computer herunterladen. Alternativ können Sie die Datei mit dem Befehl [copyFileFromInstance](#) in eine andere EPM Cloud-Umgebung kopieren.
- `connection=NAME` (optional) gibt einen Verbindungsnamen (Speicherort einer Instanz) an, der in Oracle Enterprise Data Management Cloud definiert ist. Wenn er angegeben ist, lädt EPM Automate die exportierte Datei in den Standardspeicherort für Uploads der Zielumgebung hoch.

Hinweis:

Die in der Verbindung angegebenen Zugangsdaten müssen über die Zugriffsberechtigungen zum Schreiben in der Zielumgebung verfügen.

Beispiele

- So exportieren Sie in die Staging Area: `epmautomate exportDimensionMapping USOperations Entity Loc1 Loc1Mappings.CSV`
- In die Inbox der EPM Cloud-Zielumgebung exportieren und hochladen: `epmautomate exportDimensionMapping USOperations Entity Loc1 Loc1Mappings.CSV Connection=EPM_cloud_pln`

exportEJournals

Exportiert Enterprise-Journale, die für die Aktivierung bereit sind, aus Financial Consolidation and Close in eine ZIP-Datei. Diese Datei kann dann verwendet werden, um Journaldaten in ein ERP-System zu importieren.

Nach dem Export der Journale in eine Exportdatei wird mit diesem Befehl der Aktivierungsstatus für jedes exportierte Journal von Bereit für Aktivierung in Aktivierung in Verarbeitung aktualisiert.

Gilt für

Financial Consolidation and Close

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate exportEJJournals FILE_NAME.zip [year=YEAR [period=PERIOD]]`,
wobei Folgendes gilt:

- *FILE_NAME* gibt eine ZIP-Datei an, in die CSV-Dateien des Journalexports archiviert werden sollen. Der Befehl generiert eine CSV-Datei (mit dem Namensformat `YEAR_PERIOD_JOURNALID_YYYYDDMMHHMMSS.csv`) für jedes Journal und erstellt daraus eine ZIP-Datei.
- *YEAR* (optional) ist das Jahr der Datenerfassung, für das die Journaldaten exportiert werden sollen. Wenn nichts angegeben ist, werden die Daten aus allen Jahren exportiert.
- *PERIOD* (optional) ist die Periode der Datenerfassung, für die die Journaldaten exportiert werden sollen. Kann nur festgelegt werden, wenn ein Jahr für die Datenerfassung angegeben wurde. Wenn kein Wert angegeben ist, werden die Daten aus allen Perioden exportiert.

Note:

Wenn *YEAR* und *PERIOD* nicht angegeben sind, werden mit diesem Befehl alle Journale mit dem Aktivierungsstatus `Bereit für Aktivierung` für alle Jahre und Perioden exportiert.

Beispiele

- Journaldaten für alle Jahre und Perioden exportieren:
`epmautomate exportEJJournals Journal_Export.zip`
- Journaldaten für ein bestimmtes Jahr exportieren:
`epmautomate exportEJJournals Journal_Export.zip year=2020`
- Journaldaten für eine bestimmte Kombination aus Jahr und Periode exportieren:
`epmautomate exportEJJournals Journal_Export.zip year=2021 period=March`

exportEssbaseData

Exportiert Daten aus einem Anwendungs-Cube (Oracle Essbase-Cube) in ein Archiv. Sie können entweder nur die Daten der Ebene 0 (ASO- und BSO-Cubes) oder alle Daten in einem Cube (BSO-Cubes) exportieren.

Verwenden Sie das exportierte Archiv, um Essbase-Daten auf Muster zu analysieren, zum Beispiel, um die Performance zu verbessern.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate exportEssbaseData CUBE_NAME FILE_NAME [level=0|All]`, wobei:

- `CUBE_NAME` den Cube angibt, aus dem Daten exportiert werden sollen.
- `FILE_NAME` der Name einer ZIP-Datei ist, die die exportierten Daten enthalten wird. Sie können dieses Archiv herunterladen, indem Sie den Befehl [downloadFile](#) ausführen.
- `level` optional die Ebene der zu exportierenden Daten angibt. Der Standardwert ist 0.
 - **ASO-Cubes:** Geben Sie 0 an, um Daten der Ebene 0 zu exportieren. Sie können die Option `Alle` nicht verwenden.
 - **BSO-Cubes:** Geben Sie 0 an, um Daten der Ebene 0 zu exportieren, oder geben Sie `Alle` an, um alle Daten zu exportieren.

Beispiele

- Alle Daten aus einem BSO-Cube exportieren:
`epmautomate exportEssbaseData Report1 Report1_all_data.zip level=All`
- Daten der Ebene 0 aus einem Cube exportieren:
`epmautomate exportEssbaseData Plan1 Plan1_lvl0_data.zip`

exportJobConsole

Exportiert die Jobkonsolendatensätze in eine CSV-Datei und erstellt eine Export-ZIP-Datei.

Das erste Zeichen in der Ausgabe-CSV-Datei ist das BOM-Zeichen (Byte Order Mark, Markierung für Bytereihenfolge) `\ufeff` gefolgt von einer verschlüsselten Anwendungs-ID in doppelten Anführungszeichen. Der CSV-Dateiheader steht nach der Anwendungs-ID.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate exportJobConsole FILE_NAME.zip [nDays=NUMBER_OF_DAYS]`
`[jobtypes=JOB_TYPE] [jobStatusCodes=STATUS_CODE] [exportErrorDetails=true|false]`
`[excludeApplicationId=true|false]`, wobei Folgendes gilt:

- `FILE_NAME` ist der Name der ZIP-Datei, in der die exportierten Jobkonsolendatensätze gespeichert werden. Verwenden Sie den Befehl [downloadFile](#), um diese Datei aus einer Umgebung herunterzuladen.

- `nDays` (optional) gibt die Anzahl der Tage an, für die Jobkonsolendatensätze exportiert werden sollen. Mögliche Werte sind: `all` (in Kleinbuchstaben) zum Exportieren aller verfügbaren Jobkonsolendatensätze, 1, 2, 7, 30 und 60. Der Standardwert ist 7.
- `jobTypes` (optional) ist eine durch Komma getrennte Liste mit Jobcodes, für die Konsolendatensätze exportiert werden müssen. Die Standardeinstellung lautet `Rules`. Gültige Werte sind:
 - `all` (in Kleinbuchstaben)
 - `RULES`
 - `RULESET`
 - `CLEAR_CELL_DETAILS`
 - `COPY_DATA`
 - `INVALID_INTERSECTION_RPT`
 - `COPY_VERSIONS`
 - `CONTENT_UPGRADE`
 - `PLAN_TYPE_MAP`
 - `IMPORT_DATA`
 - `EXPORT_DATA`
 - `EXPORT_METADATA`
 - `IMPORT_METADATA`
 - `CUBE_REFRESH`
 - `CLEAR_CUBE`
 - `ADMIN_MODE`
 - `COMPACT_CUBE`
 - `RESTRUCTURE_CUBE`
 - `MERGE_DATA_SLICES`
 - `OPTIMIZE_AGGREGATION`
 - `SECURITY_IMPORT`
 - `SECURITY_EXPORT`
 - `AUDIT_EXPORT`
 - `JOBCONSOLE_EXPORT`
 - `SORT_MEMBERS`
 - `SMART_PUSH`
 - `IMPORT_EXCHANGE_RATES`
- `jobStatusCodes` (optional) ist eine durch Komma getrennte Liste mit Jobstatuscodes, für die Datensätze exportiert werden sollen. Die Standardeinstellung lautet 2 (Erfolgreich abgeschlossen). Mögliche Werte sind:
 - `all` (in Kleinbuchstaben) für alle Jobs mit beliebigem Status

- 1 - Verarbeitung
 - 2 - Erfolgreich abgeschlossen
 - 3 - Abbruch mit Fehlern
 - 4 - Mit unbekanntem Status abgeschlossen
 - 5 - Mit dem Status der Schwellenwertverletzung abgeschlossen
 - 6 - Abbruch anstehend
 - 7 - Abgebrochen
 - 8 - Mit Fehlern abgeschlossen
 - 9 - Mit Warnungen abgeschlossen
- *exportErrorDetails* (optional) exportiert die Details von Jobs, die nicht erfolgreich waren oder Fehler in Logdateien gemeldet haben, wenn dieser Wert auf `true` gesetzt ist. Diese Fehlerlogdatei ist Bestandteil der Ausgabe-ZIP-Datei. Der Standardwert ist `false`. Wenn dieser Wert auf `true` gesetzt ist, werden die Statusdetails für Jobs mit den folgenden Statusangaben exportiert.
 - Mit Fehlern fehlgeschlagen
 - Mit unbekanntem Status abgeschlossen
 - Status "Abgeschlossen mit Schwellenwertverletzungen"
 - Mit Fehlern abgeschlossen
 - Mit Warnungen abgeschlossen
 - *excludeApplicationId* (optional) gibt an, ob die Anwendungs-ID in die Exportdatei geschrieben werden soll. Der Standardwert ist `false`.

 **Note:**

Daten aus exportierten Dateien, die keine Anwendungs-ID enthalten, können nicht in Oracle Enterprise Performance Management Cloud-Umgebungen importiert werden.

Beispiele

- Alle verfügbaren Jobkonsolendatensätze exportieren:
`epmautomate exportJobConsole jobs.zip nDays=all jobTypes=all jobStatusCodes=all`
- Alle verfügbaren Jobkonsolendatensätze für Regeln exportieren:
`epmautomate exportJobConsole jobs.zip nDays=all jobStatusCodes=all`
- Alle verfügbaren Jobkonsolendatensätze für Regeln ohne Anwendungs-ID exportieren:
`epmautomate exportJobConsole jobs.zip nDays=all jobStatusCodes=all excludeApplicationId=true`
- Nur die Datensätze für Regeljobs exportieren, die in den letzten 14 Tagen erfolgreich abgeschlossen wurden:
`epmautomate exportJobConsole jobs.zip nDays=14`

- Konsolendatensätze und Fehler beim Import von Metadaten exportieren sowie Cube-Jobs entfernen, die beim Ausführen in den letzten sieben Tagen mit Fehlern abgebrochen oder abgeschlossen wurden:

```
epmautomate exportJobConsole jobs.zip
jobtypes=IMPORT_METADATA,CLEAR_CUBE jobStatusCodes=3,8
exportErrorDetails=true
```

exportLibraryArtifact

Exportiert Narrative Reporting-Bibliotheksartefakte. Optional und nur für Berichtsartefakte: Dieser Befehl kann den Export in eine LCM-Datei konvertieren, die Sie dann in Financial Consolidation and Close, Planning, Planning Modules oder Tax Reporting importieren können.

Verwenden Sie nach Abschluss des Exports den Befehl [downloadFile](#), um die Export- und Fehlerdateien auf einen lokalen Computer herunterzuladen.

Gilt für

Narrative Reporting

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
 Benutzern mit den Rollen Poweruser, Benutzer und Viewer muss über die ACL zusätzliche Sicherheit zugewiesen werden.

Verwendung

```
epmautomate exportLibraryArtifact ARTIFACT_PATH EXPORT_FILE
[exportFormat=Native|File|LCM] [applicationName=APP_NAME]
[errorFile=ERROR_FILE.txt], wobei Folgendes gilt:
```

- *ARTIFACT_PATH* ist der Speicherort des Artefakts in der Narrative Reporting-Bibliothek.
- *EXPORT_FILE* ist ein eindeutiger Name für die Datei, in die Artefakte exportiert werden sollen.
- *exportFormat* (optional) entspricht einem der folgenden Formate:
 - *Native* exportiert Artefakte als ZIP-Datei, die mit anderen Narrative Reporting-Umgebungen verwendet werden kann. Das ist der Standardwert.
 - *File* exportiert Dateien im ursprünglichen binären Format (PDF, DOCX, ZIP, JPEG usw.), in dem sie in Narrative Reporting verfügbar sind. Dieser Parameter kann nur für den Export binärer Dateien verwendet werden. Er darf nicht mit Berichtsartefakten verwendet werden.
 - *LCM* konvertiert Berichte in das von Migration verwendete Format und exportiert sie in eine ZIP-Datei, die dann in Umgebungen von Financial Consolidation and Close, Planning, Planning Modules oder Tax Reporting importiert werden können.
- *applicationName* (optional) ist der Name der Zielanwendung, in die Sie die Berichte importieren möchten. Dieser Wert ist nur erforderlich, wenn Sie *LCM* als Wert für den Parameter *exportFormat* verwenden.

- *errorFile* (optional) ist ein eindeutiger Name für die Textdatei, in der Exportfehler gespeichert werden.

Beispiele

- Berichte im nativen Format exportieren, damit sie in eine andere Narrative Reporting-Umgebung importiert werden können:

```
epmautomate exportLibraryArtifact "Library/Samples/Sample Report 1"
exp_SampleReport1.doc errorFile=export_errors.txt
```
- Tabellen im ursprünglichen binären Format exportieren:

```
epmautomate exportLibraryArtifact "Library/Spreadsheets/Sheet1.xlsx"
exp_Sheet1.xlsx exportFormat=File errorFile=export_errors.txt
```
- Berichte exportieren und für den Import in Financial Consolidation and Close, Planning, Planning Modules oder Tax Reporting formatieren:

```
epmautomate exportLibraryArtifact "Library/Samples/Sample Report 1"
exp_SampleReport1.zip exportFormat=LCM applicationName=Vision
errorFile=report_exp_errors.txt
```

exportLibraryDocument

Exportiert in der Berichtsbibliothek verfügbare Dokumente in eine Datei.

Sie können die exportierte Datei mit dem Befehl [downloadFile](#) auf einen lokalen Computer herunterladen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate exportLibraryDocument ARTIFACT_PATH [jobName=JOB_NAME]
[exportFile=FILE_NAME] [exportFormat=file|zip] [errorFile=FILE_NAME.log]
[overwrite=true|false], wobei Folgendes gilt:
```

- *ARTIFACT_PATH* ist der Speicherort des Dokuments in der Berichtsbibliothek.
- *jobName* (optional) ist der Name des Bibliotheksartefakt-Exportjobs, der für den Export des Dokuments verwendet wird. Der Standardjobname lautet `Copy Artifact From Library`.
- *EXPORT_FILE* ist ein eindeutiger Name für die Datei, in die das Dokument exportiert werden soll. Wenn Sie diesen Wert nicht angeben, wird die Exportdatei mit dem Namen des Dokuments in der Bibliothek erstellt.
- *exportFormat* (optional) entspricht einem der folgenden Formate:
 - Mit dem Wert `file` wird das Dokument im ursprünglichen binären Format (PDF, DOCX, ZIP, JPEG usw.) exportiert, in dem es in der Bibliothek verfügbar ist. Das ist der Standardwert.

 **Note:**

Im Update 24.02 funktioniert diese Option nicht.

- Mit dem Wert `zip` wird eine ZIP-Datei exportiert, die das Dokument im ursprünglichen binären Format enthält. Dies ist die einzige Option, die in diesem Update funktioniert.
- `errorFile` (optional) ist ein eindeutiger Name für die Datei, in der Exportfehler gespeichert werden. Wenn Sie diesen Wert nicht angeben, wird keine Fehlerdatei erstellt.
- `overwrite` (optional) steuert, ob eine Datei mit demselben Namen, die sich derzeit im Standardspeicherort für Downloads befindet, überschrieben werden soll. Der Standardwert ist `false`, d.h., der Befehl ist nicht erfolgreich, wenn eine Datei mit demselben Namen in der Outbox vorhanden ist.

Beispiel

```
epmautomate exportLibraryDocument Library/folder1/WeeklySales.html
jobName="Copy Weekly Sales" exportFile=WeeklySales.zip
errorFile=WeeklySalesError.log overWrite=true exportFormat=zip
```

exportLogoImage

Exportiert das Unternehmenslogo, das in einem Account Reconciliation-Geschäftsprozess verwendet wird, in eine JPG-Datei, sodass Sie es in eine andere Umgebung importieren können.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate exportLogoImage IMAGE_NAME.jpg`, wobei `IMAGE_NAME` der Name der Bilddatei für das Logo ist.

Sie können die exportierte Datei mit dem Logo mit dem Befehl [downloadFile](#) herunterladen. Mit dem Befehl [uploadFile](#) können Sie sie in die Zielumgebung hochladen. Führen Sie dann den Befehl [importLogoImage](#) aus.

Beispiel

```
epmautomate exportLogoImage corpLogo.jpg
```

exportMapping

Exportiert Zuordnungsregeln einer bestimmten Dimension oder eines bestimmten Speicherortes, um eine Datei für Zuordnungsregeln zu erstellen. Sie müssen den Dateinamen und einen Speicherort in der Inbox angeben (z.B. `inbox/`

`exportedAccountMap.txt` oder `inbox/france sales/exportedAccountMap.txt`), um Zuordnungen zu exportieren.

Verwenden Sie den Befehl [downloadFile](#), um die exportierte Zuordnungsdatei auf einen lokalen Computer herunterzuladen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate exportMapping DIMENSION_NAME|ALL FILE_NAME LOCATION` wobei Folgendes gilt:

- `DIMENSION_NAME|ALL` ist die Quelldimension, aus der Zuordnungen zu exportieren sind. Geben Sie den Namen der Dimension an, aus der Zuordnungen zu exportieren sind, oder geben Sie `ALL` an, um Zuordnungen aus allen Dimensionen eines Speicherortes zu exportieren.
- `FILE_NAME` ist ein eindeutiger Name für die Zuordnungsdatei und ein Speicherort innerhalb der Outbox.
- `LOCATION` ist der Speicherort des Datenmanagements, für den Zuordnungsregeln exportiert werden sollen.

Beispiele

- `epmautomate exportMapping Account inbox/exportedAccountMap.txt "France Sales"`
- `epmautomate exportMapping ALL "inbox/france sales/exportedAccountMap.txt" "France Sales"`

exportMetadata

Metadaten werden mit den Einstellungen, die in einem Job vom Typ `export metadata` (Metadaten exportieren) festgelegt sind, in eine Datei exportiert. Die Datei mit den exportierten Daten wird am standardmäßigen Downloadspeicherort gespeichert, von dem aus er auf einen lokalen Computer heruntergeladen werden kann.

Optional können Sie einen Dateinamen für die exportierten Daten angeben, der den Standarddateinamen (den Namen des Jobs, mit dem die Metadaten exportiert werden) überschreibt. Metadaten können nur als ZIP-Datei exportiert werden.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate exportMetadata JOB_NAME [FILE_NAME]` wobei *JOB_NAME* der Name eines in der Anwendung definierten Jobs und *FILE_NAME* der Name der ZIP-Datei ist, in die Metadaten exportiert werden sollen.

Verwenden Sie den Befehl [downloadFile](#), um diese Datei auf einen lokalen Server herunterzuladen.

Beispiel

```
epmautomate exportMetadata dailyAccountexport Accountexport.ZIP
```

exportOwnershipData

Exportiert Anteilsdaten aus einer Entity in eine durch Komma getrennte CSV-Datei.

Standardanteilsdaten, die von Financial Consolidation and Close befüllt wurden, sind nicht in der Exportdatei enthalten. Nur die Daten, die Benutzer zum Überschreiben der Standardeinstellungen eingegeben haben, sind in der Exportdatei enthalten.

Gilt für

Financial Consolidation and Close und Tax Reporting.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer

Verwendung

`epmautomate exportOwnershipData Entity Scenario Year Period FILE_NAME`
wobei:

- *Entity* der Name der Entity ist, aus der Daten exportiert werden sollen.
- *Scenario* das Szenario ist, aus dem Daten exportiert werden sollen.
- *Year* das Jahr ist, aus dem Daten exportiert werden sollen.
- *Period* die Periode des Jahres ist, aus der Daten exportiert werden sollen.
- *FILE_NAME* der Name einer CSV-Datei ist, in die die Daten exportiert werden sollen. Verwenden Sie den Befehl [downloadFile](#), um diese Datei auf einen lokalen Server herunterzuladen.

Beispiel

```
epmautomate exportOwnershipData FCCS_TotalActual FY18 Dec exportfile.csv
```

exportQueryResults

Führt eine Abfrage aus, die in einer Anwendung definiert ist, und exportiert die Ergebnisse in eine Textdatei.

Die Datei mit den Abfrageergebnissen wird in "profitoutbox" gespeichert. Sie können sie mit dem Befehl [downloadFile](#) oder über den Dateixplorer von Profitability and Cost Management herunterladen.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Verwendung

```
epmautomate exportQueryResults APPLICATION_NAME fileName=FILE_NAME  
[fileOutputOptions=ZIP_ONLY|ZIP_AND_TEXT|TEXT_ONLY] [queryName=QUERY_NAME]  
[exportOnlyLevel0Flg=true|false] [roundingPrecision=2] [dataFormat=NATIVE|  
COLUMNAR] [memberFilters=JSON_FILTER] [includeHeader=true|false]  
[delimiter="DELIMITER"] [keepDuplicateMemberFormat=true|false], wobei Folgendes  
gilt:
```

- `APPLICATION_NAME` ist der Name der Profitability and Cost Management-Anwendung, für die Sie die Abfrage ausführen möchten.
- `fileName` ist der Name der Datei, in der die Abfrageergebnisse gespeichert werden. Dieser Parameterwert ist erforderlich, wenn der Parameterwert `queryName` nicht angegeben ist. Er ist optional, wenn der Parameterwert `queryName` angegeben ist. In diesem Fall wird der Abfragenname als Name für die Datei mit Abfrageergebnissen verwendet.
Das von Ihnen festgelegte Datenformat bestimmt das Format der Ausgabedatei. Wenn Sie `dataFormat=NATIVE` (Standard) verwenden, wird beim Exportprozess eine Textdatei erstellt. Wenn Sie `dataFormat=COLUMNAR` verwenden, werden beim Exportprozess mehrere der Reihe nach nummerierte Textdateien erstellt und in eine ZIP-Datei komprimiert.
- `fileOutputOptions` (optional) gibt das Ausgabeformat der Datei mit Abfrageergebnissen an. Die Standardeinstellung lautet `ZIP_ONLY`. Damit wird `fileName.ZIP` oder `queryName.ZIP` erstellt, und zwar abhängig davon, ob ein Wert für den Parameter `fileName` angegeben ist. Weitere Optionen sind `TEXT_ONLY` zum Erstellen der Ausgabedatei als Textdatei und `ZIP_AND_TEXT` zum Generieren einer Textdatei und einer ZIP-Datei.
- `queryName` ein optionaler Parameter ist, der eine in der Anwendung definierte Abfrage angibt. Abfragenamen, die Leerzeichen enthalten, müssen in doppelte Anführungszeichen gesetzt werden.
Geben Sie keinen Abfragenamen an, wenn Sie alle Oracle Essbase-Daten exportieren möchten, die zur Anwendung gehören.

Die folgenden Bedingungen können dazu führen, dass dieser Befehl eine leere Datendatei erstellt:

- Eine unzureichend erstellte Abfrage, die keine Daten abrufen.
- Eine Abfrage, die zu viele Daten generiert. Ziehen Sie in diesem Szenario in Betracht, den Bereich der Abfrage zu verkleinern, damit weniger Daten abgerufen werden, oder die Abfrage in kleinere Abfragen aufzuteilen.
Informationen hierzu finden Sie unter Oracle Profitability and Cost Management Cloud-Abfragen verwalten in der Dokumentation *Profitability and Cost Management verwalten*.

- `exportOnlyLevel0Flg` (optional) gibt an, ob die Abfrage nur Daten der Ebene 0 abrufen soll. Geben Sie diesen Parameterwert in Kleinbuchstaben an. Dieser Parameter wird ignoriert, wenn Sie alle Anwendungsdaten exportieren, ohne den Abfragenamen anzugeben.
- `roundingPrecision` (optional) gibt die Anzahl der Dezimalstellen (Rundungsgenauigkeit) an, die beim Exportieren von Abfrageergebnissen verwendet werden sollen. Nur anwendbar, wenn `queryName` angegeben ist. Der Standardwert ist 2.
- `dataFormat` (optional) gibt das Ausgabeformat an. Gültige Werte sind:
 - `NATIVE`, wobei das Abfrageergebnis als Daten im nativen Essbase-Format gepflegt wird. Das ist der Standardwert.
 - `COLUMNAR`, wobei Daten im nativen Essbase-Format konvertiert und in Spalten angeordnet werden, um sie leichter interpretieren und in andere Anwendungen importieren zu können. Mit dieser Option werden alle Essbase-Daten exportiert. Der Parameterwert `queryName` wird ignoriert. Sie können die Daten filtern, indem Sie den Parameterwert `memberFilters` festlegen.

 **Hinweis:**

Der Befehl berücksichtigt die folgenden optionalen Parameter nur, wenn für `dataFormat` `COLUMNAR` angegeben ist.

- `memberFilters` (optional) akzeptiert eine mit JSON formatierte Zeichenfolge, um nach Dimension und nach Elementen der Ebene 0 zu filtern. Beispiel:
`{"Dim1\":[\"Mem1\"],\"Dim2\":[\"Mem21\",\"Mem22\"]}`
- `includeHeader` (optional) fügt Dimensionsnamen als Spaltenheader hinzu. Setzen Sie diesen Wert auf `false`, um Spaltenheader auszuschließen. Der Standardwert ist `true`.
- `delimiter` (optional) gibt das Trennzeichen an, das verwendet werden muss, um Dimensionselemente in der Datei mit Abfrageergebnissen zu trennen. Trennzeichen müssen in doppelte Anführungszeichen gesetzt werden. Der Standardwert ist das Leerzeichen (" ").
- `keepDuplicateMemberFormat` (optional) gibt an, ob das Elementformat im Essbase-Format für doppelte Elemente gedruckt werden soll, z.B. `[Account]@[Account]`. Setzen Sie diesen Wert auf `false`, um nur den Elementnamen zu drucken. Der Standardwert ist `true`.

Beispiele

- Alle Anwendungsdaten exportieren:
`epmautomate exportQueryResults BksML12 fileName="BksML12_MyQuery1.txt" fileOutputOptions=TEXT_ONLY`
- Ergebnisse einer bestimmten Abfrage exportieren:
`epmautomate exportQueryResults BksML12 queryName="My Product Query" roundingPrecision=3`
- Daten der Ebene 0 im `NATIVE`-Datenformat exportieren:

```
epmautomate exportQueryResults BksML30 fileName="BksML30_ExportLevel0-Data"
fileOutputOptions=ZIP_AND_TEXT exportOnlyLevel0Flg=true
```

- Daten der Ebene 0 im COLUMNAR-Datenformat mit einer einzelnen Dimension und einem einzelnen Elementfilter exportieren:

```
epmautomate exportQueryResults BksML30 fileName="BksML30_Level0-Data"
dataFormat="COLUMNAR" memberFilters="{\"Period\": [\"December\"]}"
includeHeader="true" delimiter="," roundingPrecision="3"
```

- Daten der Ebene 0 im COLUMNAR-Datenformat mit einer einzelnen Dimension und mehreren Elementfiltern exportieren:

```
epmautomate exportQueryResults BksML30 fileName="BksML30_Level0-Data"
dataFormat="COLUMNAR" memberFilters="{\"Period\": [\"November\", \"December\"]}"
includeHeader="true" delimiter="," roundingPrecision="3"
```

- Daten der Ebene 0 im COLUMNAR-Datenformat mit mehreren Dimensionen und mehreren Elementfiltern exportieren:

```
epmautomate exportQueryResults BksML30 fileName="BksML30_Level0-Data"
dataFormat="COLUMNAR" memberFilters="{\"Year\": [\"2016\"], \"Period\": [\"November\", \"December\"]}"
includeHeader="true" delimiter="," roundingPrecision="3"
```

exportSnapshot

Wiederholt einen zuvor ausgeführten Exportvorgang, um einen Snapshot des Inhalts der Migration zu erstellen.

Mit der Option Migration können Sie die gewünschten Artefakte auswählen und in einen Snapshot exportieren, z.B. `January16FullApp`. Verwenden Sie den Snapshot-Namen mit diesem Befehl, um anschließend den Exportvorgang zu wiederholen, der nur die Artefakte exportiert, die während des ursprünglichen Exportvorgangs ausgewählt wurden. Informationen hierzu finden Sie unter Artefakte und Anwendungen exportieren in der Dokumentation *Migration für Oracle Enterprise Performance Management Cloud verwalten*.

- Folgende Daten sind nicht Teil der Anwendungs-Snapshots für Planning, Planning Modules und FreeForm:
 - Auditdaten
 - Daten der Jobkonsole

Verwenden Sie den Befehl [cloneEnvironment](#) oder das Feature "Umgebung klonen", wenn Sie Audit- und Jobkonsolendaten in die Zielumgebung kopieren möchten.

- Snapshots enthalten nicht die Staging-Tabellendaten für die Komponente Datenmanagement. Um diese Daten zu importieren, verwenden Sie die Befehle [exportDataManagement](#) und [importDataManagement](#) oder die Schnittstelle für Systemwartungsskripte der Komponente Datenmanagement. Sie können den Befehl [cloneEnvironment](#) oder das Feature "Umgebung klonen" verwenden, um eine identische Kopie der Umgebung zu erstellen, einschließlich der Staging-Tabellendaten für die Komponente Datenmanagement.

Sie können den exportierten Snapshot mit dem Befehl [downloadFile](#) vom Standardspeicherort herunterladen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, die der Anwendungsrolle Migrationsadministrator zugewiesen sind

Verwendung

`epmautomate exportSnapshot SNAPSHOT_NAME wobei SNAPSHOT_NAME der Name des vorhandenen Snapshots in der Migration ist. Dieser Snapshot wird durch den neuen Snapshot ersetzt.`

Beispiel

```
epmautomate exportSnapshot January16FullApp
```

exportTemplate

Exportiert eine Anwendung als Vorlage in eine ZIP-Datei. Die exportierte Datei wird in profitoutbox gespeichert.

Sie können die exportierte Datei mit dem Befehl [downloadFile](#) auf einen lokalen Computer herunterladen.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate exportTemplate APPLICATION_NAME File_Name wobei:`

- `APPLICATION_NAME` der Name der Profitability and Cost Management-Anwendung ist, die als Vorlage exportiert werden soll.
- `File_Name` der Name der Vorlagendatei ist.

Beispiel

```
epmautomate exportTemplate BksML12 template1
```

exportTaskManagerAccessControl

Exportiert den Benutzerdetailbericht für Task Manager-, Supplemental Data- und Enterprise Journal-Benutzerzuweisungen in Financial Consolidation and Close und Tax Reporting. Der Bericht enthält Informationen zu den Benutzern mit vordefinierten Rollen in der Umgebung und listet Attribute der einzelnen Benutzer (z.B. Name und E-

Mail-Adresse) sowie ihren jeweiligen Status, ihre Teams, vordefinierten Rollen, Workflowrollen, Organisationen, Gruppen und Zeitstempel der letzten Anmeldung in einer Excel- oder CSV-Datei auf.

Beispiel für einen Zugriffskontrollebericht in Task Manager:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Name	User Login	Status	Teams	Email	Role	Workflow Task Assignee	Task Approver	Organizatio	Last Login	Form Preparer	Form Approver	Form Integrator	Groups	
1															
2	John Doe	john.doe@example.com	Available	Admins	john.doe@example.com	Administrator									Service Administrator
3	Jane Doe	jane.doe@example.com	Available		jane.doe@example.com	Power User									Power User
4	user Ats	ats_user	Available		example1@example.com	User									User
5	user2 Ats	ats_user2	Available		example2@example.com	user									User
6	View User	viewUser	Available		view.user@exaple.com	Viewer									Viewer

Gilt für

Financial Consolidation and Close und Tax Reporting

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmAutomate exportTaskManagerAccessControl REPORT_NAME`, wobei `REPORT_NAME` der Name der Exportdatei ist, die den Bericht enthält (einschließlich einer gültigen Erweiterung "CSV" oder "XLS").

Dieser Bericht kann im CSV- oder XSL-Format generiert werden. Sie können sie mit dem Befehl [downloadFile](#) herunterladen.

Beispiele

- `epmAutomate exportTaskManagerAccessControl aclreport.csv`
- `epmAutomate exportTaskManagerAccessControl aclreport.xls`

exportValidIntersections

Exportiert gültige Schnittmengengruppen aus dem Geschäftsprozess in eine ZIP-Datei, die Sie mit dem Befehl [downloadFile](#) auf einen lokalen Computer herunterladen können. Gültige Schnittmengen sind Zellschnittmengen, die basierend auf von Ihnen definierten Regeln, sogenannte gültige Schnittmengenregeln, gefiltert werden. Diese Regeln filtern bestimmte Zellschnittmengen für Benutzer, wenn sie Daten eingeben oder Runtime Prompts auswählen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate exportValidIntersections FILE_NAME.zip  
[names=INTERSECTION_NAMES], wobei Folgendes gilt:
```

- *FILE_NAME* ist ein Name für die ZIP-Exportdatei. Alle im Befehl angegebenen gültigen Schnittmengen werden in eine Microsoft Excel-Datei exportiert und anschließend in diese Datei komprimiert.
- *names* (optional) gibt eine durch Komma getrennte Liste gültiger Schnittmengen an, die Sie exportieren möchten. Wenn dieser Parameterwert nicht angegeben ist, exportiert EPM Automate alle gültigen Schnittmengen in der Anwendung.

Beispiele

- **So exportieren Sie spezifische gültige Schnittmengen:**

```
epmautomate exportValidIntersections VI_export_File.zip  
names=VIAccountPeriod,VIEntityPeriod,VIProductPeriod
```
- **So exportieren Sie alle gültigen Schnittmengen:**

```
epmautomate exportValidIntersections VI_export_File.zip
```

extractDimension

Extrahiert eine Oracle Enterprise Data Management Cloud-Dimension in eine Datei oder eine globale Verbindung.

Gilt für

Oracle Enterprise Data Management Cloud

Erforderliche Rollen

Serviceadministrator, Benutzer (mit der Berechtigung "Datenmanager")

Verwendung

```
epmautomate extractDimension APPLICATION DIMENSION EXTRACT_PROFILE  
FILE_NAME [connection=NAME], wobei Folgendes gilt:
```

- *APPLICATION* ist der Name einer Oracle Enterprise Data Management Cloud-Anwendung.
- *DIMENSION* ist der Name der Dimension, die extrahiert werden soll.
- *EXTRACT_PROFILE* ist der Name eines in der Anwendung definierten Profils zum Extrahieren. Dieses Profil wird zum Extrahieren der Dimension verwendet.
- *FILE_NAME* ist der Name einer Datei (CSV für den Export in eine Datei oder ZIP für den Export in Oracle Financials Cloud) zum Speichern der extrahierten Daten. Wenn der Wert des Verbindungsparameters nicht festgelegt ist, wird diese Datei in der Staging Area erstellt. Mit dem Befehl [downloadFile](#) können Sie sie auf einen lokalen Computer herunterladen. Alternativ können Sie sie mit dem Befehl [copyFileFromInstance](#) in eine andere Oracle Enterprise Data Management Cloud-Umgebung kopieren.
- *connection=NAME* (optional) gibt einen globalen Verbindungsnamen (Speicherort einer Instanz) an, der in Oracle Enterprise Data Management Cloud als

Speicherort der Datei definiert ist. Wenn der Wert angegeben ist, wird die extrahierte Datei in die Zielumgebung hochgeladen (Inbox für Oracle Enterprise Performance Management Cloud und das angegebene Dokumentenkonto für Oracle ERP).

 **Note:**

Die in der globalen Verbindung angegebenen Zugangsdaten müssen über die Zugriffsberechtigungen zum Schreiben in der Zielumgebung verfügen.

Beispiele

- So extrahieren Sie eine Datei in die Staging Area von Oracle Enterprise Data Management Cloud: `epmautomate extractDimension USOperations Entity EntityExtProfile EntityData.CSV`
- So extrahieren Sie eine Datei und laden sie in Oracle ERP hoch: `epmautomate extractDimension USOperations Entity EntityExtProfile EntityData.zip Connection=ora_fusion_gl`
- So extrahieren Sie eine Datei und laden sie in die EPM Cloud-Zielinbox hoch: `epmautomate extractDimension USOperations Entity EntityExtProfile EntityData.CSV Connection=EPM_cloud_pln`

feedback

Sendet Feedback an Oracle und an die Serviceadministratoren der Umgebung, und lädt automatisch alle EPM Automate-Logdateien aus dem aktuellen Verzeichnis hoch, die in den letzten 24 Stunden erstellt wurden.

Sie können optional zusätzliche Dateien hochladen (z.B. Fiddler-Tracedateien), die Oracle Support verwenden soll, um zu diagnostizieren, weshalb das aktuelle Problem auftritt.

Dieser Befehl, der der Funktion "Feedback geben" des Service entspricht, ist besonders hilfreich beim Senden von Feedback an Oracle, wenn die Benutzeroberfläche nicht reagiert oder wenn beim Ausführen von EPM Automate ein Problem auftritt.

Informationen zur Funktion "Feedback geben" finden Sie unter Oracle beim Erfassen von Diagnoseinformationen mit dem Utility "Feedback geben" unterstützen in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*.

Dieser Befehl gibt eine Nachricht wie die folgende mit der Information zurück, dass durch Feedback keine Serviceanforderung erstellt wird. Wenn Sie zum Beheben eines Problems Hilfe von Oracle benötigen, müssen Sie eine Serviceanforderung stellen. Durch den Befehl wird eine UDR-Referenznummer angezeigt, die Sie in die Serviceanforderung, die Sie stellen, einschließen müssen.

```
Thank you for providing feedback. If you need Oracle's assistance with this issue please log into My Oracle Support and log a service request.
Make a note of the feedback reference below as you will be asked to provide this information during the SR submission process.
Reference is UDR_502367689_example@example.com_2022_10_17_06_29_41
feedback completed successfully

c:\Oracle\EPM Automate\bin>
```


Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Verwendung

`epmautomate feedback "Comment" [Screenshot="FILE_PATH"] [File="FILE_PATH"]`, wobei Folgendes gilt:

- `Comment` ist Text, der das Problem beschreibt, zu dem dieses Feedback übergeben wird. Kommentare müssen in Anführungszeichen gesetzt werden.
- `Screenshot` (optional) gibt den Namen einer Grafikdatei an, die das Problem veranschaulicht, für das dieses Feedback übergeben wird. Sie können mehrere Screenshots übergeben, indem Sie diesen Parameter und diesen Wert nach Bedarf wiederholen.
- `File` (optional) identifiziert den Namen einer Datei, die Oracle Support zum Beheben des aktuellen Problems verwenden soll. Mit diesem Parameter können Sie Fiddler-Tracedateien oder auch andere Dateien an Oracle weiterleiten. Sie können mehrere Dateien übergeben, indem Sie diesen Parameter und diesen Wert nach Bedarf wiederholen.

Beispiele

- **Windows:** `epmautomate Feedback "runplantypemap CampaignToReporting ClearData=True did not clear data from aggregate storage" Screenshot=C:/feedback/issue.jpg File=exampleScript.ps1 file=trace.har`
- **Linux:** `epmautomate Feedback "runplantypemap CampaignToReporting ClearData=True did not clear data from aggregate storage" Screenshot=/scratch/screens/issue.jpg File=/home/feedback/trace.har`

getApplicationAdminMode

Prüft, ob sich die Anwendung im Administrationsmodus befindet, bei dem der Zugriff auf Serviceadministratoren beschränkt ist.

Dieser Befehl, der `true` zurückgibt, wenn sich die Anwendung im Administrationsmodus befindet (ansonsten wird `false` zurückgegeben), ist sinnvoll für die Prüfung des Status der Anwendung vor dem Ausführen von Automatisierungsskripten. Beispiel: Für den Befehl [refreshCube](#) ist es erforderlich, dass sich die Anwendung im Administrationsmodus befindet. Sie können diesen Befehl im Automatisierungsskript wie folgt verwenden, um zu prüfen, ob sich die Anwendung im Administrationsmodus befindet.

```
adminMode = `epmautomate.sh getApplicationAdminMode`
if ["$adminMode" == "true"]
    epmautomate.sh refreshCube
```

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Account Reconciliation, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate getApplicationAdminMode
```

Beispiel

```
epmautomate getApplicationAdminMode
```

getDailyMaintenanceStartTime

Zeigt in der Konsole die koordinierte Weltzeit (UTC) oder optional die Zeitzone für den geplanten Start der täglichen Wartung für die Umgebung an.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate getDailyMaintenanceStartTime [timezone=true|false] wobei  
timezone=true (optional) angibt, ob die Startzeit der täglichen Wartung in der beim Einrichten  
angegebenen Zeitzone angezeigt werden soll, z.B. America/Los_Angeles. Der Standardwert  
ist false.
```

Beispiele

- Wartungszeit in der beim Einrichten angegebenen Zeitzone anzeigen:

```
epmautomate getDailyMaintenanceStartTime timezone=true
```
- Wartungszeit in UTC anzeigen:

```
epmautomate getDailyMaintenanceStartTime
```

getEssbaseQryGovExecTime

Zeigt die aktuell geltende maximale Zeit (in Sekunden) an, die eine Oracle Essbase-Abfrage zum Abrufen und Ausgeben von Informationen verwenden darf, bevor die Abfrage beendet wird.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Profitability and Cost Management, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate getEssbaseQryGovExecTime
```

Beispielbefehlsausgabe:

```
c:\Oracle\EPM Automate\bin>epmautomate getEssbaseQryGovExecTime
300

c:\Oracle\EPM Automate\bin>epmautomate setEssbaseQryGovExecTime 600
setEssbaseQryGovExecTime completed successfully

c:\Oracle\EPM Automate\bin>epmautomate getEssbaseQryGovExecTime
600
```

Beispiel

```
epmautomate getEssbaseQryGovExecTime
```

getIdleSessionTimeout

Zeigt das Sessiontimeout (in Minuten) der Umgebung Oracle Enterprise Performance Management Cloud an. Wenn eine Session für diese Zeitdauer inaktiv ist, werden Benutzer zur Anmeldeseite umgeleitet.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate getIdleSessionTimeout
```

Beispielbefehlsausgabe:

```
c:\Oracle\EPM Automate\bin>epmautomate getIdleSessionTimeout  
75
```

getIPAllowlist

Zeigt für OCI-Umgebungen (2. Generation) die IP-Adressen und Classless Inter-Domain Routings (CIDRs) an, die in der aktuellen Ausnahmeliste enthalten sind.

Mit diesem Befehl können Sie prüfen, ob bestimmte IP-Adressen oder CIDRs derzeit auf eine OCI-Umgebung (2. Generation) zugreifen dürfen.



Note:

Dieser Befehl kann nicht dazu verwendet werden, IP-Adressen und CIDRs in Classic-Umgebungen aufzulisten. Für Classic-Umgebungen können Sie auf dem Bildschirm "Servicedetails" von "Meine Services (Classic)" mit Ausnahmelisten- oder Ablehnungslistenregeln arbeiten.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmAutomate getIPAllowlist
```

 **Note:**

Um alle vorhandenen IP-Adressen und CIDRs in eine Datei zu schreiben, leiten Sie die Ausgabe in eine Textdatei um. Diese können Sie bearbeiten (IP-Adressen und CIDRs ganz oder teilweise entfernen) und in die Umgebung hochladen. Anschließend können Sie mit dem Befehl [setIPAllowlist](#) die Einträge in der Datei aus der Ausnahmeliste entfernen. Beispiel für Befehlsausführung:

```
epmAutomate getIPAllowlist > myRemoveList.txt
epmAutomate uploadFile myRemoveList.txt
epmAutomate setIPAllowlist remove myRemoveList.txt
```

Beispiel

IP-Adressen und CIDRs in der aktuellen Ausnahmeliste anzeigen:

```
epmAutomate getIPAllowlist
```

getSubstVar

Ruft die Werte von Substitutionsvariablen ab und zeigt sie im Fenster an.

Das Anzeigeformat ist *CUBE_NAME.SUBSTVAR=value*, z.B. *Plan2.CurYear=2016*. Werte von Substitutionsvariablen auf Anwendungsebene werden im Format *ALL.SUBSTVAR=value* angezeigt. Beispiel: *ALL.CurYear=2016*

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser (mit Regelausführungszugriff)

Verwendung

`epmautomate getSubstVar CUBE_NAME|ALL [name=VARIABLE_NAME]` wobei Folgendes gilt:

- *CUBE_NAME* ist der Cube (z.B. Plan1, Plan2), aus dem Sie die Substitutionsvariable abrufen möchten. Verwenden Sie *ALL*, um Substitutionsvariablen auf Anwendungsebene abzurufen.
- *name=VARIABLE_NAME* gibt optional die Substitutionsvariable an, deren Wert Sie abrufen möchten. Wenn Sie keinen Variablennamen angeben, ruft der Befehl den Wert aller Substitutionsvariablen ab.

Beispiele

- Wert aller Substitutionsvariablen auf Anwendungs- und Cube-Ebene abrufen:
`epmautomate getSubstVar ALL`
- Wert einer bestimmten Substitutionsvariablen auf Anwendungsebene abrufen:
`epmautomate getSubstVar ALL name=CurYear`
- Wert aller Substitutionsvariablen auf Cube-Ebene abrufen: `epmautomate getSubstVar Plan2`
- Wert einer bestimmten Substitutionsvariablen auf Cube-Ebene abrufen: `epmautomate getSubstVar Plan2 name=CurYear`

getVirusScanOnFileUploads

Überprüft, ob in Ihrer OCI-Umgebung (2. Generation) das Scannen aller hochgeladenen Dateien aktiviert ist, um sicherstellen, dass diese virenfrei sind.

Dieser Befehl überprüft, ob vor dem Hochladen von Dateien in die Umgebung ein Virensan erzwungen wird.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate getVirusScanOnFileUploads
```

Dieser Befehl gibt `true` aus, wenn in der Umgebung das Virenschannen von hochgeladenen Dateien aktiviert ist. Ist dies nicht der Fall, wird `false` ausgegeben.

groupAssignmentAuditReport

Erstellt einen Bericht, in dem Benutzer und Gruppen aufgelistet werden, die während eines angegebenen Datumsbereichs zu Zugriffskontrollgruppen hinzugefügt oder daraus entfernt wurden.

Dieser Bericht wird als CSV-Datei generiert und kann zur Unterstützung von Sicherheitsauditvorgängen verwendet werden. Die Zeilen der generierten CSV-Datei enthalten die Benutzer bzw. Gruppen, die hinzugefügt oder entfernt wurden, die Gruppen, denen Benutzer oder Gruppen hinzugefügt bzw. aus denen sie entfernt wurden, den Serviceadministrator, der die Aktion ausgeführt hat, und Datum und Uhrzeit des Aktionsabschlusses. Dieser Bericht enthält keine Auditinformationen dazu, wann Gruppen zur Zugriffskontrolle hinzugefügt oder daraus gelöscht wurden.

	A	B	C	D	E
1	User/Group Name	Group	Action	Administrator	Date and Time
2	testGroup1	exampleGroup1	Added	john.smith@example.com	May 10, 2022 23:13:20 UTC
3	john.doe@example.com	exampleGroup1	Added	joe.doe@example.com	May 11, 2022 23:14:20 UTC
4	JaneDoe@example.com	testGroup1	Added	joe.doe@example.com	May 11, 2022 23:14:50 UTC
5	jaDoe@example.com	testGroup1	Added	john.smith@example.com	May 12, 2022 23:25:20 UTC
6	JaneDoe@example.com	testGroup1	Removed	john.smith@example.com	May 13, 2022 18:13:20 UTC
7	jaDoe@example.com	testGroup1	Removed	john.smith@example.com	May 13, 2022 18:22:20 UTC
8	testGroup1	exampleGroup1	Removed	joe.doe@example.com	May 13, 2022 23:13:20 UTC

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmAutomate groupAssignmentAuditReport FROM_DATE TO_DATE REPORT_NAME`. Dabei gilt:

- `FROM_DATE` gibt das Startdatum der Periode (im Format YYYY-MM-DD) an, für die der Bericht generiert werden soll.
- `TO_DATE` gibt das Enddatum der Periode (im Format YYYY-MM-DD) an, für die der Bericht generiert werden soll.
- `REPORT_NAME` ist der Name einer CSV-Datei für den Bericht. Sie können den generierten Bericht mit dem Befehl [downloadFile](#) herunterladen.

Beispiel

```
epmAutomate groupAssignmentAuditReport 2022-03-01 2022-05-01
GroupAssignmentReport.CSV
```

help

Zeigt Hilfe für alle EPM Automate-Befehle an.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Verwendung

`epmautomate help`

Beispiel

```
epmautomate help
```

importAppAudit

Importiert Datensätze für den Datenaudit aus einer ZIP-Datei, die Sie durch Exportieren der Auditdaten aus einer Umgebung erstellt haben.

Sie erstellen die Importdatei mit dem Befehl [exportAppAudit](#) (`epmautomate exportAppAudit auditData ndays=All`). Verwenden Sie diesen Befehl, um Datensätze für den Datenaudit aus einer Umgebung bei der Migration in eine andere Umgebung oder für die Disaster Recovery zu klonen.

Gilt für

Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate importAppAudit FILE_NAME [logFilename=LOG_FILE_NAME]`, wobei Folgendes gilt:

- `FILE_NAME` ist der Name einer ZIP-Datei mit den Datensätzen für den Datenaudit, die Sie in die Anwendung importieren möchten. Bevor Sie diesen Befehl ausführen, müssen Sie mit dem Befehl [uploadFile](#) die Datei in die Umgebung hochladen.
- `logFileName` (optional) gibt die Fehlerlogdatei an, in der die beim Import gefundenen Fehler aufgezeichnet werden. Wenn dieser Wert nicht angegeben ist, generiert der Befehl eine Fehlerdatei, die nach folgender Konvention benannt wird: `username_date_timestamp`. Sie können diese Datei mit dem Befehl [downloadFile](#) herunterladen.

Beispiel

```
epmautomate importAppaudit Audit_data.zip logFileName=auditImportLog
```

importAppSecurity

Lädt Zugriffsberechtigungen für Benutzer oder Gruppen einer Anwendung aus einer CSV-Datei, die in der Inbox verfügbar ist.

Wenn Sie Zugriffsberechtigungen importieren, werden bestehende Zuweisungen nur für importierte Elemente, Eingabeformulare, Eingabeformularordner, Aufgabenlisten, Calculation Manager-Geschäftsregeln und Geschäftsregelordner überschrieben. Alle anderen bestehenden Zugriffsberechtigungen bleiben unverändert.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate importAppSecurity ACL_FILE_NAME ERROR_FILE [clearall=true|false]` wobei Folgendes gilt:

- `ACL_FILE_NAME` ist der Name einer CSV-Datei mit Zugriffsberechtigungen, die Sie in die Anwendung importieren möchten. Bevor Sie diesen Befehl ausführen, müssen Sie diese Datei mit dem Befehl [uploadFile](#) in die Inbox hochladen. Der Inhalt einer Beispieleingabedatei kann wie in der folgenden Abbildung aussehen:

	A	B	C	D	E	F	G	H
1	Object Name	Name	Parent	Is User	Object Type	Access Type	Access Mode	Remove
2	AllIAB	Interactive User		N	SL_DIMENSION	READWRITE	@IDESCENDANTS	N
3	FormsB	Interactive User	/	N	SL_FORMFOLDER	READWRITE	@IDESCENDANTS	N
4	Statistics	Interactive User		N	SL_DIMENSION	READWRITE	@IDESCENDANTS	N
5	TD	Interactive User		N	SL_DIMENSION	READWRITE	@IDESCENDANTS	N
6	No Entity	Interactive User		N	SL_DIMENSION	READWRITE	MEMBER	N

Eine Beschreibung der Spaltenheader und der möglichen Werte finden Sie unter "Sicherheitsberechtigungen importieren" in der Dokumentation *REST-API für Oracle Enterprise Performance Management Cloud*.

- `ERROR_FILE` ist der Name einer CSV-Datei, die EPM Automate in der Outbox erstellt, um die während des Vorgangs erkannten Fehler zu erfassen. Sie können diese Datei auf einen lokalen Computer herunterladen, um die erfassten Fehler zu analysieren und zu korrigieren. Der Inhalt einer Beispielfehlerdatei kann wie in der folgenden Abbildung aussehen. Die Spalten dieser Datei entsprechen den Headerspalten der Eingabedatei:

	A	B	C	D	E	F	G	H
1	AllIAB	Interactive User		N	SL_DIMENSION	READWRITE	@IDESCENDANTS	N
2	FormsB	Interactive User	/	N	SL_FORMFOLDER	READWRITE	@IDESCENDANTS	N

- `clearall` gibt optional an, ob die vorhandenen Zugriffsberechtigungen gelöscht werden sollen, bevor neue Berechtigungen aus der Datei geladen werden. Der Standardwert ist `false`.

Beispiel

```
epmautomate importAppSecurity Acl_file.CSV Acl_import_error.CSV
clearall=true
```

importARApplicationProperties

Importiert Anwendungseinstellungen (Redwood-Design, Theme, E-Mail-Benachrichtigung und Geschäftsprozessname), Logo und verfügbare

Hintergrundbilder in einer JSON-Exportdatei in eine Account Reconciliation-Umgebung.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate importARApplicationProperties FILE_NAME`, wobei *FILE_NAME* der Name der JSON-Datei ist, die aus einer Umgebung exportiert wird.

Diese Datei wird mit dem Befehl [exportARApplicationProperties](#) aus einer anderen Umgebung exportiert und muss in der Umgebung verfügbar sein, in der Sie die Anwendungseinstellungen wiederherstellen.

Beispiel

```
epmautomate importARApplicationProperties myProp.JSON
```

importBackgroundImage

Importiert das Hintergrundbild aus einer Exportdatei in eine Account Reconciliation-Umgebung.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate importBackgroundImage FILE_NAME.jpg`, wobei *FILE_NAME* der Name der Datei für das Hintergrundbild ist, die aus einer anderen Umgebung exportiert wurde.

Beispiel

```
epmautomate importBackgroundImage image_file.jpg
```

importBalances

Verwendet Datenmanagement zum Importieren von Saldendaten aus einer Dataload-Definition.

Gilt für

Account Reconciliation.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Benutzern mit den Rollen Poweruser, Benutzer und Viewer muss über die ACL zusätzliche Sicherheit zugewiesen werden.

Verwendung

epmautomate importBalances *DL_DEFINITION PERIOD*, wobei Folgendes gilt:

- *DL_DEFINITION* ist eine vorhandene Dataload-Definition in Account Reconciliation.
- *PERIOD* ist der Name einer Periode.

Beispiel

epmautomate importBalances DailyLoad "January 2020"

importCellLevelSecurity

Importiert Sicherheitseinstellungen auf Zellebene aus einer ZIP-Datei, die eine Excel-Datei mit Sicherheitsdatensätzen enthält, in den Geschäftsprozess. Bevor Sie diesen Befehl ausführen, müssen Sie mit dem Befehl [uploadFile](#) die Importdatei in die Umgebung hochladen.

Für einen erfolgreichen Import der Sicherheit auf Zellebene muss Ihre importierte ZIP-Datei eine Excel-Datei mit zwei Arbeitsblättern ("Rules" und "Sub Rules") enthalten. Das Arbeitsblatt "Rules" muss Sicherheitsdefinitionen auf Zellebene, einschließlich Dimensionen, und Eigenschaften wie "Unspecified Valid" (Nicht angegebene Elemente gültig) und "Additional Dims Required" (Zusätzliche Dimensionen erforderlich) enthalten. Das Arbeitsblatt "Sub Rules" muss Elementauswahlen und Ausschlüsse enthalten. Die beste Methode zum Abrufen der Formatvorlage für die Importdatei ist, die Sicherheit auf Zellebene aus der Anwendung zu exportieren. In den folgenden Abbildungen ist ein Beispielformat dargestellt.

	Name	Position	Description	Enabled	Valid Cubes	Anchor Dim Name	Anchor Dimension Apply to Selected	Dim1	Dim1 Required	Dim2	Dim2 Required
1											
2	Sample-CLS	1		true	All	Product	true	Account	false		
3	Sample-CLS-Dup	2		true	All	Product	true	Account	false		
4											
5											

	Name	Users	User Groups	Restriction	Anchor Members	Anchor Exclusion	Dim1 Members	Dim1 Exclusion	Dim2 Members	Dim2 Exclusion	Dim3 Members	Dim3 Exclusion
1												
2	Sample-CLS		Service Administrator	Deny Write	P_TP		Statistics					
3	Sample-CLS-Dup		Service Administrator	Deny Write	P_TP		Statistics					
4												

Gilt für

Planning, Planning Modules, FreeForm, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate importCellLevelSecurity FILE_NAME.ZIP [ErrorFile=FILE_NAME.txt]`, wobei Folgendes gilt:

- `FILE_NAME` ist der Name der ZIP-Datei, die die Excel-Datei mit den Informationen zur Sicherheit auf Zellenebene enthält.
- `ErrorFile` (optional) gibt den Namen der Textdatei an, in die Fehlerdatensätze geschrieben werden. Wenn dieser Parameterwert nicht angegeben ist, erstellt EPM Automate automatisch eine Fehlerdatei. Den Namen dieser Datei können Sie in der Jobkonsole anzeigen.
Mit dem Befehl [downloadFile](#) können Sie die Fehlerdatei auf einen lokalen Computer herunterladen.

Beispiel

```
epmautomate importCellLevelSecurity ImportCLSDRecordsFile.zip
ErrorFile=ImportCLSDRecords_errors.txt
```

importConsolidationJournals

Importiert Konsolidierungsjournale aus einer JLF-Datei in Financial Consolidation and Close.

- Erstellen Sie mit dem Befehl [exportConsolidationJournals](#) die JLF-Datei, die als Eingabe für diesen Befehl verwendet wird.
- Bevor Sie diesen Befehl ausführen, müssen Sie die Eingabedatei mit dem Befehl [uploadFile](#) in die Umgebung hochladen.

Gilt für

Financial Consolidation and Close

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate importConsolidationJournals jobName [fileName=FILE_NAME]
[errorFileName=ERROR_FILE_NAME], wobei
```

- `jobName` ist der Name eines Journalimportjobs, der in Financial Consolidation and Close erstellt wurde.
- `fileName` (optional) ist der Name einer JLF-Datei, aus der die Journaleinträge importiert werden sollen.

- `errorFileName` (optional) ist der Name der Logdatei zur Aufzeichnung der Meldungen, die während des Importprozesses generiert werden.

Beispiel

```
epmautomate importConsolidationJournals "JIMPORT1"
fileName="TestImport1.jlf" errorFileName="TestImport1_error.log"
```

importData

Die Daten aus einer Datei werden mit den Einstellungen in die Anwendung importiert, die in einem Job vom Typ `import data` (Daten importieren) festgelegt sind.

Verwenden Sie den Befehl [uploadFile](#), um die Datei mit den Anwendungsdaten an den Standardspeicherort zum Hochladen hochzuladen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate importData JOB_NAME [FILE_NAME] errorFile=ERROR_FILE.zip` wobei Folgendes gilt:

- `JOB_NAME` ist der Name eines in der Anwendung definierten Jobs.
- `FILE_NAME` gibt optional den Namen der ZIP-, CSV- oder TXT-Datei (Datendatei im Essbase-Format) an, aus der Daten importiert werden sollen. Wenn Sie einen Dateinamen angeben, wird der Importdateiname im Job ignoriert. Wenn der Job so definiert ist, dass er Daten im Essbase-Format importiert, muss die ZIP-Datei eine TXT-Datei im Essbase-Format enthalten. Für andere Importjobs darf die ZIP-Datei mindestens eine CSV-Datei enthalten, in der die Importreihenfolge in den Dateinamen enthalten ist, z.B. `data1-3.csv`, `data2-3.csv` und `data3-3.csv`.
- `errorFile` gibt optional den Namen einer ZIP-Datei an, in der Datensätze dokumentiert werden, die während der Importvorgänge abgelehnt wurden (sofern vorhanden). Sofern in der Outbox eine ZIP-Datei mit demselben Namen vorhanden ist, wird diese überschrieben. Sie können diese Datei mit dem Befehl [downloadFile](#) herunterladen.

Beispiel

```
epmautomate importData dailydataload dailydata.zip
errorFile=dataImport_error.zip
```

importDataManagement

Importiert Datenmanagement-Datensätze aus einer ZIP-Datei in eine Umgebung.

Mit diesem Befehl werden Daten aus einer mit dem Befehl [exportDataManagement](#) erstellten ZIP-Datei in Setup- und Staging-Tabellen importiert. Verwenden Sie den Befehl [uploadFile](#), z.B. `epmAutomate uploadFile "C:/datafile/datafile.zip" inbox`, um die Import-ZIP-Datei in die Datenmanagement-Inbox oder einen darin enthaltenen Ordner hochzuladen.

 **Note:**

Mit diesem Befehl können nur die Datenmanagement-Datensätze importiert werden, die aus einer anderen Umgebung mit derselben monatlichen Aktualisierung exportiert wurden. Beispiel: Datensätze, die aus einer 21.11 Oracle Enterprise Performance Management Cloud-Umgebung exportiert werden, können nur in eine andere 21.11-Umgebung importiert werden.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Account Reconciliation, Tax Reporting, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate importDataManagement FILE_NAME.zip`, wobei *FILE_NAME* der Name der ZIP-Datei ist, in der die zu importierenden Datenmanagement-Daten enthalten sind.

Beispiele

- Aus der Datenmanagement-Inbox importieren:
`epmautomate importDataManagement inbox/dataFile.zip`
- Aus einem Ordner in der Inbox importieren:
`epmautomate importDataManagement inbox/dm_data/dataFile.zip`

importDimension

Importiert eine Dimension aus einer Datei in eine Oracle Enterprise Data Management Cloud-Anwendung.

Mit diesem Befehl können Sie eine Importdatei aus einer Verbindung importieren, die in Oracle Enterprise Data Management Cloud oder in der Staging Area definiert ist.

Wenn die Datei aus der Staging Area von Oracle Enterprise Data Management Cloud importiert werden soll, müssen Sie sie mit dem Befehl [uploadFile](#) in die Oracle Enterprise Data Management Cloud-Zielumgebung hochladen. Sie können auch mit dem Befehl [copyFileFromInstance](#) die Datei aus einer anderen Oracle Enterprise Performance Management Cloud-Umgebung kopieren.

Gilt für

Oracle Enterprise Data Management Cloud.

Erforderliche Rollen

Serviceadministrator, Benutzer (mit der Berechtigung "Datenmanager")

Verwendung

`epmautomate importDimension APPLICATION DIMENSION IMPORT_TYPE FILE_NAME`
[`connection=NAME`] wobei Folgendes gilt:

- `APPLICATION` ist der Name einer Oracle Enterprise Data Management Cloud-Anwendung.
- `DIMENSION` ist der Name der Anwendungsdimension, die importiert wird.
- `IMPORT_TYPE` gibt an, wie der Import durchgeführt werden soll. Die gültigen Importtypen lauten wie folgt:
 - `ResetDimension` zum Löschen aller vorhandenen Dimensionsdaten und Importieren der neuen Daten
 - `ReplaceNodes` zum Hinzufügen oder Aktualisieren von Knoten und Ersetzen vorhandener Hierarchien beim Import
 - `Merge` zum Verarbeiten inkrementeller Änderungen an den Knoten und Hierarchien mit einer Importanforderung
- `FILE_NAME` ist der Name der Datei (CSV oder ZIP), die die zu importierenden Dimensionsdaten enthält. Der Dateiname muss auf den Dimensionsnamen enden, dem ein Unterstrich ("_") vorangestellt wird. Beispiel: `import_Entity.csv`. Wenn für den Import eine ZIP-Datei verwendet wird, die mehrere Importdateien enthält, ist dieser Befehl vom Dateinamen in der ZIP-Datei abhängig, damit die richtige Importdatei erkannt wird.
Wenn für `connection` ein Wert angegeben ist, muss die Dimension aus einer ZIP-Datei importiert werden, z.B. von `importdata_Entity.zip`.
- `connection=NAME` (optional) gibt einen Verbindungsnamen (Speicherort einer Instanz) an, der in Oracle Enterprise Data Management Cloud als Speicherort der Importdatei definiert ist. Sollte er nicht angegeben sein, wird die lokale Staging Area vom Importprozess nach der Importdatei durchsucht.

Beispiele

- So importieren Sie aus einer Datei, die in die Staging Area hochgeladen wurde:
`epmautomate importDimension USOperations Entity ReplaceNodes data_Entity.CSV`
- Aus der Outbox einer anderen EPM Cloud-Umgebung importieren: `epmautomate importDimension USOperations Entity ReplaceNodes data_Entity.ZIP Connection=EPM_Cloud_pln`

importJobConsole

Klont Jobkonsolendatensätze mit einer ZIP-Datei, in der die entsprechenden aus einer Umgebung exportierten Jobkonsolendatensätze enthalten sind.

Der Import von Datensätzen der Jobkonsole mit diesem Befehl ist eine einmalige Aufgabe, die nach der Ausführung des Befehls `recreate` ausgeführt werden muss. Wenn Sie bereits Jobkonsolendatensätze mit diesem Befehl importiert haben,

verlaufen nachfolgende Aufrufe des Befehls nicht erfolgreich, bis Sie die Umgebung neu erstellen.

Verwenden Sie den Befehl `exportJobConsole` (`epmAutomate exportJobConsole FILE_NAME.zip nDays=All jobTypes=All jobStatusCode=All`), um die ZIP-Datei zu erstellen, die als Eingabe für diesen Befehl verwendet wird.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate importJobConsole FILE_NAME.zip [logFileName=jobConsoleLog]`. Dabei gilt:

- `FILE_NAME` ist der Name der ZIP-Datei, in der die Jobkonsolendatensätze enthalten sind, die Sie importieren möchten. Verwenden Sie den Befehl `uploadFile`, um diese Datei in die Umgebung hochzuladen.
- `logFileName` (optional) gibt `jobConsoleLog` als Logdatei an, in der die beim Import gefundenen Fehler aufgezeichnet werden. Wenn dieser Wert nicht angegeben ist, generiert der Befehl eine Fehlerdatei mit der folgenden Namenskonvention: `usernameimportLog_date_timestamp.zip`. Sie können diese Datei mit dem Befehl `downloadFile` herunterladen.

Beispiel

```
epmautomate importJobConsole jobConsole.zip jobConsoleLog
```

importLibraryArtifact

Importiert Bibliotheksartefakte aus einem Archiv oder einer Datei in die Narrative Reporting-Bibliothek.

Bevor Sie diesen Befehl ausführen, laden Sie das Quellarchiv bzw. die Quelldatei mit dem Befehl `uploadFile` in die Umgebung hoch.

Gilt für

Narrative Reporting

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzern mit den Rollen Poweruser, Benutzer und Viewer muss über die ACL zusätzliche Sicherheit zugewiesen werden.

Verwendung

```
epmautomate importLibraryArtifact SOURCE_FILE [errorFile=ERROR_FILE.txt]
[importFormat=Native|File] [importFolder=FOLDER_PATH] [importPermission=true|
false] [overwrite=true|false], wobei Folgendes gilt:
```


- *SOURCE_FILE* ist der Name des Archivs, das die Artefakte enthält, die in die Bibliothek importiert werden sollen. Diese Datei muss in der Inbox verfügbar sein.
- *errorFile* (optional) ist ein eindeutiger Name für die Textdatei, in der Importfehler gespeichert werden.
- *importFormat* (optional) entspricht einem der folgenden Formate:
 - *Native* importiert Artefakte aus einer ZIP-Datei, die mit dem Befehl [exportLibraryArtifact](#) mit der Option `exportFormat=Native` erstellt wurde. Das ist der Standardwert.
 - *File* importiert eine Binärdatei.

Hinweis:

Verwenden Sie den Befehl [importSnapshot](#), um ZIP-Dateien mit Bibliotheksartefakten (die mit dem Befehl [exportLibraryArtifact](#) mit der Option `exportFormat=LCM` erstellt wurden) in Umgebungen in Financial Consolidation and Close, Planning, Planning Modules oder Tax Reporting zu importieren.

- *importFolder* (optional) ist der Speicherort der Bibliothek, in der die importierten Artefakte gespeichert werden. Geben Sie den Pfad an, wenn dieser Speicherort sich von *Library* (dem Standardspeicherort für Importe) unterscheidet.
- *importPermission* gibt an, ob die Zugriffsberechtigungen importiert werden, die für die Artefakte festgelegt sind. Der Standardwert ist `False`.
- *overwrite* gibt an, ob Artefakte mit identischem Namen (sofern vorhanden) am angegebenen Bibliotheksspeicherort überschrieben werden sollen. Der Standardwert lautet `False`. Das bedeutet, dass beim Prozess ein Artefakt nicht importiert wird, wenn am Speicherort für den Import ein Artefakt mit identischem Namen vorhanden ist.

Verwenden Sie nach Abschluss des Imports den Befehl [downloadFile](#), um die Fehlerdateien auf einen lokalen Computer herunterzuladen.

Beispiele

- Dateien im binären Format importieren:

```
epmautomate importLibraryArtifact newReports.doc
errorFile=report_imp_errors.txt importFormat=File
importFolder="Library/My Reports" importPermission=true overwrite=true
```
- Artefakte aus einer exportierten ZIP-Datei importieren:

```
epmautomate importLibraryArtifact newReports.zip
errorFile=report_imp_errors.txt importFormat=Native
importFolder="Library/My Reports" importPermission=true overwrite=true
```
- Importiert Berichte aus einer exportierten ZIP-Datei in eine Umgebung in Financial Consolidation and Close, Planning, Planning Modules oder Tax Reporting:

```
epmautomate importSnapshot newReports.zip
```

importLogoImage

Importiert das Unternehmenslogo, das in einer Account Reconciliation-Umgebung verwendet wird, aus einer Exportdatei in eine andere Umgebung.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate importLogoImage IMAGE_NAME.jpg`, wobei *IMAGE_NAME* der Name der Bilddatei für das Logo ist.

Sie können das exportierte Bild mit dem Befehl [downloadFile](#) herunterladen. Laden Sie es mit dem Befehl [uploadFile](#) in die Zielumgebung hoch, und führen Sie dann den Befehl [importLogoImage](#) aus, um es zu importieren.

Beispiel

```
epmautomate importLogoImage corpLogo.jpg
```

importMapping

Importiert Zuordnungen aus einer Zuordnungsimportdatei, die zuvor in die Umgebung hochgeladen wurde.

Verwenden Sie den Befehl [uploadFile](#), um Dateien in die Datenmanagement-Inbox oder in einen zugehörigen Unterordner hochzuladen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

```
epmautomate importMapping DIMENSION_NAME|ALL FILE_NAME IMPORT_MODE  
VALIDATION_MODE LOCATION wobei Folgendes gilt:
```

- *DIMENSION_NAME|ALL* gibt den Empfänger der Zuordnung an. Geben Sie den Namen der Dimension an, in die Zuordnungen importiert werden sollen, oder geben Sie *ALL* an, um alle Zuordnungen in die entsprechenden Dimensionen zu importieren, die in der Datei enthalten sind.
- *FILE_NAME* ist der Name und der Speicherort der Zuordnungsimportdatei, die in der Datenmanagement-Inbox oder einem zugehörigen Unterverzeichnis verfügbar ist. Geben Sie den Dateinamen (TXT-Dateien im standardmäßigen Data Management-Format) und

den zugehörigen Pfad an (z.B. `inbox/AccountMap.txt` oder `inbox/pbcs_maps/AccountMap.txt`).

- `IMPORT_MODE` lautet entweder `REPLACE`, um vor dem Importieren von Zuordnungen vorhandene Zuordnungsregeln zu löschen, oder `MERGE`, um vorhandenen Regeln neue Zuordnungsregeln hinzuzufügen.
- `VALIDATION_MODE` ist auf `TRUE` festgelegt, um Zielelemente anhand der Anwendung zu validieren, oder auf `FALSE`, um die Zuordnungsdatei zu laden, ohne Validierungen auszuführen.
- `LOCATION` ist der Datenmanagement-Speicherort, für den Zuordnungsregeln geladen werden sollen.

Beispiele

- `epmautomate importMapping Account inbox/AccountMap.txt MERGE FALSE "France Sales"`
- `epmautomate importMapping ALL "inbox/France Sales/AllMaps.txt" MERGE FALSE "France Sales"` (lädt Zuordnungen aus der Zuordnungsimportdatei in alle zugeordneten Dimensionen im Speicherort "France Sales").

importMetadata

Metadaten werden mit den Importeinstellungen, die in einem Job vom Typ `import metadata` (Metadaten importieren) festgelegt sind, in die Anwendung importiert. Sie können optional den Namen der ZIP-Datei angeben, aus der die Metadaten importiert werden sollen.

Verwenden Sie den Befehl [uploadFile](#), um die Datei mit den Metadaten in den Standardspeicherort zum Hochladen hochzuladen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate importMetadata JOB_NAME [FILE_NAME] errorFile=ERROR_FILE.zip`
wobei Folgendes gilt:

- `JOB_NAME` ist der Name eines in der Anwendung definierten Jobs.
- `FILE_NAME` gibt optional den Namen der ZIP-Datei an, aus der Metadaten importiert werden sollen. Falls angegeben, hat der Inhalt dieser ZIP-Datei Vorrang vor den im Job definierten Dateinamen. Die ZIP-Datei kann mindestens eine CSV-Datei enthalten. Die Namen der Dateien, die Metadaten für Dimensionen enthalten, müssen mit den im Job definierten Importdateinamen übereinstimmen oder auf `_DIMENSIONNAME.csv` enden, z.B. `metadata_Entity.csv`, `metadata_HSP_Smart Lists.csv` und `metadata_Exchange Rates.csv`.

- `errorFile` gibt optional den Namen einer ZIP-Datei an, in der Datensätze dokumentiert werden, die während der Importvorgänge abgelehnt wurden (sofern vorhanden). Sofern in der Outbox eine ZIP-Datei mit demselben Namen vorhanden ist, wird diese überschrieben. Sie können diese Datei mit dem Befehl [downloadFile](#) herunterladen.

 **Hinweis:**

- Sie können Elemente nicht umbenennen, indem Sie einen Job zum Importieren von Metadaten mit einer Ladedatei ausführen, in der die Eigenschaften `old_name` oder `unique_name` geändert wurden. Das Umbenennen von Elementen wird ignoriert.
- Sie können keine Attributdimensionen löschen, während Sie Metadaten mit diesem Befehl importieren.
- Es werden nur Metadaten für Dimensionen importiert, für die der Import von Metadaten im Job eingerichtet wurde. Metadaten für anderen Dimensionen, die in der ZIP-Datei enthalten sind, werden ignoriert.

Wenn beide der folgenden Bedingungen für die ZIP-Datei wahr sind, wird eine mehrdeutige Importsituation erstellt:

- Die ZIP-Datei enthält eine Metadatendatei mit einem Namen, der mit dem im Job definierten Dateinamen übereinstimmt.
- Die ZIP-Datei enthält eine Metadatendatei oder Dateien mit Namen, die auf `_DIMENSIONNAME.CSV` oder `_DIMENSIONNAME.TXT` enden, wobei `DIMENSIONNAME` der Name der Dimension ist, in die Metadaten importiert werden.

Oracle empfiehlt, dass die ZIP-Datei eine Metadatendatei mit einem Namen enthalten sollte, der identisch mit dem im Job referenzierten Namen ist, oder eine Datei mit einem Namen, der auf `_DIMENSIONNAME.CSV` (oder `_DIMENSIONNAME.TXT`) endet, jedoch nicht beide. Beispiel: Wenn Sie einen Job laden, der die Metadatendatei `Employees_A-Z.CSV` in der `Employees`-Dimension referenziert, enthält Ihre ZIP-Datei möglicherweise die Datei `Employees_A-Z.CSV` oder `New_Employees.CSV`, jedoch nicht beide. Wenn Ihre ZIP-Datei `Employees_A-Z.CSV` und `New_Employees.CSV` enthält, kann EPM Automate eine der beiden Dateien für den Import auswählen, je nachdem in welcher Reihenfolge die Dateien in der ZIP-Datei enthalten sind. Die Datei `Employees_A-Z.CSV` stellt eine mögliche Übereinstimmung für den Import dar, da ihr Name mit dem im Job referenzierten Namen übereinstimmt. Die Datei `New_Employees.CSV` stellt ebenfalls eine mögliche Übereinstimmung dar, da ihr Name mit dem Muster `_DIMENSIONNAME.CSV` übereinstimmt.

Beispiel

```
epmautomate importMetadata importAccount importAccount.zip
errorFile=metadataImport_error.zip
```

importOwnershipData

Importiert Anteilsdaten aus einer in der Umgebung verfügbaren CSV-Datei in eine Periode.

Bevor Sie diesen Befehl ausführen, müssen Sie die CSV-Datei der Importquelle mit dem Befehl [uploadFile](#) in die Umgebung hochladen.

Der Header dieser CSV-Datei sieht folgendermaßen aus:

```
Scenario, Year, Period, Entity, Parent, POwn, Control, Method
```

Die Werte `POwn`, `Control` und `Method` sind optional.

Die importierten Anteilsdaten werden mit allen vorhandenen Daten zusammengeführt. Dadurch kommt es möglicherweise zu ungültigen Anteilseinträgen. Wenn eine Entity in mehr als einem Zweig einer Hierarchie vorhanden ist, können die importierten Anteilsdaten dazu führen, dass der kombinierte Anteilsprozentsatz der Entity 100 % überschreitet. Sie müssen den Anteilsprozentsatz manuell korrigieren, um sicherzustellen, dass er nicht 100 % überschreitet.

Gilt für

Financial Consolidation and Close und Tax Reporting.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer
Benutzer benötigen Schreibzugriff für die Entity.

Verwendung

```
epmautomate importOwnershipData Scenario Year Period FILE_NAME wobei:
```

- `Scenario` das Szenario ist, in das Daten exportiert werden sollen.
- `Year` das Jahr ist, in das Daten exportiert werden sollen.
- `Period` die Periode des Jahres ist, in die Anteilsdaten exportiert werden sollen.
- `FILE_NAME` der Name der CSV-Datei ist, aus der Daten importiert werden sollen.

Beispiel

```
epmautomate importOwnershipData FCCS_TotalActual FY19 Jan importfile.csv
```

importPreMappedBalances

Importiert vorkonfigurierte Saldendaten aus einer Datei in das Account Reconciliation-Repository.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

```
epmautomate importPreMappedBalances PERIOD FILE_NAME BALANCE_TYPE  
CURRENCY_BUCKET wobei:
```

- *PERIOD* der Name einer Periode ist.
- *FILE_NAME* der Name der CSV-Datei ist, die die zu importierenden Daten enthält.
- *BALANCE_TYPE* SRC oder SUB lautet.
- *CURRENCY_BUCKET* Entered, Functional oder Reporting lautet.

Beispiel

```
epmautomate importPreMappedBalances "January 2015" dailydata.csv SRC Reporting
```

importPreMappedTransactions

Importiert vorab zugeordnete Transaktionen aus einer CSV-Datei in das Account Reconciliation-Repository.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

```
epmautomate importPreMappedTransactions PERIOD TRANSACTION_TYPE FILE_NAME  
DATE_FORMAT wobei Folgendes gilt:
```

- *PERIOD* der Name einer Periode ist.
- *TRANSACTION_TYPE* enthält einen der folgenden Werte:
 - BEX zum Laden von Saldenklärungen
 - SRC zum Laden von Korrekturen am Quellsystem
 - SUB zum Laden von Korrekturen am Untersystem
 - VEX zum Laden von Erläuterungen zu Abweichungsanalysen
- *FILE_NAME* ist der Name der CSV-Datei, aus der Daten importiert werden sollen.
- *DATE_FORMAT* ist die Textzeichenfolge für ein Datumsformat, z.B. MMM d, yyyy.

Beispiel

```
epmautomate importPreMappedTransactions "January 2015" "BEX" transactions.csv  
"MMM d, yyyy"
```

importProfiles

Importiert neue Profildefinitionen aus einer CSV-Datei in das Account Reconciliation-Repository.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

`epmautomate importProfiles FILE_NAME PROFILE_TYPE METHOD DATE_FORMAT` wobei:

- `FILE_NAME` ist der Name der CSV-Datei, aus der Daten importiert werden sollen.
- `PROFILE_TYPE` entweder `profiles` oder `children` lautet.
- `METHOD` entweder `Replace` oder `Update` lautet.
- `DATE_FORMAT` die Textzeichenfolge für ein Datumsformat ist, z.B. `MMM d, yyyy`.

Beispiel

```
epmautomate importProfiles NewRecProfiles.csv Profiles Replace "MMM d,
YYYY"
```

importRates

Importiert Währungskurse aus einer CSV-Datei in das Account Reconciliation-Repository.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

`epmautomate importRates PERIOD RATE_TYPE REPLACE_MODE FILE_NAME` wobei:

- `PERIOD` der Name einer Periode ist.
- `RATE_TYPE` ein vordefinierter Kurstyp ist.
- `REPLACEMENT_MODE` `Replace` oder `ReplaceAll` lautet.
- `FILE_NAME` der Name der CSV-Datei ist, aus der die Kurse importiert werden sollen.

Beispiel

```
epmautomate importRates "January 2015" Actual ReplaceAll avgrates.csv
```

importRCAttributeValues

Importiert Attributwerte in die Listen- oder Gruppenattribute für die Account Reconciliation-Abstimmungscompliance.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser

Poweruser benötigen möglicherweise zusätzliche Sicherheit, die über ACLs bereitgestellt wird.

Verwendung

```
epmautomate importRCAttributeValues ATTRIBUTE_NAME FILE_NAME [METHOD=REPLACE|REPLACE ALL|UPDATE] [DATEFORMAT=DD/MM/YYYY|DD-MMM-YYYY|MMM d,yyyy|All], wobei Folgendes gilt:
```

- *ATTRIBUTE_NAME* ist der Name eines Listen- oder Gruppenattributs, in das die Werte importiert werden sollen.
- *FILE_NAME* ist eine CSV-Importdatei, aus der die Werte importiert werden sollen. Verwenden Sie den Befehl [uploadFile](#), um diese Datei in die Umgebung hochzuladen, bevor Sie diesen Befehl ausführen.
- *METHOD* (optional) gibt an, wie die Werte importiert werden sollen. Gültige Werte:
 - Mit `Replace` werden bei der Abstimmungscompliance alle Werte aus der Importdatei als Attributwerte hinzugefügt. Vorhandene Attributwerte werden durch die Werte aus der Importdatei ersetzt. Werte, die nicht im Attribut, jedoch in der Importdatei vorhanden sind, werden hinzugefügt. Werte, die im Attribut, jedoch nicht in der Importdatei vorhanden sind, werden nicht geändert. Beachten Sie, dass alle Attributdaten für einen bestimmten Schlüsselwert durch den Inhalt aus der Datei ersetzt oder gelöscht werden. Neue Werte werden in der Reihenfolge, in der sie in der Datei vorkommen, unten hinzugefügt. Dieser Importtyp ist am nützlichsten, wenn Sie nur die neuesten Änderungen aus einem Quellsystem verschieben, z.B. wenn Sie neue Speicherdaten aus einer Anschaffung hinzufügen, um nur bestimmte Attributwerte, falls vorhanden, durch die Werte aus der Importdatei zu ersetzen. Dies ist die Standardeinstellung.
 - Mit `Replace All` wird der vorhandene Attributwert durch die Werte aus dem Import ersetzt. Werte, die im Attribut, jedoch nicht in der Importdatei vorhanden sind, werden gelöscht. Dieser Importtyp ist am nützlichsten, um Werte aus einem Quellsystem mit einer vollständigen Aktualisierung zu spiegeln, z.B., um wöchentliche Aktualisierungen vorzunehmen, um eine Synchronisierung mit Speicherdaten aus dem Quellsystem auszuführen.
 - Mit `Update` werden alle Werte aus der Importdatei im Attribut ersetzt oder hinzugefügt. Die vorhandenen Attributwerte werden durch die Werte aus der Importdatei ersetzt. Werte, die in der Importdatei, jedoch nicht im Attribut vorhanden sind, werden hinzugefügt. Werte, die im Attribut, jedoch nicht in der Importdatei vorhanden sind, werden nicht geändert. Nur die Attributdaten für einen bestimmten Schlüsselwert werden durch den Inhalt aus der Datei ersetzt. Daten für Attribute, die

in der Datei nicht verfügbar sind, bleiben unverändert. Schlüssel, die in der Importdatei, jedoch nicht im Attribut vorhanden sind, führen zu einem Fehler. Dieser Importtyp eignet sich am besten, um einige Attribute über alle Attributwerte hinweg zu aktualisieren, z.B. um die Shopmanager nach einer Reorganisation auf den aktuellen Stand zu bringen, ohne dabei die übrigen Shopdaten zu beeinträchtigen.

- **Dateformat (optional)** gibt die gültigen Datumsformate an, die geparkt werden sollen. (Beispiel: DD/MM/YYYY, DD-**MMM**-YYYY (Standard), **MMM** d, yyyy und All). Sie können mehrere Datumsformatwerte angeben und diese durch ein Semikolon voneinander trennen.

Beispiel

```
epmautomate importRCAttributeValues Stores StoreData.csv METHOD=Replace
DATEFORMAT="All"
```

importReconciliationAttributes

Importiert Abstimmungen in vorhandene Abstimmungen aus einer Datei, die Sie mit dem Befehl [uploadFile](#) in die Account Reconciliation-Umgebung hochgeladen haben.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

```
epmautomate importReconciliationAttributes FILE.CSV Period
[Rules=RULE_NAME] [Reopen=true|false] [Dateformat=DATE_FORMAT], wobei
Folgendes gilt:
```

- **FILE** ist der Name der CSV-Datei, die die Abstimmungsattribute enthält, die Sie in die Abstimmungen importieren möchten.
- **Period** gibt die Periode an, der die Abstimmungen zugeordnet sind.
- **Rules (optional)** gibt die Regeln an, die für betroffene Abstimmungen nach dem Import der Attribute ausgeführt werden sollen. Mehrere Regelnamen können durch ein Komma getrennt werden. Gültige Werte sind:
 - **None**: Führt keine Regeln für betroffene Abstimmungen aus. Dies ist der Standardwert. Er darf nicht mit anderen Werten kombiniert werden.
 - **ALL**: Führt alle Regeln aus, die für die Abstimmungen in der angegebenen Periode definiert sind. Dieser Wert muss alleine verwendet werden. Er kann nicht mit anderen Regelnamen kombiniert werden.
 - **SET_ATTR_VAL**: Führt die vordefinierte Regel zum Festlegen des Attributwerts aus.
 - **CRT_ALT**: Führt die vordefinierte Regel zum Erstellen des Alerts aus.

- `AUTO_APP`: Führt die vordefinierte Regel zum automatischen Genehmigen der Abstimmung aus.
- `AUTO_SUB`: Führt die vordefinierte Regel zum automatischen Weiterleiten der Abstimmung aus.
- `EMAIL_ON_SAVE`: Führt die vordefinierte Regel zum automatischen Senden einer E-Mail nach dem Aktualisieren der Abstimmung aus.
- `Reopen` (optional) gibt an, ob die geänderten Abstimmungen nach Fertigstellung des Importvorgangs erneut geöffnet werden sollen. Der Standardwert ist `false`.
- `Dateformat` (optional) gibt die gültigen Datumsformate an, die geparkt werden sollen. (Beispiel: `MM-dd-yyyy`, `dd-MMMM-yy`, `MMM d` und `yyyy`.) Sie können mehrere Datumsformatwerte angeben und diese durch ein Semikolon voneinander trennen.

Beispiele

- **So importieren Sie Attribute für eine Periode und führen mehrere Regeln mit vielen Datumsformaten aus:**

```
epmAutomate importReconciliationAttributes Reconciliations.csv "July 2020"
Rules=SET_ATTR_VAL,CRT_ALT,AUTO_APP,AUTO_SUB" Reopen=true "Dateformat=MM-dd-
yyyy;dd-MMM-yy;MMM d, yyyy"
```
- **So importieren Sie Attributwerte für eine Periode, ohne Regeln auszuführen:**

```
epmAutomate importReconciliationAttributes Reconciliations.csv "July 2020"
```
- **So importieren Sie Attributwerte für eine Periode, führen alle anwendbaren Regeln aus und öffnen betroffene Abstimmungen erneut:**

```
epmAutomate importReconciliationAttributes Reconciliations.csv "July 2020"
Rules=ALL Reopen=true
```

importSnapshot

Importiert die Inhalte eines Snapshots in die Serviceumgebung. Die importierten Snapshots müssen am Standardspeicherort zum Hochladen verfügbar sein.

Verwenden Sie den Befehl [uploadFile](#), um einen Snapshot hochzuladen, oder den Befehl [copySnapshotFromInstance](#), um ihn aus einer anderen Instanz zu kopieren.

- Folgende Daten sind nicht Teil der Anwendungs-Snapshots für Planning, Planning Modules und FreeForm:
 - Auditdaten
 - Daten der Jobkonsole

Verwenden Sie den Befehl [cloneEnvironment](#) oder das Feature "Umgebung klonen", wenn Sie Audit- und Jobkonsolendaten in die Zielumgebung kopieren möchten. Der Snapshot-Import kann unter Umständen nicht erfolgreich verlaufen, wenn der Planning-Geschäftsprozess ein umbenanntes vordefiniertes Periodenelement enthält, das durch ein benutzerdefiniertes Periodenelement ersetzt wurde. Beispiel: Sie haben das vordefinierte Periodenelement `YearTotal` in `unused_YearTotal` umbenannt und dann ein Periodenelement mit einem anderen Typ mit dem ursprünglichen vordefinierten Elementnamen (in diesem Beispiel `YearTotal`) hinzugefügt. In diesem Szenario kann es vorkommen, dass der Import des Snapshots nicht erfolgreich verläuft.

- Snapshots enthalten nicht die Staging-Tabellendaten für die Komponente Datenmanagement. Um diese Daten zu importieren, verwenden Sie die Befehle [exportDataManagement](#) und [importDataManagement](#) oder die Schnittstelle für

Systemwartungsskripte der Komponente Datenmanagement. Sie können den Befehl `cloneEnvironment` oder das Feature "Umgebung klonen" verwenden, um eine identische Kopie der Umgebung zu erstellen, einschließlich der Staging-Tabellendaten für die Komponente Datenmanagement.

Die Aktivitäten, die Sie mit diesem Befehl ausführen können, hängen von Ihrer Rolle ab.

- Serviceadministratoren können nur Anwendungsartefakte in eine Umgebung importieren.
- Sie benötigen die Rolle Serviceadministrator und die Rolle Identitätsdomainadministrator, um Anwendungsinhalt in die Serviceumgebung und Identitätsdomainartefakte (Benutzer und deren vordefinierte Rollenzuweisungen) in die Identitätsdomain der Umgebung zu importieren.
Wenn ein Benutzer, der nicht in der Identitätsdomain enthalten ist, in dem Snapshot referenziert wird, der importiert wird, erstellt EPM Automate einen Benutzer in der Identitätsdomain und weist das Standardkennwort zu, das Sie in dem Befehl angeben. Alternativ wird jedem Benutzer ein temporäres, eindeutiges Kennwort zugewiesen, wenn Sie in dem Befehl kein Kennwort angeben. Standardmäßig muss der Benutzer das Kennwort bei der ersten Anmeldung neu festlegen.

 **Hinweis:**

- Geschäftsprozesse außer Account Reconciliation, Profitability and Cost Management und Oracle Enterprise Data Management Cloud: Beim Laden von Metadaten kann Oracle Enterprise Performance Management Cloud mehrere Ladedurchgänge vornehmen, wenn der vorherige Versuch zu abgelehnten Datensätzen geführt hat, da gemeinsame Elemente vor Basiselementen in der Modellstruktur stehen. Diese Versuche können die Befehlsverarbeitungszeit verlängern.
- Benutzer, die Mitglieder von Gruppen in Zugriffskontrolle sind, müssen einer vordefinierten Rolle zugewiesen sein. Der Versuch, einen Benutzer, der keiner vordefinierten Rolle zugewiesen ist, einer Gruppe zuzuweisen, ist nicht zulässig.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, die der Anwendungsrolle Migrationsadministrator zugewiesen sind

Die Rolle Identitätsdomainadministrator ist zum Importieren von Benutzern und Zuweisungen von vordefinierten Rollen erforderlich.

Verwendung

```
epmautomate importSnapshot SNAPSHOT_NAME [importUsers=true|false]
[userPassword=DEFAULT_PASSWORD] [resetPassword=true|false], wobei Folgendes gilt:
```

- `SNAPSHOT_NAME` ist der Name eines Snapshots am Standardspeicherort zum Hochladen.
- `importUsers` (optional) gibt an, ob Benutzer und die zugehörigen vordefinierten Rollenzuweisungen aus dem Snapshot importiert werden sollen. Der Standardwert ist `false`. Mit `importUsers=true` können Sie Benutzer und vordefinierte Rollenzuweisungen in die Identitätsdomain importieren, wenn der Quell-Snapshot Daten zu neuen Benutzern enthält oder wenn aktuellen Benutzern neue Rollen zugewiesen wurden. Bei Benutzeranmeldewerten wird nicht zwischen Groß- und Kleinschreibung unterschieden. Beispiel: Der Benutzeranmeldewert `jane.doe@example.com` wird genauso wie `Jane.Doe@example.com` oder jede andere Variation der Groß-/Kleinschreibung behandelt. Wenn eine Variation mit einer vorhandenen Benutzeranmeldung in der Identitätsdomain übereinstimmt, wird der Benutzer mit diesem Befehl nicht aus dem Snapshot importiert.

Hinweis:

- Der Import von Benutzern und ihren vordefinierten Rollen verläuft nicht erfolgreich, wenn ein Benutzer, der kein Identitätsdomainadministrator ist, den Importvorgang ausführt. Ein Fehler wie der Folgende wird im Migrationsstatusbericht aufgezeichnet: `Externes Verzeichnisartefakt ARTIFACT_NAME konnte nicht exportiert werden. Benutzer USER_NAME ist nicht zur Ausführung dieses Vorgangs autorisiert. Der Benutzer muss über die Rolle "Identitätsdomainadministrator" verfügen, um diesen Vorgang auszuführen.`
- Wenn Sie keine Benutzer importieren, und ein Benutzer im Quell-Snapshot nicht zu einer vordefinierten Rolle in der Zielumgebung zugewiesen ist, wird ein Fehler (EPMIE-00070: Benutzer konnte beim Import von zugewiesenen Rollen nicht gefunden werden) angezeigt.
- Änderungen an den vordefinierten Rollen des Benutzers werden basierend auf den im Quell-Snapshot zugewiesenen Rollen aktualisiert. Rollenzuweisungen im Ziel werden jedoch nicht entfernt, damit sie denen im Quell-Snapshot entsprechen. Beispiel: `jdoe` ist der vordefinierten Rolle `Poweruser` in der Zielumgebung zugewiesen, hat aber nur die Rolle `Benutzer` im Quell-Snapshot. In dieser Situation weist dieser Befehl `jdoe` der Rolle `Benutzer` zu und entfernt nicht die Zuweisung der Rolle `Poweruser` in der Zielumgebung.
- Mit diesem Befehl werden Benutzer, die nicht im Quell-Snapshot vorhanden sind, nicht aus der Zielumgebung gelöscht. Beispiel: `jdoe` hat einen Account in der Zielumgebung, der aber nicht im Quell-Snapshot vorhanden ist. In dieser Situation wird der Account von `jdoe` nicht aus der Zielumgebung gelöscht.
- Mit diesem Befehl werden Benutzer hinzugefügt, die nicht in der Zielumgebung vorhanden sind. Aktuelle Benutzereigenschaften werden nicht in der Zielumgebung aktualisiert, selbst wenn sie sich von denen im Quell-Snapshot unterscheiden.

Beispiel: Der Nachname von `jdoe` ist im Quell-Snapshot anders geschrieben als in der Zielumgebung. Diese Änderung wird nicht in der Zielumgebung vorgenommen.

- Mit diesem Befehl werden Benutzerkennwörter in der Zielumgebung nicht geändert, selbst wenn sie im Quell-Snapshot anders lauten.
- `userPassword` (optional) gibt das Standardkennwort an, um neue Benutzer zuzuweisen, die in der Identitätsdomain erstellt werden. Das angegebene Kennwort muss die Mindestanforderungen für Kennwörter erfüllen. Wenn Sie für diesen Parameter keinen Wert angeben, wird jedem Benutzer ein eindeutiges temporäres Kennwort zugewiesen.
- `resetPassword` (optional) gibt an, ob der neue Benutzer das Kennwort bei der ersten Anmeldung ändern muss. Der Standardwert ist `true`. Wenn dieser Wert angegeben ist, müssen neue Benutzer das Kennwort bei der ersten Anmeldung ändern. Wenn Sie diesen Wert auf `true` setzen, erhalten neue Benutzer E-Mails zur Accountaktivierung, in denen sie aufgefordert werden, das Kennwort zu ändern.

Beispiele

- So importieren Sie ausschließlich Anwendungsartefakte: `epmautomate importSnapshot April16FullApp`
- So importieren Sie Anwendungs- und Identitätsdomainartefakte (hierfür sind die Rollen `Serviceadministrator` und `Identitätsdomainadministrator` erforderlich):
 - So weisen Sie jedem neuen Benutzer ein eindeutiges temporäres Kennwort zu und erzwingen, dass die Benutzer ihr Kennwort nach der ersten Anmeldung neu festlegen:


```
epmautomate importSnapshot April16FullApp importUsers=true
```
 - So weisen Sie ein spezifisches Kennwort zu und verhindern, dass Benutzer dieses bei Bedarf ändern. Nicht für Importvorgänge in Produktionsumgebungen empfohlen:


```
epmautomate importSnapshot April16FullApp importUsers=true userPassword=P@ssw0rd1 resetPassword=false
```

importSupplementalCollectionData

Importiert Supplemental Data einer Collection aus einer Datei in die Anwendung

Mit dem Befehl `uploadFile` können Sie die Datei hochladen, die die Daten für den Standardpeicherort zum Hochladen enthält. Die Importdatei weist folgendes Format auf:

```
#Workflow
Workflow_Dimension_1_Name,Workflow_Dimension_2_Name,Workflow_Dimension_
n_Name
Workflow_Dimension_1_Member,Workflow_Dimension_2_Member,Workflow_Dimens
ion_n_Member
#Collection
Collection_Attribute_1,Collection_Attribute_2,Collection_Attribute_n
Record1_Attr_Value_1,Record1_Attr_Value_2, Record1_Attr_Value_n
```

Beispiel:

```
#Workflow
Entity
9100
#Collection
Custody Account Code,Trade Currency Code,Account Description,Base Currency
Code,CIC Code,IFRS 13 Tier,SII Portfolio Type,WPM Detailed NAV ID,WPM Asset
Description
1,,,,111,,,,6
```

Gilt für

Financial Consolidation and Close, und Tax Reporting.

Erforderliche Rollen

Serviceadministrator

Verwendung



Hinweis:

Alle Befehlsparameter müssen in doppelten Anführungszeichen stehen.

```
epmautomate importSupplementalCollectionData "FILE_NAME""COLLECTION_NAME" "YEAR"
"PERIOD" "[FREQUENCY_DIMENSION=MEMBER]", wobei:
```

- **FILE_NAME** der Name einer CSV-Datei ist, die in dem Standardspeicherort zum Hochladen verfügbar ist, der ordnungsgemäß formatierte Supplemental Data enthält.
- **COLLECTION_NAME** der Name der Collection ist, in die die Supplemental Data in der Datei importiert werden sollen.
- **YEAR** das Element der Year-Dimension ist, das für die Collection verwendet werden soll.
- **PERIOD** der Name der Period-Dimension ist, die für die Collection verwendet werden soll.
- **FREQUENCY_DIMENSION** der optionale Name der Frequency-Dimension ist, die für die Collection verwendet werden soll. Sie können beliebig viele Frequency-Dimensionen im Format "**FREQUENCY_DIMENSION1=MEMBER**" "**FREQUENCY_DIMENSION2=MEMBER**" angeben.

Beispiel

```
epmautomate importSupplementalCollectionData "datafile.csv" "Journal Data
Collection" "FY20" "Jan" "Account=PAYROLL" "JournalID=LNR 113"
```

importSupplementalData

Importiert Supplemental Data aus einer Datei in die Anwendung.

Verwenden Sie den Befehl [uploadFile](#), um die Datei mit den Daten in den Standardspeicherort zum Hochladen hochzuladen.

Gilt für

Financial Consolidation and Close und Tax Reporting.

Erforderliche Rollen

Serviceadministrator

Verwendung

Hinweis:

Alle Befehlsparameter müssen in doppelten Anführungszeichen stehen.

```
epmautomate importSupplementalData "FILE_NAME" "DATA_SET_NAME" "YEAR"  
"PERIOD_NAME" "SCENARIO_NAME", wobei:
```

- `FILE_NAME` der Name einer CSV-Datei ist, die in dem Standardspeicherort zum Hochladen verfügbar ist, der ordnungsgemäß formatierte Supplemental Data enthält.
- `DATA_SET_NAME` ist der Name des Datasets, in das die Supplemental Data in der Datei importiert werden sollen.
- `YEAR` ist das Jahr, für das das Dataset bereitgestellt wird.
- `PERIOD_NAME` ist der Name der Periode, in der das Dataset bereitgestellt wird.
- `SCENARIO_NAME` ist der Name des Szenarios, in dem das Dataset bereitgestellt wird.

Beispiel

```
epmautomate importSupplementalData "DatasetImport.csv" "EmployeeDataSet"  
"FY17" "Jan" "Actual"
```

importTemplate

Erstellt eine Anwendungsstruktur durch Import aus einer Vorlagendatei, die in profitinbox vorhanden ist.

Sie können mit dem Befehl [uploadFile](#) eine Vorlagendatei in profitinbox hochladen.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

```
epmautomate importTemplate APPLICATION_NAME File_Name  
isApplicationOverwrite=true|false wobei:
```

- *APPLICATION_NAME* der Name der Profitability and Cost Management-Anwendung ist, die durch Import der Vorlage erstellt werden soll.
- *File_Name* der Name der ZIP-Datei ist, die die Anwendungsvorlage enthält. Diese Datei muss in profitinbox vorhanden sein.
- *isApplicationOverwrite* angibt, ob die vorhandene Anwendung überschrieben werden soll, sofern vorhanden. Geben Sie diesen Parameterwert in Kleinbuchstaben an.

Beispiel

```
epmautomate importTemplate BksML12 template1.zip isApplicationOverwrite=true
```

importTMAtributeValues

Importiert Werte in die Account Reconciliation-Gruppenattribute für den Transaktionsabgleich.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser

Poweruser benötigen möglicherweise zusätzliche Sicherheit, die über ACLs bereitgestellt wird.

Verwendung

```
epmautomate importTMAtributeValues ATTRIBUTE_NAME FILE_NAME [METHOD=REPLACE|REPLACE ALL|UPDATE] [DATEFORMAT=DD/MM/YYYY|DD-MMM-YYYY|MMM d,yyyy|All], wobei Folgendes gilt:
```

- *ATTRIBUTE_NAME* ist der Name eines Gruppenattributs, in das die Werte importiert werden sollen.
- *FILE_NAME* ist eine CSV-Importdatei, aus der die Werte in den Transaktionsabgleich importiert werden sollen. Verwenden Sie den Befehl [uploadFile](#), um diese Datei in die Umgebung hochzuladen, bevor Sie diesen Befehl ausführen.
- *METHOD* (optional) gibt an, wie die Werte importiert werden sollen. Gültige Werte:
 - Mit `Replace` werden alle Werte aus der Importdatei in Gruppenattribute für den Transaktionsabgleich hinzugefügt. Vorhandene Attributwerte werden durch die Werte aus der Importdatei ersetzt. Werte, die nicht im Attribut, jedoch in der Importdatei vorhanden sind, werden hinzugefügt. Werte, die im Attribut, jedoch nicht in der Importdatei vorhanden sind, werden nicht geändert. Beachten Sie, dass alle Attributdaten für einen bestimmten Schlüsselwert durch den Inhalt aus der Datei ersetzt oder gelöscht werden. Neue Werte werden in der Reihenfolge, in der sie in der Datei vorkommen, unten hinzugefügt. Dieser Importtyp ist am nützlichsten, wenn Sie nur die neuesten Änderungen aus einem Quellsystem verschieben, z.B. wenn Sie neue Speicherdaten aus einer Anschaffung hinzufügen, um nur bestimmte Attributwerte, falls vorhanden, durch die Werte aus der Importdatei zu ersetzen. Dies ist die Standardeinstellung.
 - Mit `Replace All` wird der vorhandene Attributwert durch die Werte aus dem Import ersetzt. Werte, die im Attribut, jedoch nicht in der Importdatei vorhanden sind, werden gelöscht.

Dieser Importtyp ist am nützlichsten, um Werte aus einem Quellsystem mit einer vollständigen Aktualisierung zu spiegeln, z.B., um wöchentliche Aktualisierungen vorzunehmen, um eine Synchronisierung mit Speicherdaten aus dem Quellsystem auszuführen.

- Mit `Update` werden alle Werte aus der Importdatei im Attribut ersetzt oder hinzugefügt. Die vorhandenen Attributwerte werden durch die Werte aus der Importdatei ersetzt. Werte, die in der Importdatei, jedoch nicht im Attribut vorhanden sind, werden hinzugefügt. Werte, die im Attribut, jedoch nicht in der Importdatei vorhanden sind, werden nicht geändert. Nur die Attributdaten für einen bestimmten Schlüsselwert werden durch den Inhalt aus der Datei ersetzt. Daten für Attribute, die in der Datei nicht verfügbar sind, bleiben unverändert. Schlüssel, die in der Importdatei, jedoch nicht im Attribut vorhanden sind, führen zu einem Fehler.
Dieser Importtyp eignet sich am besten, um einige Attribute über alle Attributwerte hinweg zu aktualisieren, z.B. um die Shopmanager nach einer Reorganisation auf den aktuellen Stand zu bringen, ohne dabei die übrigen Shopdaten zu beeinträchtigen.
- `Dateformat (optional)` gibt die gültigen Datumsformate an, die geparkt werden sollen. (Beispiel: `DD/MM/YYYY`, `DD-MMM-YYYY` (Standard), `MMM d, yyyy` und `All`). Sie können mehrere Datumsformatwerte angeben und diese durch ein Semikolon voneinander trennen.

Beispiel

```
epmautomate importTMAttributeValues TMGA TMGA.csv METHOD=Replace  
DATEFORMAT="All"
```

importTmPremappedTransactions

Importiert für eine bestimmte Datenquelle vorab zugeordnete Transaktionsdaten aus einer Datei im Account Reconciliation-Repository im Modul "Transaktionsabgleich".

Verwenden Sie den Befehl [uploadFile](#), um die Transaktionsdatei in den Service hochzuladen.

Dieser Befehl zeigt den Importstatus und den Namen einer Importlogdatei in der Konsole an. Verwenden Sie den Befehl [downloadFile](#), um die Logdatei auf einen lokalen Computer herunterzuladen.

Weitere Informationen zu den Anforderungen für das Importdateiformat und Informationen zum Importieren von Daten finden Sie unter [Daten importieren](#) in der Dokumentation *Konten mit Account Reconciliation abstimmen*.

 **Hinweis:**

- Sie können Transaktionen jeweils nur für einen Abgleichstypen importieren. Jedoch können parallele Importe in verschiedene Abgleichstypen ausgeführt werden.
- Im Gegensatz zum Fenster "Jobs" können Sie vorab zugeordnete Transaktionsdaten immer nur aus jeweils einer Datei importieren.
- Nachdem Sie vorab zugeordnete Transaktionen für alle Datenquellen importiert haben, führen Sie den `runautomatch`-Befehl aus.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

`epmautomate importTmPremappedTransactions MATCH_TYPE DATA_SOURCE FILE_NAME [DATE_FORMAT]` wobei Folgendes gilt:

- `MATCH_TYPE` ist ein in Account Reconciliation definierter Abgleichstyp.
- `DATA_SOURCE` ist die ID der Datenquelle, die mit dem angegebenen Abstimmungstyp verknüpft ist.
- `FILE_NAME` ist der Name der CSV-Datei, die die zu importierenden Transaktionen enthält. Diese Datei muss in dem Service verfügbar sein.
- `DATE_FORMAT` ist ein optionaler Parameter, der das Format der Datumsfelder angibt, die in der Transaktionsimportdatei enthalten sind. Der Standardwert ist `dd-MMM-YYYY`. Andere unterstützte Datenformate sind: `MM/dd/yyyy`, `dd/MM/yyyy`, `MM-dd-yyyy`, `d-M-yyyy` und `MMM d. yyyy`.

Beispiel

```
epmautomate importTmPremappedTransactions "INTERCOMPANY" "AP" dailydata.csv d-M-YYYY
```

importValidIntersections

Importiert gültige Schnittmengengruppen aus einer ZIP-Datei, die eine Excel-Datei mit gültigen Schnittmengendefinitionen enthält, in den Geschäftsprozess. Bevor Sie diesen Befehl ausführen, müssen Sie mit dem Befehl [uploadFile](#) die Importdatei in die Umgebung hochladen.

Für einen erfolgreichen Import gültiger Schnittmengen muss Ihre importierte ZIP-Datei eine Excel-Datei mit zwei Arbeitsblättern ("Rules" und "Sub Rules") enthalten. Das erste Arbeitsblatt, "Rules", muss die Schnittmengengruppe, einschließlich Dimensionen, und Eigenschaften wie zum Beispiel "Unspecified Valid" (Nicht angegebene Elemente gültig) und

"Additional Dims Required" (Zusätzliche Dimensionen erforderlich) enthalten. Das zweite Arbeitsblatt, "Sub Rules", muss Elementauswahlen und Ausschlüsse enthalten. Weitere Informationen finden Sie in den folgenden Themen in der Dokumentation *Planning verwalten*.

- Ankerdimensionen und nicht verankerte Dimensionen
- Beispiele für gültige Schnittmengen

Die beste Methode zum Abrufen der Formatvorlage für die Importdatei ist, die gültigen Schnittmengen aus der Anwendung zu exportieren. In den folgenden Abbildungen ist ein Beispielformat dargestellt.

	A	B	C	D	E	F	G	H	I	J
	Name	Position	Description	Enabled	Anchor Dim Name	Anchor Dimension	Dim1	Dim1 Required	Dim2	Dim2 Required
1										
2	Region - Product	1		true	Entity	true	Product	false		
3	Region-Product-VI-Copy	2		true	Entity	true	Product	false		
4										
5										
6										

	A	B	C	D	E	F	G
	Name	Anchor Members	Anchor Exclusion	Dim1 Members	Dim1 Exclusion	Dim2 Members	Dim2 Exclusion
1							
2	Region - Product	Children(403)	410,421	IDescendants(P_TP)			
3	Region - Product	410		IDescendants(P_TP)	P_260,P_270,P_280		
4	Region - Product	421		IDescendants(P_TP)	P_220,P_250		
5	Region-Product-VI-Copy	Children(403)	410,421	IDescendants(P_TP)			
6	Region-Product-VI-Copy	410		IDescendants(P_TP)	P_260,P_270,P_280		
7	Region-Product-VI-Copy	421		IDescendants(P_TP)	P_220,P_250		
8							

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate importValidIntersections FILE_NAME.zip
[ErrorFile=ERROR_FILE_NAME.txt], wobei Folgendes gilt:
```

- FILE_NAME ist der Name der ZIP-Datei, die die Excel-Datei mit der gültigen Schnittmengendefinition enthält.
- ErrorFile (optional) gibt den Namen der Textdatei an, in die Fehlerdatensätze geschrieben werden. Wenn dieser Parameterwert nicht angegeben ist, erstellt

EPM Automate automatisch eine Fehlerdatei. Den Namen dieser Datei können Sie in der Jobkonsole anzeigen.

Beispiel

```
epmautomate importValidIntersections VI_Import_File.zip
ErrorFile=VI_Import_Log.txt
```

invalidLoginReport

Erstellt in OCI-Umgebungen (2. Generation) den Bericht zu ungültigen Anmeldungen, in dem die nicht erfolgreichen Versuche zur Anmeldung bei der Umgebung in einem festgelegten Zeitraum gemäß dem Aufbewahrungszeitraum für Auditdaten für Ihre Umgebung aufgelistet sind. Der Standardaufbewahrungszeitraum beträgt 30 Tage. Sie können den Zeitraum auf maximal 90 Tage verlängern, indem Sie die Einstellung für den Aufbewahrungszeitraum in Tagen in der Oracle Cloud Identity Console entsprechend anpassen. Um die Auditdaten für einen Zeitraum von mehr als 90 Tagen aufzubewahren, können Sie diesen Bericht und den [Auditbericht zur Rollenzuweisung](#) herunterladen und archivieren.

Der Bericht zu ungültigen Anmeldungen enthält beispielsweise die folgenden Informationen:

- Benutzername des Benutzers, der versucht hat, sich anzumelden
- Remote-IP-Adresse, von der aus der Benutzer versucht hat, sich anzumelden
- Zeitstempel des Anmeldeversuchs

In diesem Bericht werden alle nicht erfolgreichen Anmeldeversuche beim entsprechenden Identity Cloud Service angezeigt. Diese stehen möglicherweise nicht alle in Beziehung zu einer Oracle Enterprise Performance Management Cloud-Instanz.

	A	B	C
1	User Name	IP Address	Access Date and Time
2	john.doe@example.com	xxx.xx.xx.xx5	July 15, 2021 11:14:58 UTC
3	jane.doe@example.com	xxx.xx.xx.xx9	July 15, 2021 11:14:58 UTC
4	john.smith@example.com	xxx.xx.xx.xx3	July 15, 2021 11:14:57 UTC

Verwenden Sie den Befehl [downloadFile](#), um den Bericht auf einen lokalen Computer herunterzuladen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Identitätsdomainadministrator und alle vordefinierten Rollen (Serviceadministrator, Poweruser, Benutzer oder Leseberechtigter)

Verwendung

epmAutomate invalidLoginReport FROM_DATE TO_DATE FILE_NAME.CSV, wobei Folgendes gilt:

- *FROM_DATE* gibt das Startdatum der Periode (im Format JJJJ-MM-TT) an, für die der Bericht generiert werden soll. Dieses Datum muss innerhalb des Aufbewahrungszeitraums für Auditdaten liegen, der in der Oracle Cloud Identity-Konsole festgelegt ist.
- *TO_DATE* gibt das Enddatum der Periode (im Format JJJJ-MM-TT) an, für die der Bericht generiert werden soll.
- *FILE_NAME* ist der Name einer CSV-Datei für den Bericht.

 **Note:**

Dieser Bericht kann nur für die letzten 90 Tage generiert werden.

Beispiel

```
epmAutomate invalidLoginReport 2021-06-01 2021-06-30  
invalidLoginReport.CSV
```

listBackups

Listet die verfügbaren Backup-Snapshots von OCI (Gen 2)-Umgebungen auf, um zu bestimmen, ob ein bestimmtes Backup verfügbar ist, sodass Sie es archivieren oder zur Wiederherstellung der aktuellen Umgebung verwenden können. Dieser Befehl funktioniert nicht in Classic Oracle Enterprise Performance Management Cloud-Umgebungen.

Bevor Sie versuchen, ein bestimmtes Backup wiederherzustellen, verwenden Sie diesen Befehl, um zu prüfen, ob das erforderliche Backup in Oracle Object Storage verfügbar ist. Wenn das Backup verfügbar ist, können Sie es wiederherstellen (in Ihre Umgebung kopieren), indem Sie den Befehl [restoreBackup](#) ausführen. Nach dem Kopieren des Backups können Sie es mit dem Befehl [importSnapshot](#) importieren. Durch die Selfservice-Wiederherstellung der Umgebung können Sie Verarbeitungszeit einsparen.

Bei anderen Services als Narrative Reporting listet dieser Befehl die verfügbaren Backup-Snapshots (die vom täglichen Wartungsprozess erstellt wurden) mit der Benennungskonvention `YYYY-MM-DDTHH:MM:SS/Artifact_Snapshot.zip` auf. Beispiel: `2022-02-16T21:00:02/Artifact_Snapshot.zip`. Bei Narrative Reporting wird für verfügbare Snapshots die Benennungskonvention `YYYY-MM-DDTHH:MM:SS/EPRCS_Backup.tar.gz` verwendet. Beispiel: `2022-02-16T21:00:02/EPRCS_Backup.tar.gz`. In beiden Fällen gibt der Zeitstempel die UTC-Zeit der Snapshot-Erstellung wieder. Die folgende Darstellung enthält eine Beispielbefehlsausgabe.

```
c:\Oracle\EPM Automate\bin>epmAutomate listbackups  
  
2022-03-04T06:37:51/Artifact_Snapshot.zip  
2022-03-08T06:32:01/Artifact_Snapshot.zip  
2022-03-09T12:08:05/Artifact_Snapshot.zip  
2022-03-10T06:37:48/Artifact_Snapshot.zip  
2022-03-15T06:21:28/Artifact_Snapshot.zip  
2022-03-16T06:20:52/Artifact_Snapshot.zip  
2022-03-16T12:13:56/Artifact_Snapshot.zip  
  
Total 7
```

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, die der Anwendungsrolle Migrationsadministrator zugewiesen sind

Verwendung

```
epmAutomate listBackups
```

Beispiel

```
epmAutomate listBackups
```

listFiles

Listet die Namen der Dateien am Standardspeicherort, in den Ordnern der Komponente Datenmanagement und in profitinbox/profitoutbox (Profitability and Cost Management) auf.

Dieser Befehl listet auch inkrementelle und Backupexportdateien sowie Snapshots der Migration, Zugriffslogs und Aktivitätsberichte auf. Diese Darstellung enthält eine abgeschnittene Version der Befehlsausgabe.

```
apr/2022-01-27_05_23_36/activityreport.json
apr/2022-01-28_05_24_07/2022-01-28_05_24_07.html
apr/2022-01-28_05_24_07/access_log.zip
apr/2022-01-28_05_24_07/activityreport.json
apr/2022-01-29_05_24_06/2022-01-29_05_24_06.html
apr/2022-01-29_05_24_06/access_log.zip
outbox/Vision_99.dat
roleassign.csv
RoleAssignment.csv
sanity_no_data_22-01-18.zip
U-1.csv
U2.csv
user1.csv
user12.csv
users12.csv
Uservariables-MemberFormula.zip
UsrGrpReport.CSV
Vision_DTsetup.zip
VisionADCForms2010.zip
```

Dieser Befehl führt den aktuellen Snapshot nicht auf, wenn er ausgeführt wird, während der Snapshot der Umgebung generiert wird (z.B. während der täglichen Wartung).

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, die der Anwendungsrolle Migrationsadministrator zugewiesen sind

Verwendung

```
epmautomate listFiles
```

Beispiel

```
epmautomate listFiles
```

loadData

Lädt Daten mit einer Datei, die in profitinbox verfügbar ist, in einen Berechnungs-Cube.

Verwenden Sie den Befehl [uploadFile](#), um Dateien in profitinbox zu laden.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate loadData APPLICATION_NAME dataFileName=File_Name PARAMETER=VALUE`
wobei Folgendes gilt:

- `APPLICATION_NAME` ist der Name der Profitability and Cost Management-Anwendung, in die Sie Daten laden möchten.
- `dataFileName=File_Name` gibt eine Dataload-Datei an, die in profitinbox verfügbar ist. Der Datendateiname muss in doppelte Anführungszeichen gesetzt werden.
- `PARAMETER=VALUE` gibt Laufzeitparameter und deren Werte zum Laden von Daten an. Geben Sie so viele Parameter- und Wertpaarungen an, wie der Prozess erfordert. Gültige Parameter und deren Werte:
 - `clearAllDataFlag=true|false` gibt an, ob vorhandene Daten im Anwendungs-Cube gelöscht werden sollen.
 - `dataLoadValue=OVERWRITE_EXISTING_VALUES|ADD_TO_EXISTING` gibt an, wie vorhandene Daten behandelt werden sollen.

Beispiel

```
epmautomate loadData BksML12 dataFileName="data1.txt"clearAllDataFlag=true  
dataLoadValue="OVERWRITE_EXISTING_VALUES"
```

loadDimData

Lädt Dimensionsmetadaten von mindestens einer Datei in profitinbox in eine Anwendung.

Verwenden Sie den Befehl [uploadFile](#), um Metadatendateien in profitinbox zu laden.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate loadDimData APPLICATION_NAME dataFileName=File_Name`
[`stringDelimiter="DELIMITER"`], wobei Folgendes gilt:

- `APPLICATION_NAME` ist der Name der Profitability and Cost Management-Anwendung, in die Dimensionsmetadaten geladen werden sollen.
- `dataFileName` gibt eine Ladedatei mit Dimensionsmetadaten an, die in "profitinbox" verfügbar ist. Beim Laden von Metadaten aus mehreren Dateien müssen die Dateinamen durch ein Trennzeichen getrennt aufgelistet werden.
- `stringDelimiter` gibt das Trennzeichen an, das zum Trennen von Namen für Metadatendateien verwendet wird. Trennzeichen müssen in doppelte Anführungszeichen gesetzt werden.

Beispiel

```
epmautomate loadDimData BksML12 dataFileName="dimdata1.txt#dimdata1.txt"  
stringDelimiter="#"
```

loadViewpoint

Lädt einen Ansichtspunkt (eine Teilmenge von Knoten) aus einer Ladedatei in eine Oracle Enterprise Data Management Cloud-Anwendung.

Über Ladevorgänge von Ansichtspunkten können Sie Daten in ungebundenen, gebundenen oder teilweise gebundenen Ansichtspunkten laden. Die Ladedatei des Ansichtspunkts, eine CSV- oder Excel (XLSX)-Datei oder eine ZIP-Datei, die eine CSV- oder XLSX-Datei enthält, muss in der Umgebung verfügbar sein, in die Sie den Ansichtspunkt laden. Sie können die Ladedatei mit dem Befehl [uploadFile](#) oder [copyFileFromInstance](#) in die Umgebung hochladen.

Gilt für

Oracle Enterprise Data Management Cloud

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate loadViewpoint VIEW VIEWPOINT PURPOSE FILE_NAME  
[loadType=ReplaceNodes|Merge]
```

wobei Folgendes gilt:

- `VIEW` ist der Name einer Oracle Enterprise Data Management Cloud-Ansicht.
- `VIEWPOINT` ist der Name des Ansichtspunkts, den Sie laden möchten.
- `PURPOSE` ist eine Textzeichenfolge in doppelten Anführungszeichen, die angibt, warum der Ansichtspunkt geladen wird.
- `FILE_NAME` ist der Name der Datei einschließlich Erweiterung, aus der der Ansichtspunkt geladen werden soll.
- `loadType`, optional, gibt an, wie der Ansichtspunkt geladen werden soll. Gültige Werte sind `Zusammenführen` und `Knoten ersetzen`.
 - Verwenden Sie `Zusammenführen`, um vorhandene Beziehungen zu erhalten, indem inkrementelle Änderungen verarbeitet werden.
 - Verwenden Sie `Knoten ersetzen`, um alle Beziehungen außer den in der Ladedatei enthaltenen aus der Hierarchie zu löschen (einschließlich verwaister Beziehungen und Beziehungen, die von anderen Ansichtspunkten mit demselben Hierarchieset verwendet werden). Dies ist der Standard-Load-Typ.

Beispiele

- **Inkrementelle Änderungen zusammenführen:**

```
epmautomate loadViewpoint  
USOperations Entity "Daily Upstream Load" data_Entity.CSV  
loadType=Merge
```

- **Vorhandene Hierarchien ersetzen:** `epmautomate loadViewpoint USOperations Entity "Replace US Operations data" data_Entity.CSV`

login

Stellt eine sichere Verbindung zu einer Umgebung her. Dieser Befehl unterstützt die Anmeldung bei einer Umgebung mit einem Nur-Text-Kennwort, mit einer verschlüsselten Kennwortdatei, die das Kennwort enthält, oder mit einem OAuth 2.0-Aktualisierungstoken. Die Anmeldung über das OAuth 2.0-Aktualisierungstoken wird nur für OCI-Umgebungen (2. Generation) unterstützt.

Sie melden sich an, um eine Session zu starten, die bis zu Ihrer Abmeldung aktiv bleibt.

Hinweis:

- EPM Automate unterstützt keine Anmeldung mit den SSO-Zugangsdaten Ihrer Organisation.
- EPM Automate funktioniert nicht mit SOCKS-Proxy, nur mit HTTP/HTTPS-Proxy.
- Wenn dieser Befehl in Batchdateien zum Automatisieren von Aktivitäten verwendet wird, empfiehlt Oracle die Verwendung von verschlüsselten Kennwörtern oder OAuth 2.0-Aktualisierungstokens, um in Batchdateien die Aufzeichnung von Kennwörtern in Klartext zu vermeiden.
- Auf Windows-Computern ermittelt dieser Befehl automatisch, ob das Sicherheitszwischenzertifikat für Proxyserver fehlt. Durch sein Fehlen kann das Herstellen einer Verbindung verhindert werden, weshalb es der unter `C:\Oracle\EPM Automate` installierten JRE hinzugefügt wird. Dadurch werden Anmeldefehler in Verbindung mit Sicherheitszertifikaten bei der Verwendung von Proxyservern für den Internetzugriff verhindert. Auf Linux-Computern ermittelt der Anmeldebefehl das fehlende Sicherheitszertifikat für den Proxyserver, lädt es herunter und zeigt einen Fehler an. Benutzer mit `root`-Zugriff können dann das heruntergeladene Zertifikat in der JRE installieren, die in der Umgebungsvariable `JAVA_HOME` verfügbar ist, die in den Umgebungsvariablen ermittelt wurde. Informationen hierzu finden Sie in den folgenden Informationsquellen:
 - [Java Runtime Environment und EPM Automate](#)
 - Java-Dokumentation zu "keytool"

Wenn Sie eine ältere Version von EPM Automate verwenden, wird bei der Anmeldung eine Meldung zum Upgraden angezeigt. Sie können den Befehl [upgrade](#) verwenden, um das Upgrade Ihrer Installation im Hintergrund auszuführen.

Wenn Sie den Befehl [addUsers](#), [removeUsers](#), [assignRole](#) oder [unassignRole](#) ausführen möchten, melden Sie sich nicht mit dem OAuth-Aktualisierungstoken an. Für diese Befehle müssen Sie die Basisauthentifizierung verwenden. Alle anderen Befehle funktionieren mit OAuth 2.0 in OCI (Gen 2)-Umgebungen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Verwendung

- **Bei Verwendung eines nicht verschlüsselten Kennworts:** `epmautomate login USERNAME PASSWORD URL [IDENTITYDOMAIN]`
`[ProxyServerUserName=PROXY_USERNAME ProxyServerPassword=PROXY_PASSWORD ProxyServerDomain=PROXY_DOMAIN]] [KeystorePassword=PASSWORD]`
- **Bei Verwendung einer verschlüsselten Datei:** `epmautomate login USERNAME PASSWORD_FILE URL [IDENTITYDOMAIN]`
`[ProxyServerUserName=PROXY_USERNAME]`
`[ProxyServerPassword=PROXY_PASSWORD] [ProxyServerDomain=PROXY_DOMAIN]`
`[KeystorePassword=KEYSTORE_PASSWORD]`

In diesen Befehlen gilt Folgendes:

- `USERNAME` ist der Benutzername des Benutzers.
- `PASSWORD` ist das Kennwort des Benutzers.
- `PASSWORD_FILE` ist der Name und der Speicherort der Datei, in der das verschlüsselte Kennwort oder das OAuth 2.0-Aktualisierungstoken des Benutzers gespeichert ist. Informationen hierzu finden Sie im Abschnitt zum Befehl [encrypt](#).
- `URL` ist die URL der Umgebung, zu der eine Verbindung hergestellt werden soll. Sie können eine benutzerdefinierte oder eine Vanity-URL anstelle der Oracle Enterprise Performance Management Cloud-URL verwenden. Zum Erstellen einer Vanity-URL für eine Umgebung können Sie einen Drittanbieter zum Kürzen von Links verwenden, beispielsweise T.ly, Bitly, Rebrandly, TinyUrl, is.gd usw. Ausführliche Informationen zum Erstellen von Vanity-URLs finden Sie in der Dokumentation des Drittanbieters zum Kürzen von Links.

Hinweis:

Wenn Sie ein API-Gateway oder einen Reverse-Proxy verwenden, verwenden Sie die zugehörige URL und den für Ihre Umgebung definierten Kontext anstelle der EPM Cloud-URL.

- `IDENTITYDOMAIN` (optional) ist die Identitätsdomain der Umgebung. Dieser Wert wird automatisch von der EPM Cloud-URL abgeleitet. Alle von Ihnen angegebenen Werte werden ignoriert. Dieser Wert ist jedoch erforderlich, wenn Sie eine API-Gateway- oder Reverse-Proxy-URL für die Verbindung zu einer klassischen EPM Cloud-Umgebung verwenden.
- `ProxyServerUserName` ist der Benutzername zum Authentifizieren einer sicheren Session auf dem HTTP-Proxyserver, der den Zugriff auf das Internet steuert. Geben Sie den Benutzernamen ohne einen Domainnamen als Präfix an. Nur

erforderlich, wenn die Authentifizierung auf dem Proxyserver für Ihr Netzwerk aktiviert ist.

- `ProxyServerPassword` das Kennwort zum Authentifizieren des Benutzers beim Proxyserver. Nur erforderlich, wenn die Authentifizierung auf dem Proxyserver für Ihr Netzwerk aktiviert ist. Dieses Kennwort kann verschlüsselt werden. Informationen hierzu finden Sie im Abschnitt zum Befehl `encrypt`. Wenn dieses Kennwort verschlüsselt ist, wird es aus der Datei `PASSWORD_FILE` gelesen.
- `ProxyServerDomain` ist der Name der Domain, die für den HTTP-Proxyserver definiert ist (nicht der Servername oder der Hostname des Proxyservers). Nur erforderlich, wenn die Authentifizierung auf dem Proxyserver für Ihr Netzwerk aktiviert ist und eine Proxyserverdomain konfiguriert ist.
- `KeystorePassword` (optional) ist das Keystore-Kennwort, das für den Import des Sicherheitszertifikats für Proxyserver erforderlich ist. Verwenden Sie diesen Parameter nur unter Windows und nur, wenn Ihnen die folgenden Fehler in Umgebungen angezeigt werden, in denen der Internetzugriff über einen Proxyserver erfolgt:

EPMAT-7: Verbindung kann nicht hergestellt werden, da einige SSL-Zertifikate nicht im Keystore vorhanden sind

EPMAT-7: Verbindung kann nicht hergestellt werden, da die oben genannten SSL-Zertifikate im Keystore nicht vorhanden sind

Hinweis:

EPM Automate ermittelt und verwendet die HTTP/HTTPS-Proxyeinstellungen auf Ihrem Computer.

EPM Automate unterstützt die folgenden Authentifizierungsmechanismen für die Verbindung zum Proxyserver:

- Basisauthentifizierung
- Digest-Authentifizierung
- Kerberos-Authentifizierung
- Negotiate Proxy-Authentifizierung
- NTLM-Authentifizierung

Welche Authentifizierungsmethode in welcher Konfiguration verfügbar ist, hängt vom verwendeten Proxyserver ab.

Wenn auf Linux-Computern gemäß den Proxyeinstellungen erforderlich ist, dass Sie sich beim Proxyserver authentifizieren, müssen Sie die Proxyserverdomain, den Benutzernamen und das Kennwort als Parameter für diesen Befehl eingeben.

Wenden Sie sich an Ihren Netzwerkadministrator zur Unterstützung beim Domainnamen des Proxyservers und bei den Zugangsdaten.

Beispiele

- Nicht verschlüsseltes EPM Cloud-Kennwort verwenden, keine Proxyauthentifizierung:
`epmautomate login serviceAdmin P@ssword1 https://test-cloud-pln.pbcs.us1.oraclecloud.com`
- Verschlüsselte Datei verwenden, keine Proxyauthentifizierung:

```
epmautomate login serviceAdmin C:\mySecuredir\password.epw https://  
test-cloud-pln.pbc.s.us1.oraclecloud.com
```

- **Verschlüsselte Datei verwenden, wenn die Authentifizierung auf dem Proxyserver mit Serverdomain aktiviert ist:**

```
epmautomate login serviceAdmin C:\mySecuredir\password.epw https://  
test-cloud-pln.pbc.s.us1.oraclecloud.com  
ProxyServerUserName=john.doe@example.com ProxyServerPassword=example  
ProxyServerDomain=example
```

- **Verschlüsselte Datei verwenden, wenn die Authentifizierung auf dem Proxyserver ohne Serverdomain aktiviert ist:**

```
epmautomate login serviceAdmin C:\mySecuredir\password.epw https://  
test-cloud-pln.pbc.s.us1.oraclecloud.com  
ProxyServerUserName=john.doe@example.com ProxyServerPassword=example
```

- **Verschlüsseltes EPM Cloud- und Proxyserverkennwort verwenden, wenn die Authentifizierung auf dem Proxyserver mit Serverdomain aktiviert ist:**

```
epmautomate login serviceAdmin C:\mySecuredir\password.epw https://  
test-cloud-pln.pbc.s.us1.oraclecloud.com  
ProxyServerUserName=john.doe@example.com ProxyServerDomain=example
```

- **Verschlüsseltes EPM Cloud- und Proxyserverkennwort verwenden, wenn die Authentifizierung auf dem Proxyserver ohne Serverdomain aktiviert ist:**

```
epmautomate login serviceAdmin C:\mySecuredir\password.epw https://  
test-cloud-pln.pbc.s.us1.oraclecloud.com  
ProxyServerUserName=john.doe@example.com
```

- **Verschlüsselte Datei mit Apigee API-Gateway verwenden:**

```
epmautomate login serviceAdmin C:\mySecuredir\password.epw https://  
exampleapigee.apigee.com/epm example_ID_DOM
```

logout

Beendet Ihre aktuelle Verbindung mit einer Umgebung.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Verwendung

```
epmautomate logout
```

Beispiel

```
epmautomate logout
```

maskData

Maskiert Anwendungsdaten, um Datenschutz sicherzustellen. Verwenden Sie diesen Befehl nur in Testumgebungen, um vertrauliche Daten vor Anwendungsentwicklern zu verbergen.

WARNUNG: Verwenden Sie diesen Befehl nicht in Produktionsumgebungen, weil er aktuelle Anwendungsdaten zufällig festlegt, sodass diese bedeutungslos werden. Sie können die Auswirkungen dieses Befehls nicht rückgängig machen. Wenn Sie die Daten in einer Serviceumgebung versehentlich maskiert haben, müssen Sie die Daten aus einem Backup oder aus dem Wartungs-Snapshot wiederherstellen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate maskData [-f]`, wobei `-f` eine Option ist, mit der der Beginn des Maskierungsvorgangs ohne Benutzerbestätigung erzwungen werden kann. Wenn Sie die Option `-f` nicht verwenden, werden Sie von EPM Automate zur Bestätigung der Aktion aufgefordert.

Beispiel

```
epmautomate maskData [-f]
```

mergeDataSlices

Führt alle inkrementellen Datensegmente eines Aggregate Storage-Cubes im Hauptdatenbanksegment zusammen und entfernt optional Zellen mit einem Nullwert.

Gilt für

Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate mergeDataSlices CUBE_NAME [keepZeroCells=true|false]` wobei Folgendes gilt:

- `CUBE_NAME` gibt den Aggregate Storage-Cube an, für den alle Datensegmente zusammengeführt werden sollen.
- `keepZeroCells` gibt optional an, ob Zellen mit einem Nullwert entfernt werden sollen (das logische Löschen von Daten aus einem Bereich führt zu einer Zelle mit einem Nullwert). Der Standardwert ist `true`.

Beispiel

```
epmautomate mergeDataSlices repl keepZeroCells=false
```

mergeSlices

Führt inkrementelle Datensegmente im Haupt-Cube der Datenbank zusammen und entfernt optional die Oracle Essbase-Zellen, die 0 (Null) als Wert enthalten, um den Cube kompakt zu machen.

Durch das Entfernen von Zellen, die 0 enthalten, wird die Cube-Performance optimiert.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

```
epmautomate mergeSlices applicationName [removeZeroCells=true|false] wobei:
```

- `applicationName` der Name einer Profitability and Cost Management-Anwendung ist.
- `removeZeroCells` (optional) angibt, ob Zellen, die 0 enthalten, entfernt werden sollen. Der Standardwert dieses Parameters ist `false`.

Beispiele

- Segmente zusammenführen, ohne Zellen zu entfernen, die 0 enthalten:
 - `epmautomate mergeSlices BksML30`
 - `epmautomate mergeSlices BksML30 removeZeroCells=false`
- Segmente zusammenführen und Zellen entfernen, die Nullen enthalten:
`epmautomate mergeSlices BksML30 removeZeroCells=true`

optimizeASOCube

Optimiert die Performance von Abfragen für die Auswahl von Aggregatsichten für die Datenextraktion aus ASO-Cubes.

Mit diesem Befehl können Sie Vorgänge zur Abfrageoptimierung bei ASO-Cubes in den Fällen durchführen, in denen die Standardaggregation vermutlich nicht ausreicht, um Ihre Anforderungen an Datenextraktion oder Reporting aufgrund großer Datenmengen zu erfüllen. Typischer Optimierungsprozess:

- Verwerfen Sie standardmäßige und abfragebasierte Aggregationen.
- Starten Sie das Abfrage-Tracking.
- Führen Sie Beispielabfragen über den Profitability and Cost Management-Abfragemanager, Oracle Smart View for Office oder das Datenmanagement sowie alle anderen MDX-Abfragen aus, die repräsentativ für die Art von Abfragen sind, für die die Optimierung in Oracle Essbase trainiert werden soll.

- Erstellen Sie die Aggregation basierend auf optimierten oder Standardabfragen.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate optimizeASOCube APPLICATION_NAME OPTIMIZATION_TYPE`, wobei Folgendes gilt:

- `APPLICATION_NAME` ist der Name der Profitability and Cost Management-Anwendung, zu der der ASO-Cube gehört.
- `OPTIMIZATION_TYPE` ist ein Optimierungsvorgang für einen Cube. Folgende Werte werden akzeptiert:
 - `clearAggregations` entfernt Standard- und abfragebasierte Ansichten.
 - `createAggregations` erstellt standardmäßige Essbase-Aggregatsichten. Mit dieser Option können Sie die Standardaggregation anstelle der abfragebasierten Aggregation durchführen.
 - `startQueryTracking` startet das Abfrage-Tracking.
 - `stopQueryTracking` stoppt das Abfrage-Tracking. Mit dieser Option können Sie das Erfassen von Optimierungsinformationen in Essbase stoppen. Essbase setzt das Erfassen von Optimierungsinformationen fort, bis Sie das Abfrage-Tracking oder Essbase stoppen. Essbase kann Ansichten basierend auf erfassten Daten aggregieren, bis das Abfrage-Tracking gestoppt wird.
 - `createQBOAggregations` erstellt Essbase-Aggregatsichten basierend auf den optimierten Abfragen, die Sie nach dem Aktivieren des Abfrage-Tracking ausführen.

Beispiele

- Standardmäßige und abfragebasierte Aggregationssichten verwerfen:
`epmautomate optimizeASOCube BksML12 clearAggregations`
- Abfrage-Tracking starten:
`epmautomate optimizeASOCube BksML12 startQueryTracking`
- Essbase-Aggregatsichten basierend auf den optimierten Abfragen erstellen, die Sie nach dem Starten des Abfrage-Tracking ausführen:
`epmautomate optimizeASOCube BksML12 createQBOAggregations`

programDocumentationReport

Erstellt den Programmdokumentationsbericht mit der Profitability and Cost Management-Anwendungslogik.

Sie können den Bericht mit dem Befehl [downloadFile](#) auf einen lokalen Computer herunterladen.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Verwendung

```
epmautomate programDocumentationReport APPLICATION_NAME POV_NAME
[fileName=FILE_NAME] [fileType=PDF|WORD|EXCEL|HTML] [useAlias=true|false]
stringDelimiter="DELIMITER", wobei Folgendes gilt:
```

- *APPLICATION_NAME* ist der Name der Profitability and Cost Management-Anwendung, für die der Programmdokumentationsbericht erstellt werden soll.
- *POV_NAME* ist der Name des Modell-POVs in der Anwendung, für die der Bericht erstellt werden soll.
- *fileName* (optional) ist ein eindeutiger Name (einschließlich Erweiterung) für die Berichtsdatei. Der Standardname der Berichtsdatei lautet `HPCMLProgramDocumentationReport_APPLICATION_NAME_POV_NAME.pdf`.
- *fileType* (optional) ist das Format der Ausgabedatei. Das Standardformat ist `PDF`.
- *useAlias* (optional) gibt an, ob Aliasnamen anstelle von Elementnamen gedruckt werden sollen. Der Standardwert ist `false`.
- *stringDelimiter* ist das in POV-Werten verwendete Trennzeichen. Trennzeichen müssen in doppelte Anführungszeichen gesetzt werden.

Beispiel

```
epmautomate programDocumentationReport BksML30 2019_Feb_Actual
fileName=Feb-Actual.xls fileType=Excel useAlias=true stringDelimiter="_"
```

provisionReport

Generiert einen Rollenzuweisungsbericht (CSV) und speichert ihn am Standardspeicherort für Downloads.

Der Bericht enthält die vordefinierten Rollen (z.B. Poweruser für Servicename) und die Anwendungsrollen (z.B. die Planning-Anwendungsrolle "Massenzuweisung"), die Benutzern zugewiesen sind. Verwenden Sie den Befehl [downloadFile](#), um den Bericht herunterzuladen.

Es können zwei Versionen des Berichts generiert werden: vereinfacht oder klassisch. Der vereinfachte Bericht ist mit dem Rollenzuweisungsbericht identisch, der über das Fenster Zugriffskontrolle verfügbar ist. Der vereinfachte Bericht enthält nicht die Anwendungsrollen, die in vordefinierten Rollen zusammengefasst sind, oder die Komponentenrollen der Anwendungsrollen, die dem Benutzer zugewiesen sind. Die klassische Version des Berichts enthält die Komponentenrollen, die in den vordefinierten Rollen zusammengefasst sind, denen Benutzer zugewiesen sind. Der Bericht enthält außerdem die Anwendungsrollen, die dem Benutzer (direkt oder über Gruppen) zugewiesen sind.

Durch das Erstellen dieses Berichts werden alle Benutzer- und Rolleninformationen aktualisiert, die in der Zugriffskontrolle verfügbar sind.

Nur bei OCI (2. Generation): Oracle Enterprise Performance Management Cloud behandelt deaktivierte Benutzer als Benutzer, die keinen vordefinierten Rollen zugewiesen sind, auch wenn diese Benutzer vor der Deaktivierung vordefinierte Rollen hatten. Informationen zu deaktivierten Benutzern werden nicht in diesem Bericht aufgenommen.



Hinweis:

Dieser Befehl ist in künftigen Releases nicht mehr verfügbar. Verwenden Sie anstelle dieses Befehls den Befehl [roleAssignmentReport](#), mit dem ein ähnlicher Bericht erstellt wird.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate provisionReport REPORT_NAME [format=classic|simplified]
[userType=serviceUsers|IDAdmins] wobei:
```

- `REPORT_NAME` ein Name für den Bericht ist.
- `format` optional die Formatierung des Berichts bestimmt. Zulässige Werte:
 - `simplified`, die Standardoption, erstellt einen Bericht, der mit dem im Fenster Zugriffskontrolle generierten Rollenzuweisungsbericht identisch ist.
 - `classic` erstellt einen Bericht mit den Komponentenrollen, die in den vordefinierten Rollen zusammengefasst sind, denen Benutzer zugewiesen sind. Der Bericht enthält außerdem die Anwendungsrollen, die dem Benutzer (direkt oder über Gruppen) zugewiesen sind.
- `userType` optional die Benutzer angibt, die im Bericht enthalten sein sollen. Wenn Sie für diesen Parameter keinen Wert angeben, wird der Standardwert `serviceUsers` verwendet. Zulässige Werte:
 - `serviceUsers` erstellt einen Bericht mit Informationen zu allen Funktionsbenutzern (Identitätsdomainadministratoren sind nicht enthalten, wenn sie keiner vordefinierten Rolle zugewiesen sind, die Zugriff auf die Anwendung gewährt).
 - `IDAdmins` erstellt einen Bericht, der nur die Benutzer enthält, die der Rolle Identitätsdomainadministrator zugewiesen sind. Der Bericht ist im klassischen und im vereinfachten Format identisch.

Beispiele

- Einen klassischen Bericht erstellen: `epmautomate provisionReport myProvReport.CSV format=classic`

- Einen vereinfachten Bericht erstellen:
 - `epmautomate provisionReport myProvReport.CSV format=simplified`
 - `epmautomate provisionReport myProvReport.CSV userType=serviceUsers`
- Einen Bericht erstellen, der nur Identitätsdomainadministratoren enthält:
 - `epmautomate provisionReport myProvReport.CSV userType=IDAdmins`
 - `epmautomate provisionReport myProvReport.CSV userType=IDAdmins format=classic`

purgeArchivedTmTransactions

Löscht archivierte abgeglichene Transaktionen aus der Anwendung Account Reconciliation.

Mit dem Befehl [archiveTmTransactions](#) können Sie alte abgeglichene Transaktionen regelmäßig archivieren und dann diesen Befehl ausführen, um sie aus Account Reconciliation zu löschen, um eine optimale Anwendungsgröße sicherzustellen.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

`epmautomate purgeArchivedTMTransactions JobID=JOB_ID`, wobei `JobID` die ID des Jobs zum Archivieren von TM-Transaktionen ist, der zur Archivierung abgeglichener Transaktionen ausgeführt wurde. Diese Job-ID wird in der EPM Automate-Konsole angezeigt, wenn Sie den Befehl [archiveTmTransactions](#) ausführen. Sie finden sie auch in der Jobkonsole.

Beispiel

```
epmautomate purgeArchivedTMTransactions JobID=100000002655003
```

purgeTmTransactions

Entfernt abgeglichene Transaktionen aus Account Reconciliation.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

`epmautomate purgeTmTransactions matchType age [filterOperator=VALUE] [filterValue=VALUE] [logFilename=FILE_NAME]`, wobei Folgendes gilt:

- `matchType` ist die ID (TextID) des Abgleichstyps, aus dem die abgeglichenen Transaktionen gelöscht werden müssen.
- `age` gibt die Anzahl Tage seit dem letzten Abgleich der Transaktion an. Abgeglichene Transaktionen, die diesem Wert entsprechen oder älter sind, werden gelöscht.
- `filterOperator` (optional) stellt eine der folgenden Filterbedingungen zum Identifizieren der Konten dar, die abgeglichene Transaktionen für den Löschvorgang enthalten. Dieser Wert wird mit `filterValue` kombiniert, um die Konten zu ermitteln, aus denen abgeglichene Transaktionen entfernt werden müssen:
 - `equals`
 - `not_equals`
 - `starts_with`
 - `ends_with`
 - `contains`
 - `not_contains`
- `filterValue` (optional) ist ein Filterwert zum Ermitteln der Transaktionen, die entfernt werden sollen. Wenn `filterOperator` den Wert `equals` oder `not_equals` aufweist, können Sie eine durch Leerzeichen getrennte Liste verwenden, um mehrere Werte anzugeben. Beispiel: `filterValue=101-120 filterValue=102-202`. Wenn mehrere Werte angegeben werden, werden Transaktionen aus Konten zum Entfernen ausgewählt, die zu einer beliebigen Kombination aus Filteroperator und Filterwert passen.
- `logFilename` (optional) ist der Name einer Logdatei zum Aufzeichnen von Informationen über die Befehlsaktivität. Wenn kein Dateiname angegeben wird, wird automatisch eine Logdatei mit der Bezeichnung `PurgeTransactions_JOB_ID` generiert.

Note:

Wenn `filterOperator` und `filterValue` nicht angegeben sind, werden alle abgeglichenen Transaktionen, die dem Wert `age` entsprechen oder älter sind, aus allen Konten für den angegebenen `matchType` entfernt.

Beispiele

- Abgeglichene Transaktionen, die 180 Tage oder älter sind, für den Abgleichstyp `cashrecon` entfernen:
`epmautomate purgeTMTransactions cashrecon 180 logFile=tmlogs.log`
- Abgeglichene Transaktionen, die 180 Tage oder älter sind, für den Abgleichstyp `cashrecon` für das Konto 101-120 oder 102-202 entfernen:
`epmautomate purgeTMTransactions cashrecon 180 filterOperator>equals filterValue=101-120 FilterValue=102-202`

- Abgegliche Transaktionen, die 180 Tage oder älter sind, für den Abgleichstyp `cashrecon` für ein beliebiges Konto mit der Zeichenfolge `11` entfernen:
`epmautomate purgeTMTransactions cashrecon 180 filterOperator=contains
filterValue=11`

recomputeOwnershipData

Berechnet Anteilsdaten neu.

Eine Neuberechnung von Anteilsdaten in Financial Consolidation and Close ist in den folgenden Situationen erforderlich:

- Nach dem Hinzufügen oder Löschen von Überschreibungsregeln für Anteilsmanagementkonten
- Nach dem Ändern von Bereichseinstellungen für Konsolidierungsmethoden
- Nach einer Datenbankaktualisierung, und zwar unabhängig davon, ob die Entitystruktur geändert wurde

Eine Neuberechnung der Anteilsdaten in Tax Reporting ist nach jeder Datenbankaktualisierung erforderlich, auch wenn die Entitystruktur nicht geändert wurde.

Gilt für

Financial Consolidation and Close und Tax Reporting.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer

Verwendung

`epmautomate recomputeOwnershipData Scenario Year Period` wobei:

- `Scenario` der Name des Szenarios ist, das neu berechnet werden soll.
- `Year` das neu zu berechnende Jahr ist.
- `Period` die erste Periode des neu zu berechnenden Jahres ist.
Die ausgewählte Periode und alle nachfolgenden Perioden werden neu berechnet.

Hinweis:

Ein POV, der eine Neuberechnung erfordert, kann erst konsolidiert werden, nachdem die Anteilsdaten neu berechnet wurden.

Beispiel

`epmautomate recomputeOwnershipData FCCS_total_Actual FY19 Jan`

recreate

Setzt eine Umgebung auf einen bereinigten Status zurück, indem das Deployment neu erstellt wird.

Sie erstellen das Deployment neu, um die folgenden Aufgaben abzuschließen:

- Eine Umgebung bereinigen, bevor ein vollständiger Snapshot importiert wird
- Den Geschäftsprozess ändern, der in einer Umgebung bereitgestellt werden kann
- Die Oracle Essbase -Version ändern, die in anderen Oracle Enterprise Performance Management Cloud-Umgebungen als Narrative Reporting-, Oracle Enterprise Data Management Cloud- und Account Reconciliation-Umgebungen verwendet wird, in denen Essbase nicht verwendet wird
Standardmäßig werden EPM Standard Cloud Service- und EPM Enterprise Cloud Service-Umgebungen mit hybridfähigen Essbase-Anwendungen bereitgestellt, während Legacy-Umgebungen mit nicht hybridfähigen Essbase-Anwendungen bereitgestellt werden.

Ein Upgrade des Deployments von nicht hybridfähigen Essbase-Anwendungen in Legacy-Umgebungen ist erforderlich, um:

- Die erweiterte Dimensionalität in vorhandenen Financial Consolidation and Close-Legacy-Umgebungen zu unterstützen
- Hybrid Block Storage-(BSO-)Anwendungen in Legacy-Umgebungen von Enterprise Planning und Planning Modules zu aktivieren

Ein Downgrade des Deployments von Hybrid-fähigem Essbase in EPM Enterprise Cloud Service-Umgebungen ist erforderlich, wenn Sie einen Snapshot aus einer Umgebung importieren, die eine nicht Hybrid-fähige Version von Essbase aufweist.

Ausführliche Informationen zu Hybrid Essbase und zu den Überlegungen für ein Upgrade auf Hybrid Essbase finden Sie unter Informationen zu Essbase in EPM Cloud in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*.

▲ Achtung:

- Mit diesem Befehl werden die vorhandene Anwendung und optional alle benutzerdefinierten Artefakte aus der Umgebung gelöscht. Außerdem wird die Datenbank neu erstellt, und alle vorhandenen Daten werden entfernt. Nachdem der Service neu erstellt wurde, können Sie einen neuen Geschäftsprozess erstellen oder einen Geschäftsprozess mit Migration oder EPM Automate importieren.
- Dieser Befehl löscht die Migrationshistorie. Infolgedessen enthält der in Migration verfügbare Migrationsstatusbericht keine historischen Informationen.
- Bevor Sie diesen Befehl verwenden, führen Sie ein vollständiges Backup der Umgebung durch. Sie können einen Backup-Snapshot erstellen, indem Sie den Befehl [runDailyMaintenance](#) ausführen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate recreate [-f] [removeAll=true|false] [EssbaseChange=upgrade|downgrade] [TempServiceType=Service_type]`. Dabei gilt:

- `-f` erzwingt den Start des Neuerstellungsprozesses ohne Benutzerbestätigung. Wenn Sie die Option `-f` nicht verwenden, werden Sie von EPM Automate zur Bestätigung der Aktion aufgefordert.
- `removeAll` entfernt optional alle Snapshots sowie den Inhalt der Inbox (hochgeladene Dateien) und der Outbox (aus der Umgebung exportierte Dateien). Der Standardwert ist `false`. Damit werden die Snapshots sowie der Inhalt der Inbox und Outbox beibehalten.
- `EssbaseChange` (optional) führt ein Upgrade oder Downgrade der aktuellen Essbase-Version durch. EPM Automate behält die aktuelle Essbase-Version bei, wenn Sie diese Option nicht angeben. Zulässige Werte sind:
 - `upgrade`, um von einer nicht hybridfähigen Essbase-Anwendung zu einer hybridfähigen Essbase-Anwendung zu wechseln.
 - `downgrade`, um von einer hybridfähigen Essbase-Anwendung zu einer nicht hybridfähigen Essbase-Anwendung zu wechseln.

Achtung:

Lesen Sie vor dem Verwenden dieser Option die in Informationen zu Essbase in EPM Cloud in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators* verfügbaren Optionen.

- `TempServiceType` (optional) konvertiert eine Umgebung in eine andere Serviceumgebung. Die Geschäftsprozesse, die Sie in einer Umgebung implementieren können, werden durch Ihren Abonnementtyp bestimmt. Beispiel: Wenn Sie über ein EPM Standard Cloud Service-Abonnement verfügen, können Sie keine FreeForm-Anwendung erstellen, nachdem Sie die Umgebung von Account Reconciliation in Planning konvertiert haben. Wenn Sie über ein EPM Enterprise Cloud Service-Abonnement verfügen, können Sie einen beliebigen Geschäftsprozess in Ihrer Umgebung erstellen, nachdem Sie den Servicetyp entsprechend geändert haben. Informationen hierzu finden Sie unter Info zu den neuen EPM Cloud Services in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*. Das Verhalten dieses Parameters ist von Ihrem Abonnement abhängig.
 - **Andere Abonnements als EPM Standard Cloud Service und EPM Enterprise Cloud Service:** Sie können die Option `TempServiceType` verwenden, um eine Planning-, Enterprise Planning-, Tax Reporting- oder Financial Consolidation and Close-Umgebung vorübergehend in eine Account Reconciliation-, Oracle Enterprise

Data Management Cloud- oder Profitability and Cost Management-Umgebung zu konvertieren. Beispiel: Wenn Sie eine Planning-Umgebung gekauft haben, können Sie sie in eine Account Reconciliation-Umgebung konvertieren, indem Sie folgenden Befehl ausführen:

```
epmautomate recreate -f removeAll=true TempServiceType=ARCS
```

Nachdem Sie die Umgebung in Account Reconciliation konvertiert haben, können Sie sie mit dem entsprechenden Wert für `TempServiceType` in eine Oracle Enterprise Data Management Cloud- oder Profitability and Cost Management-Umgebung konvertieren. Zum Konvertieren in eine Profitability and Cost Management-Umgebung, müssen Sie beispielsweise den folgenden Befehl ausführen:

```
epmautomate recreate -f removeAll=true TempServiceType=PCMCS
```

Führen Sie den folgenden Befehl aus, um die Umgebung wieder in den ursprünglichen Servicetyp zu konvertieren:

```
epmautomate recreate -f
```

Profitability and Cost Management: Sie können Ihre Profitability and Cost Management-Umgebung mit dem folgenden Befehl in eine Planning-, Enterprise Planning- oder Enterprise Profitability and Cost Management-Umgebung konvertieren:

```
epmautomate recreate -f removeAll=true TempServiceType=PBCS
```

Führen Sie den folgenden Befehl aus, um die Umgebung wieder in die ursprüngliche Profitability and Cost Management-Umgebung zu konvertieren:

```
epmautomate recreate -f TempServiceType=PCMCS
```

 **Hinweis:**

Profitability and Cost Management-Umgebungen können nicht in Account Reconciliation-, Oracle Enterprise Data Management Cloud- oder Narrative Reporting-Umgebungen konvertiert werden.

– **EPM Standard Cloud Service- und EPM Enterprise Cloud Service-Abonnements:**

Sie können die Option `TempServiceType` verwenden, um eine EPM Cloud-Umgebung in eine beliebige andere unterstützte Umgebung zu konvertieren.

Für EPM Enterprise Cloud Service-Abonnements wird eine gemeinsame EPM Cloud-Plattform verwendet. Zunächst können Sie jeden unterstützten EPM Cloud-Geschäftsprozess bereitstellen.

Um von einem bereitgestellten Geschäftsprozess zu einem anderen zu wechseln, müssen Sie die Umgebung neu erstellen, indem Sie den neuen Servicetyp für die Umgebung angeben. Wenn Sie beispielsweise einen Account Reconciliation-Geschäftsprozess erstellt haben, nun aber eine Oracle Enterprise Data Management

Cloud-Umgebung erstellen möchten, müssen Sie den Neuerstellungsbefehl wie folgt ausführen.

```
epmautomate recreate -f removeAll=true TempServiceType=EDMCS
```

Wenn Sie einen Geschäftsprozess (zum Beispiel Account Reconciliation) in Planning, Tax Reporting oder Financial Consolidation and Close konvertieren, müssen Sie keinen Wert für `TempServiceType` angeben. Wenn Sie beispielsweise einen Account Reconciliation-Geschäftsprozess erstellt haben, aber nun eine Planning-Modulumgebung erstellen möchten, müssen Sie den Neuerstellungsbefehl wie folgt ausführen.

```
epmautomate recreate -f removeAll=true
```

Zulässige Werte für `TempServiceType`:

- ARCS zum Konvertieren einer Umgebung in eine Account Reconciliation-Umgebung
- EDMCS zum Konvertieren einer Umgebung in eine Oracle Enterprise Data Management Cloud-Umgebung
- EPRCS zum Konvertieren einer Umgebung in eine Narrative Reporting-Umgebung
- PCMCS zum Konvertieren einer Umgebung in eine Profitability and Cost Management-Umgebung

Beispiele

- So erstellen Sie die aktuelle Umgebung neu, stellen sie mit dem ursprünglichen Servicetyp wieder her (wenn zuvor eine Neuerstellung mit dem Parameter `TempServiceType` durchgeführt wurde) und führen ein Upgrade auf eine hybridfähige Essbase-Anwendung durch, ohne von Benutzern erstellte Snapshots und Inhalte der Inbox und Outbox zu entfernen:

```
epmautomate recreate -f EssbaseChange=upgrade
```

- So erstellen Sie die aktuelle Umgebung neu und stellen sie mit dem ursprünglichen Servicetyp wieder her (wenn zuvor eine Neuerstellung mit dem Parameter `TempServiceType` durchgeführt wurde) und entfernen Snapshots und die Inhalte der Inbox und Outbox:

```
epmautomate recreate -f removeAll=true
```

- So erstellen Sie die aktuelle Umgebung als Oracle Enterprise Data Management Cloud-Umgebung neu und entfernen den Inhalt der Inbox und Outbox sowie vorhandene Snapshots:

```
epmautomate recreate -f removeAll=true TempServiceType=EDMCS
```

- So erstellen Sie die aktuelle Account Reconciliation-Umgebung von EPM Enterprise Cloud Service als Financial Consolidation and Close-Umgebung neu und entfernen den Inhalt der Inbox und Outbox sowie die vorhandenen Snapshots:

```
epmautomate recreate -f removeAll=true
```

refreshCube

Aktualisiert den Anwendungs-Cube. Üblicherweise aktualisieren Sie den Cube nach dem Import der Metadaten in die Anwendung.

Die Zeit, die für die Ausführung eines Cube-Aktualisierungsvorgangs benötigt wird, hängt von den Änderungen, die an der Anwendungsstruktur vorgenommen werden,

und der Auswirkung auf den Cube ab. Beispiel: Eine Aktualisierung nach dem Aktualisieren eines dünn besetzten Block Storage Cube-Elements nimmt möglicherweise nur wenig Zeit in Anspruch, während eine Cube-Aktualisierung nach dem Aktualisieren eines dicht besetzten Block Storage Cube-Elements oder eines Aggregate Storage Cube-Elements eine beträchtliche Zeit dauern kann. Sie müssen sicherstellen, dass der Cube-Aktualisierungsvorgang vollständig abgeschlossen ist, bevor die Anwendung während des nächsten Wartungsfensters gesichert wird.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate refreshCube [JOB_NAME]`, wobei `JOB_NAME` optional der Name eines in der Anwendung definierten Datenbankaktualisierungsjobs ist.

Der Status des Vorgangs wird in der Konsole zurückgegeben, über die der Befehl ausgeführt wird. Sie können den Status auch über die Seite **Kürzliche Aktivität** im Fenster **Jobs** in der Anwendung anzeigen.

Beispiel

```
epmautomate refreshCube DaliyCubeRefresh
```

removeUserFromGroups

Entfernt die Mitgliedschaft eines Benutzers aus den Zugriffskontrollgruppen, die in einer ANSI- oder UTF-8-codierten CSV-Datei angegeben sind.

Das Dateiformat lautet wie folgt:

```
Group Name  
Group1  
Group2
```

Hinweis:

Diese Gruppen müssen unter Zugriffskontrolle vorhanden sein. Bei Gruppennamenswerten wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Verwenden Sie den Befehl [uploadFile](#), um die Datei in eine Umgebung hochzuladen.

Wenn die Befehlsausführung abgeschlossen ist, gibt EPM Automate Informationen zu jedem nicht erfolgreichen Eintrag an die Konsole aus. Lesen Sie diese Informationen, um zu verstehen, warum die Befehlsausführung für einige Einträge in der CSV-Datei nicht erfolgreich war.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator oder Zugriffskontrollemanager

Verwendung

`epmautomate removeUserFromGroups FILE_NAME User_Login` wobei Folgendes gilt:

- `FILE_NAME` ist der Name einer CSV-Datei, die die Namen der Zugriffskontrollgruppen enthält, aus denen die Mitgliedschaft des Benutzers entfernt werden soll.
- `User_Login` ist die Anmelde-ID eines Oracle Enterprise Performance Management Cloud-Benutzers, dessen Mitgliedschaft aus Gruppen in der Zugriffskontrolle entfernt werden soll. Diese Benutzeranmelde-ID muss in der Identitätsdomain für die Umgebung vorhanden sein und muss einer vordefinierten Rolle zugewiesen werden. Bei diesem Wert wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Beispiel

```
epmautomate removeUserFromGroups groups.csv jdoe@example.com
```

removeUsers

Löscht die Accounts der Identitätsdomain, die in einer ANSI- oder UTF-8-kodierten CSV-Datei identifiziert wurden, die in die Umgebung hochgeladen wurde.

Das Dateiformat lautet wie folgt:

```
User Login  
jane.doe@example.com  
jdoe@example.com
```

Verwenden Sie den Befehl [uploadFile](#), um Dateien in die Umgebung hochzuladen. Bei Benutzeranmeldewerten wird nicht zwischen Groß- und Kleinschreibung unterschieden. Beispiel: `jane.doe@example.com` wird genauso wie `Jane.Doe@Example.com` oder jede andere Variation der Groß-/Kleinschreibung behandelt.

 **Hinweis:**

- Die CSV-Datei darf nicht den Account des Benutzers enthalten, der diesen Befehl ausführt.
- Da Benutzeraccounts für alle Serviceumgebungen gleich sind, die ein Identitätsdomainadministrator unterstützt, wird beim Löschen eines Accounts für eine Umgebung dieser für alle Umgebungen gelöscht, die die Rolle Identitätsdomainadministrator gemeinsam verwenden.

Wenn die Befehlsausführung abgeschlossen ist, gibt EPM Automate Informationen zu jedem nicht erfolgreichen Eintrag an die Konsole aus. Lesen Sie diese Informationen, um zu verstehen, warum die Befehlsausführung für einige Einträge in der CSV-Datei nicht erfolgreich war.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Identitätsdomainadministrator und alle vordefinierten Rollen (Serviceadministrator, Poweruser, Benutzer oder Leseberechtigter)

Verwendung

`epmautomate removeUsers FILE_NAME` wobei `FILE_NAME` der Name einer CSV-Datei mit den Anmelde-IDs der Benutzer ist, die aus der Identitätsdomain entfernt werden sollen.

Beispiel

```
epmautomate removeUsers remove_users.CSV
```

removeUsersFromGroup

Entfernt Benutzer, die in einer ANSI- oder UTF-8-codierten CSV-Datei enthalten sind, aus einer in der Zugriffskontrolle verwalteten Gruppe.

Das Dateiformat lautet wie folgt:

```
User Login  
jdoe  
john.doe@example.com
```

Bei Benutzeranmeldewerten wird nicht zwischen Groß- und Kleinschreibung unterschieden. Beispiel: `jane.doe@example.com` wird genauso wie `Jane.Doe@Example.com` oder jede andere Variation der Groß-/Kleinschreibung behandelt. Verwenden Sie den Befehl [uploadFile](#), um die Datei mit den Benutzer-Logins in die Umgebung hochzuladen.

Wenn die Befehlsausführung abgeschlossen ist, gibt EPM Automate Informationen zu jedem nicht erfolgreichen Eintrag an die Konsole aus. Lesen Sie diese Informationen, um zu verstehen, warum die Befehlsausführung für einige Einträge in der CSV-Datei nicht erfolgreich war.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator oder Zugriffskontrollemanager

Verwendung

`epmautomate removeUsersFromGroup FILE_NAME GROUP_NAME` wobei:

- `FILE_NAME` der Name einer CSV-Datei mit den Anmeldenamen der Benutzer ist, die Sie aus einer in der Zugriffskontrolle verwalteten Gruppe entfernen möchten.
- `GROUP_NAME` ist der Name der Gruppe in der Zugriffskontrolle, aus der Sie Benutzer entfernen möchten. Bei diesem Wert wird nicht zwischen Groß- und Kleinschreibung unterschieden.



Hinweis:

Ein Benutzer wird aus einer Gruppe nur entfernt, wenn die beiden folgenden Bedingungen erfüllt sind:

- Die in der Datei enthaltenen Benutzer-Logins sind in der Identitätsdomain für die Umgebung vorhanden.
- Der Benutzer ist einer vordefinierten Rolle in der Identitätsdomain zugewiesen.

Beispiel

```
epmautomate removeUsersFromGroup user_file.CSV example_group
```

removeUsersFromTeam

Entfernt Benutzer von Oracle Enterprise Performance Management Cloud, die in einer CSV-Datei aufgelistet sind, aus einem Team.

Wenn ein Benutzer in der CSV-Datei kein Mitglied des Teams ist, ignoriert dieser Befehl den entsprechenden Benutzer. Bei den Werten in dieser Datei wird nicht zwischen Groß- und Kleinschreibung unterschieden. Die CSV-Datei weist das folgende Format auf:

```
User Login  
jdoe  
jane.doe@example.com
```

Verwenden Sie den Befehl [uploadFile](#), um die CSV-Datei in die Umgebung hochzuladen.

Gilt für

Financial Consolidation and Close, Tax Reporting und Account Reconciliation.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

`epmautomate removeUsersFromTeam FILE.CSV TEAM_NAME` wobei Folgendes gilt:

- `FILE` gibt eine CSV-Datei im UTF8-Format mit den Anmelde-IDs der Benutzer an, die aus dem Team entfernt werden sollen.
- `TEAM_NAME` gibt einen in der Zugriffskontrolle definierten Teamnamen an. Bei diesem Wert wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Beispiel

```
epmautomate removeUsersFromTeam example_users.csv example_team
```

renameSnapshot

Benennt einen Snapshot um, den Sie hochgeladen oder in einer Umgebung erstellt haben.

Wenn dieser Befehl zum Umbenennen eines Snapshots ausgeführt wird, der gerade generiert oder archiviert wird, tritt einer der folgenden Fehler auf:

- Datei nicht gefunden, wenn der Snapshot gerade generiert wird
- Archivprozess wird verarbeitet. Element kann nicht umbenannt oder gelöscht werden, wenn der Snapshot gerade archiviert wird

Benennen Sie den Wartungs-Snapshot in einer Umgebung nicht um. Um immer ein Backup des Wartungs-Snapshots zu haben, müssen Sie `Artifact Snapshot` aus der Umgebung auf einen lokalen Computer herunterladen und dann nach Bedarf umbenennen. Informationen hierzu finden Sie unter Überblick über den Wartungs-Snapshot in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, die der Anwendungsrolle Migrationsadministrator zugewiesen sind

Verwendung

`epmautomate renameSnapshot SNAPSHOT_NAME NEW_SNAPSHOT_NAME`, wobei Folgendes gilt:

- `SNAPSHOT_NAME` ist der Name eines vorhandenen Snapshots. Dieser Wert darf keine Sonderzeichen wie Leerzeichen, \ (umgekehrter Schrägstrich), / (Schrägstrich), * (Sternchen), ? (Fragezeichen), " (Anführungszeichen), < (Kleiner-als-Zeichen) und > (Größer-als-Zeichen) enthalten.
- `NEW_SNAPSHOT_NAME` ist der eindeutige Name, den Sie dem Snapshot zuweisen möchten.

Beispiel

```
epmautomate renameSnapshot "Example Snapshot" Example_Snapshot_18_09_25
```

replay

Gibt die Auslastung von Oracle Smart View for Office, der REST-API oder von EPM Automate in einer Umgebung wieder, um Performancetests unter starker Last zu ermöglichen. Dadurch wird geprüft, ob die Benutzererfahrung akzeptabel ist, wenn sich der Service unter einer bestimmten Last befindet.

Sie müssen eine Wiedergabedatei erstellen, die die Aktivitäten identifiziert, die für den Service ausgeführt werden müssen. Ausführliche Informationen zum Erstellen der Wiedergabedatei finden Sie unter [Ausführen des Wiedergabebefehls vorbereiten](#).

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Verwendung

```
epmautomate replay REPLAY_FILE_NAME.csv [duration=N] [trace=true]
[lagTime=t] [encrypt=true|false]. Dabei gilt:
```

- `REPLAY_FILE_NAME` ist eine CSV-Datei, in der die in der Umgebung auszuführenden Aktivitäten gespeichert werden.
- `Dauer` gibt optional die Anzahl der Minuten an, für die Aktivitäten in der Umgebung ausgeführt werden. Aktivitäten in der HAR-Datei werden einmalig ausgeführt, wenn der Wert nicht festgelegt ist. Wenn die Aktivitäten in der HAR-Datei in der durch diesen Parameter angegebenen Zeit abgeschlossen werden, führt EPM Automate die HAR-Datei erneut aus, bis die Aktivitäten abgeschlossen sind. Beispiel: Angenommen, Sie haben `duration=10` festgelegt, um eine HAR-Datei wiederzugeben, die drei Minuten für die Ausführung benötigt. In diesem Szenario führt der Befehl die HAR-Dateiaktivitäten viermal aus (mit einer Dauer von 12 Minuten), bis die vierte Iteration abgeschlossen ist.
- `trace=true` ist eine optionale Einstellung, mit der EPM Automate angewiesen wird, Tracedateien im XML-Format zu erstellen. Wenn diese optionale Einstellung angegeben ist, erstellt EPM Automate für jede HAR-Datei, die in der CSV-Wiedergabedatei enthalten ist, einen Ordner, in dem alle zugehörigen Tracedateien gespeichert werden. Für jede Aktivität in der HAR-

Datei generiert EPM Automate eine Tracedatei, die die Reaktion von Smart View enthält. Tracedateien weisen den Namen `trace-N.xml` auf. Beispiel: `trace-1.xml`, wobei N ein Zähler ist, der bei 1 beginnt.

Die Ordner, in denen die Tracedateien gespeichert werden, werden in dem Verzeichnis erstellt, aus dem EPM Automate ausgeführt wird. EPM Automate verwendet zur Benennung der Ordner eine Kombination aus der aktuellen Systemzeit der Umgebung und dem HAR-Dateinamen im Format `YYYY_MM_DD_HH_MM_SS_HAR_FILE_NAME`. Beispiel: Wenn der HAR-Dateiname `forecast1.har` lautet, lautet der Ordnername möglicherweise `2016_06_08_10_21_42_forecast1`.

- `[lagTime=t]` gibt optional die Anzahl der Sekunden an, die der Befehl zwischen der Ausführung jeder in der Wiedergabedatei enthaltenen HAR-Datei warten soll. Der Standardwert beträgt 5 Sekunden. Wenn Sie einen Wert unter 5 Sekunden angeben, wird ein Fehler angezeigt. Negative Zahlen (z.B. -1) und Brüche (z.B. 1/2) sind als Parameterwert nicht zulässig. Dezimalwerte werden unterstützt.

Nachdem die Ausführung der ersten HAR-Datei initiiert wurde, wartet der Befehl die von diesem Parameter angegebene Anzahl von Sekunden, um die Verarbeitung der nächsten HAR-Datei zu initiieren. Da Benutzeraktivitäten in der Regel nicht gleichzeitig initiiert werden, hilft die Festlegung dieses Parameters, eine realistischere Simulation der Last einer Umgebung zu erstellen.

Beispiel: Angenommen, Sie möchten die Last von 1000 Benutzern simulieren, die sich während der Spitzenzeit an einer Umgebung anmelden, um Aktivitäten auszuführen. Sie können HAR-Dateien erstellen, um diese Sessions zu simulieren, und dann diesen Befehl mit einer Verzögerung von 6 Sekunden ausführen, um die aufgewendete Last für die Umgebung zu replizieren.

- `encrypt=true|false` gibt optional an, ob alle in der Wiedergabedatei enthaltenen Kennwörter verschlüsselt werden sollen. Der Standardwert ist `true`. Zum Verschlüsseln des Kennworts wird ein zufällig generierter Verschlüsselungsschlüssel verwendet.

Detaillierte Schritte für die Ausführung dieses Befehls finden Sie unter [Wiedergabesession - Beispiel](#).

Beispiel

```
epmautomate replay forecast1.CSV duration=60 lagTime=5.6
```

resetService

Startet die Umgebung neu. Sie können optional die Umgebung automatisch optimieren, bevor Sie sie neu starten, um sicherzustellen, dass die Oracle Essbase-Indexcaches für Block Storage Option-(BSO-)Cubes für Ihre Anwendung optimiert sind.

Standardmäßig werden Ihre Umgebungen direkt neu gestartet, nachdem die tägliche Wartung abgeschlossen ist. Das automatische Optimieren Ihrer Umgebungen ist z.B. wichtig, nachdem ein Snapshot in eine Umgebung importiert wurde. Verwenden Sie diesen Befehl nur, wenn Sie eine schwerwiegende Verschlechterung der Performance feststellen oder wenn Sie Fehlermeldungen darüber erhalten, dass die Umgebung nicht verwendet werden kann. Das Neustarten einer Umgebung hat keine Auswirkungen auf Anwendungsanpassungen (z.B. lokale Änderung, Einstellungen im Hinblick auf Theme und Währung usw.). Der Neustart kann bis zu 15 Minuten dauern.

Bevor Sie diesen Befehl verwenden, stellen Sie sicher, dass keine Geschäftsregeln in der Umgebung ausgeführt werden.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate resetService "comment" [AutoTune=true|false] [-f]`, wobei:

- `comment` eine Beschreibung des Problems ist, weswegen Sie die Umgebung zurückgesetzt haben. Kommentare müssen in Anführungszeichen gesetzt werden.
- `AutoTune` (optional) angibt, ob die Umgebung automatisch abgestimmt werden soll, um BSO-Cubes der Essbase-Caches Ihrer Anwendung zu optimieren. Der Standardwert ist `false`.

Verwenden Sie diesen Parameter nur in Umgebungen, die Essbase-BSO-Cubes verwenden: Planning (einschließlich Planning Modules), Financial Consolidation and Close und Tax Reporting.

- `-f` (optional) angibt, dass Sie den Neustart der Umgebung ohne zusätzliche Benutzerinteraktion erzwingen möchten. Wenn Sie diese Option nicht verwenden, werden Sie von EPM Automate zur Bestätigung der Aktion aufgefordert. Diese Option ist nützlich, wenn Sie ein Skript planen, das diesen Befehl verwendet.

Beispiele

- `epmautomate resetService "Users experience slow connections; force restarting the environment" -f`
- `epmautomate resetservice "Users experience unacceptably slow connections"`
- `epmautomate resetService "Optimizing the Essbase cache" AutoTune=true`

restoreBackup

Kopiert einen verfügbaren Backup-Snapshot einer OCI (Gen 2)-Umgebung, um ihn für den Import in die Umgebung verfügbar zu machen. Dieser Befehl funktioniert nicht in Classic Oracle Enterprise Performance Management Cloud-Umgebungen.

Mit dem Befehl [listBackups](#) können Sie ermitteln, ob das Backup, das Sie wiederherstellen möchten, verfügbar ist. Durch die Selfservice-Wiederherstellung eines Snapshots in der Umgebung können Sie Verarbeitungszeit einsparen. Verwenden Sie nach dem Wiederherstellen des Snapshots den Befehl [importSnapshot](#), um den Snapshot in die Umgebung zu importieren.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise

Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, die der Anwendungsrolle Migrationsadministrator zugewiesen sind

Verwendung

`epmAutomate restoreBackup SNAPSHOT_NAME [targetName=TARGET_SNAPSHOT_NAME]`, wobei Folgendes gilt:

- `SNAPSHOT_NAME` ist der Name eines Backup-Snapshots, der laut Auflistung durch den Befehl `listBackups` in der Umgebung verfügbar ist.
- `targetName` (optional) ist der Name für den Backup-Snapshot ohne Erweiterung in der Zielumgebung. Wenn Sie diesen Wert nicht angeben, wird der Backup-Snapshot mit dem Wert von `SNAPSHOT_NAME`, aber mit einem `_` (Unterstrich) anstelle des `/` (Schrägstrich) in der Zielumgebung wiederhergestellt. Beispiel: Wenn `SNAPSHOT_NAME` `2022-05-14T00:08:56/Artifact_Snapshot.zip` lautet, ist der `targetName` `2022-05-14T00:08:56_Artifact_Snapshot.zip`.

Beispiele

- Für andere Services als Narrative Reporting:
`epmAutomate restoreBackup 2022-05-14T00:08:56/Artifact_Snapshot.zip`
`targetName=example_Artifact_Snapshot`
- Nur für Narrative Reporting:
`epmAutomate restoreBackup 2022-02-16T21:00:02/EPRCS_Backup`
`targetName=Example_EPRCS_Backup`

restructureCube

Führt eine vollständige Neustrukturierung eines Block Storage-Cubes durch, um die Fragmentierung zu entfernen oder zu reduzieren. Bei der Neustrukturierung werden auch leere Blöcke entfernt. Außerdem werden Änderungen aus der Anwendung nicht in den Cube gepusht.



Hinweis:

Sie müssen vor dem Ausführen dieses Befehls sicherstellen, dass niemand die Anwendung verwendet.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate restructureCube CUBE_NAME` wobei `CUBE_NAME` exakt dem Namen eines Cubes in einer Anwendung entspricht.

Beispiel

`epmautomate restructureCube Plan1`

roleAssignmentAuditReport

Erstellt in OCI-Umgebungen (2. Generation) einen Auditbericht, in dem die Änderungen aufgeführt werden, die an vordefinierten und Anwendungsrollenzuweisungen gemäß dem Aufbewahrungszeitraum für Auditdaten für Ihre Umgebung in einem bestimmten Zeitraum vorgenommen wurden. Der Standardaufbewahrungszeitraum beträgt 30 Tage. Sie können den Zeitraum auf maximal 90 Tage verlängern, indem Sie die Einstellung für den Aufbewahrungszeitraum in Tagen in der Oracle Cloud Identity Console entsprechend anpassen. Um die Auditdaten für einen Zeitraum von mehr als 90 Tagen aufzubewahren, können Sie diesen Bericht und den [Bericht zu ungültigen Anmeldungen](#) herunterladen und archivieren.

Im Auditbericht zur Rollenzuweisung wird der Benutzeranmeldename aufgeführt, für den eine Rollenänderung (in der Spalte "Aktion") vorgenommen wurde. Er enthält zudem die Rolle, die zugewiesen oder deren Zuweisung aufgehoben wurde, den Benutzer, der die Rollenänderung vorgenommen hat (Spalte "Administrator"), und den Zeitstempel (UTC) im 24-Stunden-Format mit der Angabe, wann die Aktion abgeschlossen wurde.

	A	B	C	D	E
1	User Name	Role	Action	Administrator	Date and Time
2	jane.doe@example.com	User	Assign	epm.admin@example.com	July 09, 2021 03:54:52 UTC
3	jane.doe@example.com	Run Integrations	Assign	epm.admin@example.com	July 09, 2021 03:54:52 UTC
4	john.doe@example.com	Service Administrator	Assign	epm.admin@example.com	July 09, 2021 03:51:28 UTC
5	john.smith@example.com	Power User	Unassign	epm.admin@example.com	July 09, 2021 03:53:04 UTC
6	john.smith@example.com	User	Assign	epm.admin@example.com	July 09, 2021 03:54:06 UTC

Informationen zu gelöschten Benutzern, die zuvor vordefinierten Rollen in der Umgebung zugewiesen wurden, sind mit dem Anzeigenamen (Vor- und Nachname) des Benutzers in der Spalte "Benutzername" aufgeführt. In solchen Fällen gibt die Spalte "Rolle" die vordefinierte Rolle an, die der Benutzer hatte, bevor das Benutzerkonto gelöscht wurde. Diese Änderung gilt nicht für Anwendungsrollen, die dem gelöschten Benutzer ggf. zugewiesen waren. Solche Zuweisungen werden mit dem Benutzeranmeldenenamen des jeweiligen Benutzers angezeigt. Ein Beispiel finden Sie in der folgenden Abbildung im roten Kasten.

	A	B	C	D	E
1	User Name	Role	Action	Administrator	Date and Time
2	Jane Doe	User	Assign	epm.admin@example.com	July 09, 2021 03:54:52 UTC
3	jane.doe@example.com	Run Integrations	Assign	epm.admin@example.com	July 09, 2021 03:54:52 UTC
4	john.doe@example.com	Service Administrator	Assign	epm.admin@example.com	July 09, 2021 03:51:28 UTC
5	john.smith@example.com	Power User	Unassign	epm.admin@example.com	July 09, 2021 03:53:04 UTC
6	john.smith@example.com	User	Assign	epm.admin@example.com	July 09, 2021 03:54:06 UTC

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator oder Identitätsdomainadministrator und alle vordefinierten Rollen (Serviceadministrator, Poweruser, Benutzer oder Leseberechtigter)

Verwendung

`epmAutomate roleAssignmentAuditReport FROM_DATE TO_DATE FILE_NAME.CSV`, wobei Folgendes gilt:

- *FROM_DATE* gibt das Startdatum der Periode (im Format *JJJJ-MM-TT*) an, für die der Bericht generiert werden soll. Dieses Datum muss innerhalb des Aufbewahrungszeitraums für Audits liegen, der in der Oracle Cloud Identity-Konsole festgelegt ist.
- *TO_DATE* gibt das Enddatum der Periode (im Format *JJJJ-MM-TT*) an, für die der Bericht generiert werden soll.
- *FILE_NAME* ist der Name einer CSV-Datei für den Bericht. Sie können den generierten Bericht mit dem Befehl [downloadFile](#) herunterladen.

Beispiel

`epmAutomate roleAssignmentAuditReport 2021-06-01 2021-07-30 RoleAuditReport.CSV`

roleAssignmentReport

Generiert einen Rollenzuweisungsbericht (CSV).

Dieser Bericht enthält standardmäßig die vordefinierten Rollen (z.B. Serviceadministrator) und die Anwendungsrollen (z.B. die Planning-Anwendungsrollen Zuweisender der Eigentümerrechte an Genehmigungen, Genehmigungssupervisor, Genehmigungsadministrator und Genehmigungsprozessdesigner), die Benutzern zugewiesen sind. Optional kann dieser Bericht auch generiert werden, um die Identitätsdomainadministratoren für Ihre Umgebung aufzulisten. Dieser Bericht entspricht der CSV-Version des Rollenzuweisungsberichts, der über die Zugriffskontrolle generiert wird.

	A	B	C	D	E	F
1	User Login	First Name	Last Name	Email	Role	Granted through Group
2	Jdoe	John	Doe	jdoe@example.com	Planner	
3	jdoe	John	Doe	jdoe@example.com	Power User	
4	Jdoe	John	Doe	jdoe@example.com	Service Administrator	
5	jdoe	John	Doe	jdoe@example.com	Viewer	
6	Jdoe	John	Doe	jdoe@example.com	Mass Allocation	example->Power User
7	jdoe	John	Doe	jdoe@example.com	Run Integration	
8	jane.doe@example.com	Jane	Doe	jane.doe@example.com	Planner	
9	jane.doe@example.com	Jane	Doe	jane.doe@example.com	Power User	
10	jane.doe@example.com	Jane	Doe	jane.doe@example.com	Viewer	
11	jane.doe@example.com	Jane	Doe	jane.doe@example.com	Mass Allocation	example

Durch das Erstellen dieses Berichts werden alle Benutzer- und Rolleninformationen aktualisiert, die in der Zugriffskontrolle verfügbar sind.

Nur bei OCI (2. Generation): Oracle Enterprise Performance Management Cloud behandelt deaktivierte Benutzer als Benutzer, die keinen vordefinierten Rollen zugewiesen sind, auch wenn diese Benutzer vor der Deaktivierung vordefinierte Rollen hatten. Informationen zu deaktivierten Benutzern werden nicht in diesem Bericht aufgenommen.



Hinweis:

Mit diesem Befehl wird ein Bericht generiert, der dem mit dem Befehl `provisionReport` erstellten Bericht entspricht.

Sie können den Bericht mit dem Befehl `downloadFile` herunterladen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator oder Zugriffskontrollmanager

Verwendung

```
epmautomate roleAssignmentReport REPORT_NAME.CSV [userType=IDAdmins | serviceUsers], wobei:
```

- `REPORT_NAME` ein Name für den Bericht ist.
- `userType` optional den Typ der Benutzer angibt, deren Informationen im Bericht enthalten sein sollen. Der Standardwert ist `serviceUsers`. Gültige Werte sind:
 - `serviceUsers` erstellt einen Bericht mit Informationen zu allen Funktionsbenutzern (Identitätsdomainadministratoren sind nicht enthalten, wenn sie keiner vordefinierten Rolle zugewiesen sind, die Zugriff auf die Anwendung gewährt).
 - `IDAdmins` erstellt einen Bericht, der nur die Benutzer enthält, die der Rolle Identitätsdomainadministrator zugewiesen sind.

Beispiele

- Generierung des Berichts, sodass nur funktionale Benutzer aufgelistet werden:
 - `epmautomate roleAssignmentReport myReport.CSV`
 - `epmautomate roleAssignmentReport myReport.CSV userType=serviceUsers`
- Generierung des Berichts, sodass nur Identitätsdomainadministratoren aufgelistet werden:
`epmautomate roleAssignmentReport myReport.CSV userType=IDAdmins`

runAutomatch

Führt den Prozess "Automatischer Abgleich" aus, um Transaktionen mit den Regeln abzugleichen, die von einem Serviceadministrator definiert werden.

Hinweis:

Führen Sie diesen Befehl aus, nachdem Sie Transaktionsdaten mit dem Befehl [importTmPremappedTransactions](#) oder dem Befehl [runDataRule](#) in das Modul "Transaktionsabgleich" importiert haben.

Sie können den Status des automatischen Abgleichsprozesses in der Registerkarte **Jobhistorie** in Account Reconciliation überwachen.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

`epmautomate runAutomatch RECONCILIATION_TYPE wobei RECONCILIATION_TYPE ein in Account Reconciliation definierter Abstimmungstyp ist.`

Beispiel

```
epmautomate runAutomatch INTERCOMPANY
```

runBatch

Führt einen Data Management-Batch aus.

Wenn der Batchausführungsmodus in Data Management auf "Seriell" festgelegt ist, wird die Steuerung zurückgegeben, wenn alle Jobs im Batch abgeschlossen sind. Wenn der Modus auf "Parallel" festgelegt ist, wird die Steuerung zurückgegeben, wenn alle Jobs im Batch zur Ausführung weitergeleitet wurden.

Hinweis:

Dieser Befehle kann nicht verwendet werden, um die direkte Dataload-Integration aus Datenquellen in Oracle Enterprise Performance Management Cloud auszuführen. Mit EPM Integration Agent können Sie direkte Dataloads integrieren. Ausführliche Informationen finden Sie unter [Direkten Dataload mit EPM Integration Agent durchführen](#) in der Dokumentation *Komponente "Datenintegration" für Oracle Enterprise Performance Management Cloud verwalten*.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate runBatch BATCH_NAME`, wobei *BATCH_NAME* der Name eines in Data Management definierten Batches ist.

Beispiele

```
epmautomate runBatch Accounting_batch
```

runBusinessRule

Eine Geschäftsregel wird gestartet.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser (wenn Regelausführungszugriff erteilt wurde)

Verwendung

`epmautomate runBusinessRule RULE_NAME [PARAMETER=VALUE]` wobei Folgendes gilt:

- *RULE_NAME* entspricht exakt dem Namen einer Geschäftsregel, so wie er in der Umgebung definiert wurde.
- *PARAMETER=VALUE* gibt die optionalen Laufzeitparameter und ihre Werte an, die für die Ausführung der Geschäftsregel erforderlich sind.

 **Hinweis:**

- Dieser Befehl kann nur eine einzige Geschäftsregel ausführen. Um ein Regelset auszuführen, müssen Sie den Befehl [runRuleSet](#) verwenden.
- Die Regel wird für den Plantyp ausgeführt, für den sie bereitgestellt wurde.
- Standardwerte werden verwendet, wenn Sie keine Werte für Laufzeitparameter angeben. Der Befehl ignoriert Runtime Prompts, die keine genauen Übereinstimmungen mit den in der Regel definierten Prompts darstellen.
- Mit *PARAMETER=VALUE*-Paaren können Sie so viele Runtime Prompts angeben wie für die Geschäftsregel erforderlich sind. Das folgende Beispiel verwendet zwei Runtime Prompts (*Period* und *Entity*) und deren Werte (*Q1* und *USA*).
Informationen zum Eingeben mehrerer Werte für einen Parameter finden Sie unter [Mehrere Werte für Parameter angeben](#).

Beispiel

```
epmautomate runBusinessRule RollupUSSales Period=Q1 Entity=USA
```

runCalc

Führt Berechnungen in einer Anwendung aus.

Wenn Sie diesen Befehl verwenden, können Sie Berechnungen mit Regeln in einem Modell-POV für Daten in einem anderen Daten-POV ausführen, ohne dass Regeln zwischen POVs kopiert werden müssen.

Gilt für

Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

```
epmautomate runCalc APPLICATION_NAME POV_NAME [DATA_POV_NAME] PARAMETER=VALUE  
[comment="comment"] stringDelimiter="DELIMITER", wobei Folgendes gilt:
```

- *APPLICATION_NAME* ist der Name der Profitability and Cost Management-Anwendung, die den zu berechnenden POV enthält.
- *POV_NAME* ist der Name des zu berechnenden Modell-POV.
- *DATA_POV_NAME* ist der optionale Name des Daten-POV, der mit den Regeln des Modell-POV berechnet werden soll.

Wenn *DATA_POV_NAME* nicht angegeben ist, wird standardmäßig *POV_NAME* verwendet.

Sie können nur `exeType=ALL_RULES` verwenden, wenn Sie *DATA_POV_NAME* angeben.

- `PARAMETER=VALUE` gibt Laufzeitparameter und deren Werte zum Ausführen der Berechnung an. Geben Sie so viele Parameter- und Wertpaarungen an, wie der Prozess erfordert. Gültige Parameter und deren Werte:
 - `exeType=ALL_RULES|RULESET_SUBSET|SINGLE_RULE` gibt den Regelausführungstyp an. Dies ist ein erforderlicher Parameter. Abhängig von dem für `exeType` festgelegten Wert, können die folgenden Parameter angegeben werden:
 - * Wenn Sie `exeType=ALL_RULES` angeben, schließen Sie keine Parameter ein, die zu einer Regelteilmenge oder einer einzelnen Regel gehören, wie z.B. `subsetStart`, `subsetEnd`, `ruleSetName` und `ruleName`. Sie müssen diesen Wert für `exeType` verwenden, wenn Sie den Parameter `DATA_POV_NAME` festlegen.
 - * Wenn Sie `exeType=SINGLE_RULE` angeben, geben Sie nur die Werte für `ruleSetName` und `ruleName` an.
 - * Wenn Sie `exeType=RULESET_SUBSET` angeben, geben Sie die Werte für `subsetStart` und `subsetEnd` an.
 - `subsetStart` gibt die Abfolgenummer der ersten Regel im Regelset an, das ausgeführt werden soll.
 - `subsetEnd` gibt die Abfolgenummer der letzten Regel im Regelset an, das ausgeführt werden soll.
 - `ruleSetName` gibt das Regelset an, das die Berechnungen enthält, die ausgeführt werden sollen.
 - `ruleName` entspricht dem Namen der Regel, die ausgeführt werden soll (auszuführen als eine einzelne Regel).
 - `isClearCalculated=true|false` gibt an, ob vorhandene Berechnungen gelöscht werden sollen.
 - `isExecuteCalculations=true|false` gibt an, ob Berechnungen ausgeführt werden sollen.
 - Setzen Sie für `isRunNow=true|false` diesen Wert auf `true`, um den Prozess jetzt auszuführen.
 - `optimizeReporting=true|false` setzt diesen optionalen Wert auf `false`, wenn die Berechnungen ohne Optimierung für das Reporting durchgeführt werden sollen. Der Standardwert ist `true`.
Best Practice:
 - * Legen Sie `optimizeReporting=false` nur fest, wenn es notwendig ist, um Verarbeitungszeit zu sparen, z.B. wenn Sie eine einzelne Regel oder eine sequentielle Reihe von mehreren POVs ausführen.
 - * Wenn Sie mehrere gleichzeitige Berechnungsjobs ausführen, legen Sie `optimizeReporting=true` für alle Jobs fest. Nur der letzte Job, der abgeschlossen wird, führt die Aggregation durch, vermeidet redundante Verarbeitung und verhindert, dass sich Jobs, die ausgeführt werden, verlangsamen.

 **Hinweis:**

Parameterwerte (`true` oder `false`) müssen in Kleinbuchstaben angegeben werden.

- `comment` stellt einen optionalen Kommentar dar, der in doppelte Anführungszeichen gesetzt werden muss.
- `stringDelimiter` ist das in POV-Werten verwendete Trennzeichen. Trennzeichen müssen in doppelte Anführungszeichen gesetzt werden.

Beispiel

```
epmautomate runCalc BksML12 2012_Jan_Actual Jan-2016 isClearCalculated=true
isExecuteCalculations=true isRunNow=true subsetStart=10 subsetEnd=20
ruleSetName="Utilities Expense Adjustment" ruleName="Occupancy Expense
Allocations" exeType="ALL_RULES" comment="Test calculation" stringDelimiter="_"
```

runComplianceReport

Generiert einen Bericht, der in Abstimmungscompliance definiert ist.

Informationen hierzu finden Sie in den folgenden Abschnitten in der Dokumentation *Account Reconciliation verwalten*:

- Anweisungen zum Definieren von Berichten finden Sie unter *Berichte verwenden*.
- Eine Liste vordefinierter Berichte im Modul "Abstimmungscompliance" und die Parameter zu ihrer Erstellung finden Sie unter *Vordefinierte Berichte im Modul "Abstimmungscompliance" generieren*.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

```
epmautomate runComplianceReport FILE_NAME GROUP_NAME REPORT_NAME REPORT_FORMAT
[Param=value] wobei:
```

- `FILE_NAME` ein eindeutiger Dateiname für den Bericht ist, der generiert wird. Wenn ein Bericht mit diesem Namen auf dem Server vorhanden ist, wird er überschrieben. Verwenden Sie den Befehl [downloadFile](#), um diesen Bericht auf einen lokalen Computer herunterzuladen.
- `GROUP_NAME` der Name der Gruppe ist, der der Bericht zugeordnet ist.
- `REPORT_NAME` ein eindeutiger Name für den Bericht ist, der generiert werden soll.
- `REPORT_FORMAT` eines der folgenden Formate für den Bericht ist:
 - PDF

- HTML (für Grafiken und Diagramme nicht unterstützt)
- XLSX (für Grafiken nicht unterstützt)
- CSV
- CSV2

 **Hinweis:**

REPORT_FORMAT CSV lässt die Formatierung von Daten anhand einer Vorlage nicht zu, CSV2 hingegen lässt dies zu. Das Generieren eines CSV2-formatierten Berichts ist im Vergleich zur CSV-Ausgabe zeitaufwendiger.

- `Param=value` optional die erforderlichen Parameter zum Generieren des Berichts identifiziert. Beispiel: Für den Bericht `Balance By Account Type` sind zwei Parameter `Period` mit dem Wert `July 2017` und `Currency Bucket` mit dem Wert `Entered` erforderlich. Sie müssen diese Parameter als `"Period=July 2017"` `"Currency Bucket=Entered"` angeben.

Beispiel

```
epmautomate runComplianceReport "Example_File Name""Reconciliation
Manager" "Balance By Account Type" PDF "Period=July 2017" "Currency
Bucket=Entered"
```

runDailyMaintenance

Startet den Prozess zur täglichen Servicewartung umgehend, anstatt bis zum geplanten täglichen Wartungsfenster zu warten.

Mit diesem Befehl können Sie die Erstellung eines Backup-Snapshots und die Aktualisierung der Umgebung erzwingen. Sie müssen vor dem Ausführen dieses Befehls sicherstellen, dass niemand die Umgebung verwendet. Der tägliche Wartungsplan ist von diesem Befehl nicht betroffen. Sie verwenden diesen Befehl, wenn Sie nicht bis zum nächsten Wartungsfenster warten möchten, damit die Änderungen an der Umgebung wirksam werden, z.B. nach der Anwendung eines One-off-Patches.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate runDailyMaintenance [skipNext=true|false] [-f] wobei:
```

- `skipNext` optional angibt, ob das nächste Vorkommen des täglichen Wartungsprozesses übersprungen werden soll. Der Standardwert lautet "false".
- `-f` optional angibt, ob der Start des Wartungsprozesses ohne Benutzerbestätigung erzwungen werden soll. Wenn Sie die Option `-f` nicht verwenden, werden Sie von EPM Automate zur Bestätigung der Aktion aufgefordert.

Beispiele

- So erzwingen Sie den Start einer außerplanmäßigen täglichen Wartung, ohne die nächste geplante Wartung zu überspringen: `epmautomate runDailyMaintenance -f`
- So erzwingen Sie den Start einer außerplanmäßigen täglichen Wartung und überspringen die nächste geplante Wartung: `epmautomate runDailyMaintenance skipNext=true -f`
- So starten Sie eine außerplanmäßige tägliche Wartung und überspringen die nächste geplante Wartung: `epmautomate runDailyMaintenance skipNext=true -f`

runDataRule

Führt eine Dataload-Regel von Data Management basierend auf der Start- und Endperiode aus, und importiert oder exportiert die von Ihnen angegebenen Optionen.



Hinweis:

Dieser Befehle kann nicht verwendet werden, um die direkte Dataload-Integration aus Datenquellen in Oracle Enterprise Performance Management Cloud auszuführen. Mit EPM Integration Agent können Sie direkte Dataloads integrieren. Ausführliche Informationen finden Sie unter [Direkten Dataload mit EPM-Integrations-Agent durchführen](#) in der Dokumentation *Komponente "Datenintegration" für Oracle Enterprise Performance Management Cloud verwalten*.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate runDataRule RULE_NAME START_PERIOD END_PERIOD IMPORT_MODE EXPORT_MODE [FILE_NAME]` wobei:

- `RULE_NAME` der Name einer in Data Management definierten Dataload-Regel ist. Sie müssen den Regelnamen in Anführungszeichen setzen, wenn er Leerzeichen enthält.
- `START_PERIOD` die erste Periode ist, für die Daten geladen werden sollen. Dieser Periodenname muss in der Periodenzuordnung von Data Management definiert sein.

- *END_PERIOD* die letzte Periode ist, für die Daten geladen werden sollen (bei Dataloads für mehrere Perioden). Geben Sie zum Laden einer einzelnen Periode dieselbe Periode als Startperiode ein. Dieser Periodenname muss in der Periodenzuordnung von Data Management definiert sein.
- *IMPORT_MODE* bestimmt, wie die Daten in Data Management importiert werden.
Bei den Einstellungen für den Importmodus muss die Groß-/Kleinschreibung beachtet werden. Folgende Werte werden akzeptiert:
 - *APPEND*, um die vorhandenen POV-Daten in Data Management hinzuzufügen.
 - *REPLACE*, um die POV-Daten zu löschen und mit den Daten aus der Datei zu ersetzen.
 - *RECALCULATE*, um die Daten neu zu berechnen.
 - *NONE*, um den Datenimport in die Staging-Tabelle von Data Management zu überspringen.
- *EXPORT_MODE* bestimmt, wie die Daten in die Anwendung exportiert werden.
Bei den Einstellungen für den Exportmodus muss die Groß-/Kleinschreibung beachtet werden. Folgende Werte werden akzeptiert:
 - *STORE_DATA*, um die Daten in der Staging-Tabelle der Komponente Datenmanagement mit den vorhandenen Daten zusammenzuführen. Verwenden Sie in den Datenmanagement-Jobs zum Laden von Metadaten immer diese Exportoption.
 - *ADD_DATA*, um die Daten in der Staging-Tabelle der Komponente Datenmanagement der Anwendung hinzuzufügen.
 - *SUBTRACT_DATA*, um die Daten in der Staging-Tabelle der Komponente Datenmanagement von den vorhandenen Daten zu subtrahieren.
 - *REPLACE_DATA*, um die POV-Daten zu löschen und mit den Daten in der Staging-Tabelle von Data Management zu ersetzen. Die Daten werden für Szenario, Version, Jahr, Periode und Entity gelöscht.
 - *NONE*, um den Datenexport aus der Komponente Datenmanagement in die Anwendung zu überspringen.

 **Hinweis:**

Bei Financial Consolidation and Close werden nur die folgenden Exportmodi unterstützt:

- MERGE, um die Daten in der Staging-Tabelle von Datenmanagement mit den vorhandenen Daten zusammenzuführen.
- REPLACE, um Einträge aus der DM-Staging-Tabelle zu entfernen und durch Einträge aus Dataload zu ersetzen.
- NONE, um den Datenexport aus Data Management in die Anwendung zu überspringen.

Bei Oracle Fusion Cloud als Ziel werden nur diese Exportmodi unterstützt:

- MERGE, um die Daten in der Staging-Tabelle von Datenmanagement mit den vorhandenen Daten zusammenzuführen.
- NONE, um den Datenexport aus Data Management in die Anwendung zu überspringen.

- *FILE_NAME* ist ein optionaler Dateiname. Wenn Sie keinen Dateinamen angeben, importiert EPM Automate die Daten in der Datei, deren Name in der Dataload-Regel angegeben ist. Diese Datei muss im Inbox-Ordner oder in einem zugehörigen Unterordner verfügbar sein.

Wenn Sie Bank Administration Institute (BAI)-Formatdateien für Account Reconciliation laden, geben Sie für diesen Parameter keinen Wert an. Sie müssen immer den Dateinamen für das Laden von BAI-Dateien in der Definition für die Datenregel angeben.

 **Hinweis:**

Wenn in der Datenregel ein Pfad angegeben ist, geben Sie im Befehl nicht den Dateipfad an, sondern nur den Dateinamen. Wenn in der Datenregel kein Pfad angegeben ist, geben Sie den vollständigen Pfad zur Datendatei an.

Beispiele

- Import mit mehreren Perioden:

```
epmautomate runDataRule VisionActual Mar-15 Jun-15 REPLACE STORE_DATA inbox/Vision/GLActual.dat
```
- Import mit einer Periode:

```
epmautomate runDataRule "Vision Actual" Mar-15 Mar-15 REPLACE STORE_DATA inbox/Vision/GLActual.dat
```

runDMReport

Erstellt einen Data Management-Bericht und speichert ihn im Ordner `outbox/reports`.

Der Name des generierten Berichts basiert auf der ID des Data Management-Jobs, der den Bericht generiert, und dem Berichtsformat. Beispiel: Wenn die Berichtsjob-ID "2112" lautet und Sie PDF als Berichtsausgabeformat angeben, lautet der Berichtsname `2112.pdf`. Der Berichtsname wird in der Konsole angezeigt, nachdem der Bericht generiert wurde. Sie

können den Berichtsnamen auch in Datenmanagement in der Registerkarte "Prozessdetails" oder über den Befehl [listFiles](#) identifizieren.

Verwenden Sie den Befehl [downloadFile](#), um den Bericht auf einen lokalen Computer herunterzuladen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

`epmautomate runDMReport REPORT_NAME PARAMETER=Value "Report Output Format=[PDF|HTML|XLS|XLSX]"` wobei Folgendes gilt:

- `REPORT_NAME` ist der Name der Datenmanagement-Berichtsvorlage, mit der der Bericht generiert wird.
- `PARAMETER=Value` gibt Berichtsparemeter und die zugehörigen Werte an. Sie geben so viele Parameter an, wie es für das Format `PARAMETER=Value` erforderlich ist. Die Liste der erforderlichen Parameter hängt vom Bericht ab, den Sie generieren möchten.

Hinweis:

Berichtslaufzeitparameter werden definiert, wenn Sie Ihre Berichte entwerfen. Zum Ausführen dieses Befehls müssen Sie diese Parameter und Werte generieren und von der Registerkarte "Workflow" in EPM Automate kopieren. Um Laufzeitparameter eines Berichts zu generieren, klicken Sie in der Registerkarte "Workflow" von Data Management auf **Berichtsausführung**, und wählen Sie anschließend unter **Berichtsgruppe** eine Gruppe aus. Wählen Sie den Bericht aus, für den die Parameter generiert werden sollen, und klicken Sie auf **Berichtsskript erstellen**. Optional können Sie Berichtsparemeterwerte angeben, ein Ausgabeformat auswählen und anschließend auf **OK** klicken. Verwenden Sie die Parameter, die unter **Berichtsskript generieren** angezeigt werden, um Laufzeitparameter und Werte zum Generieren des Berichts anzugeben.

- `Report Output Format` gibt das Berichtsausgabeformat an. Zu den gültigen Optionen zählen PDF, HTML, XLS und XLSX. Das Standardberichtsformat ist PDF.

Beispiel

```
epmautomate runDMReport "TB Current Location By Target Acct (Cat,Per)"
"Period=Jul 14" "Category=Forecast" "Location=FCSTtoVISCONSOL1" "Rule
Name=FCSTtoVISCONSOL1" "Report Output Format=HTML"
```

runIntegration

Führt einen Data Integration-Job aus, um Daten in einen Oracle Enterprise Performance Management Cloud-Geschäftsprozess zu importieren oder aus einem Geschäftsprozess in ein externes System zu exportieren.

Dieser Befehl ersetzt den veralteten Befehl [runDataRule](#). Oracle empfiehlt, dass Sie diesen Befehl anstelle des Befehls [runDataRule](#) verwenden.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser

Verwendung

```
epmautomate runIntegration JOB_NAME importMode=Append|Replace|"Map and Validate"|"No Import"|Direct exportMode=Merge|Replace|Accumulate|Subtract|"No Export"|Check periodName={PERIOD_NAME} [inputFileName=FILE_NAME] [PARAMETERS]
```

- Bei Integrationen im Standardmodus müssen Sie die Werte für `importMode`, `exportMode` und `periodName` angeben.
- Bei Integrationen im Schnellmodus müssen Sie den Wert für `exportMode` angeben.
- Bei Parameternamen und -werten wird zwischen Groß- und Kleinschreibung unterschieden.

In diesem Befehl gilt Folgendes:

- `JOB_NAME` ist der Name eines Integrationsjobs, der in Data Integration definiert ist.
- `importMode` bestimmt, wie die Daten in Data Integration importiert werden. Zulässige Importmodi:
 - `Append`, um die vorhandenen POV-Daten in Data Integration hinzuzufügen.
 - `Replace`, um die POV-Daten zu löschen und durch die Daten aus der Datei zu ersetzen.
 - `Map and Validate`, um den Datenimport zu überspringen, die Daten aber mit aktualisierten Zuordnungen und Logikaccounts neu zu verarbeiten.
 - `No Import`, um den Datenimport in die Staging-Tabelle von Data Integration zu überspringen.
- `exportMode` bestimmt, wie die Daten in der Zielumgebung geladen werden. Bei Integrationen im Schnellmodus können Sie `Check` oder `No Export` nicht als Wert des Parameters `exportMode` verwenden. Folgende Exportmoduswerte sind zulässig:
 - `Merge`, um die vorhandenen Daten zu aktualisieren und neue Daten hinzuzufügen.
 - `Replace`, um die vorhandenen Daten im POV zu löschen und neue Daten zu laden. Im Standardmodus werden die Daten für Szenario-, Version-, Year-, Period- und Entity-Dimensionen gelöscht. Im Schnellmodus werden Daten für Year-, Period- und

Entity-Dimensionen gelöscht. Sie können benutzerdefinierte zu löschende Regionen für beide Modi definieren.

- `Accumulate`, um die Daten den vorhandenen Daten hinzuzufügen. Anwendbar für `Planning`, `Planning Modules`, `Financial Consolidation and Close`, `Tax Reporting`, `Profitability and Cost Management` und `Enterprise Profitability and Cost Management`.
- `Subtract`, um die Daten vom vorhandenen Saldo abzuziehen. Dieser Wert gilt für `Profitability and Cost Management` und `Enterprise Profitability and Cost Management`.

Für Integrationen im Schnellmodus gilt:

- * Sie dürfen nicht `Check` oder `No Export` als Parameterwert verwenden.
- * Für `Planning`, `Planning Modules` und `Financial Consolidation and Close` sind nur die Werte `Replace`, `Merge` und `Accumulate` gültig.

- `No Export`, um den Datenexport zu überspringen. Verwenden Sie diesen Modus, um Daten zur Prüfung in die Staging-Tabelle zu laden, bevor sie in der Zielanwendung geladen werden.
- `Check`, um nur eine Datenvalidierungsprüfung auszuführen.

Bei Oracle Fusion Cloud als Ziel werden nur diese Exportmodi unterstützt:

- * `MERGE`, um die Daten in der Staging-Tabelle von Data Integration mit den vorhandenen Daten zusammenzuführen.
- * `NONE`, um den Datenexport aus Data Integration in die Anwendung zu überspringen.

- `periodName` ist der Name der Perioden oder Periodenbereiche (jeweils in geschweifte Klammern eingeschlossen), für die Daten importiert oder exportiert werden sollen. Zulässige Namenskonventionen für Perioden:
 - Geben Sie für Ladevorgänge einzelner Perioden den Periodennamen in geschweiften Klammern an. Beispiel: `{Jan-21}`
 - Geben Sie für Ladevorgänge mit mehreren Perioden die Namen der Start- und Endperiode in geschweiften Klammern an. Beispiel: `{Jan-21}{Mar-21}` (zum Laden der Daten für alle Perioden von Januar 21 bis März 21)
 - **Für `Planning`, `Planning Modules`, `Financial Consolidation and Close`, `FreeForm` und `Tax Reporting`:** Sie können den Namen und das Jahr der Periode des Geschäftsprozesses im Format `{Jan#FY21}{Mar#FY21}` angeben, um Daten für alle Perioden von Januar 21 bis März 21 zu laden. Der Periodenname muss in geschweiften Klammern angegeben werden.
 - * Einzelne Periode: Bezieht sich auf den Periodennamen der Komponente Datenmanagement für eine einzelne in der Periodenzuordnung definierte Periode.
 - * Mehrfachperiode: Bezieht sich auf einen Ladevorgang für mehrere Perioden. Der Parameter wird im Format `{Month-Year}{Month-Year}` angegeben. Beispiel: `{Jan-20}{Mar-20}` gibt einen Ladevorgang für mehrere Perioden von Januar 20 bis März 20 an.
 - * Planning-Periodenname: Bezieht sich auf einen Planning-Periodennamen im Format `{Month#Year}`. Beispiel: `{Jan#FY20}{Mar#FY20}`. Bei dieser Konvention müssen Sie keine Data Integration-Periodennamen angeben.

Stattdessen geben Sie die Elementnamen für die Year- und Scenario-Dimensionen an.

Dieser Parameter wird in den Geschäftsprozessen Planning, Tax Reporting und Financial Consolidation and Close unterstützt. Er funktioniert sowohl für Ihre Serviceanwendungen als auch für Cloud-Deployments, die von On-Premise-Datenquellen abgeleitet wurden.

Diese Konvention eignet sich bei Auslösung durch ein EPM Cloud-Groovy-Skript, indem die Namen von Jahres- und Periodenelementen erfasst werden. Die Periodenzuordnung der Anwendung oder die globale Periodenzuordnung muss mit dem Jahr und Monat in den Zielwerten der Periodenzuordnung vorhanden sein.

- * Substitutionsvariable: Dabei handelt es sich um eine Erweiterung des oben genannten Planning-Periodennamensformats. Dabei kann eine Substitutionsvariable anstelle der tatsächlichen Jahres- und Monatelementnamen im Format `{Month#&CurYr}{&FestMonth#&CurYr}` angegeben werden. Beispiel: `{Jan#&CurYr}{&FestMonth#&CurYr}`. Tatsächliche Elementnamen können auch mit Substitutionsvariablen kombiniert werden.

Dieses Format wird in den Geschäftsprozessen Planning, Tax Reporting und Financial Consolidation and Close unterstützt.

Die Periodenzuordnung der Anwendung oder die globale Periodenzuordnung muss in Data Integration der Umgebung vorhanden sein, in der dieser Befehl ausgeführt wird. Dabei müssen Jahres- und Monatswerte in den Zielwerten der Periodenzuordnung verfügbar sein. In diesem Fall beziehen sich Jahr und Monat auf den aktuellen Wert der Substitutionsvariablen bei der Ausführung.

- * Globaler POV: Führt den Dataload für die globale POV-Periode aus. Verwenden Sie das Format `{GLOBAL_POV}`.

Note:

Wenn Sie einen anderen Parameter für die Periodenbenennung als die hier beschriebenen verwenden, wird die Fehlermeldung `Ungültige Eingabe - HTTP 400` angezeigt.

Kopieren Sie den Abschnitt `periodName` aus der REST-API-Dokumentation. Dort werden die verschiedenen Formate wie globaler POV, Substitutionsvariable usw. erläutert.

- `{GLOBAL_POV}` zum Ausführen des Dataloads für die im globalen POV im System oder in den Anwendungseinstellungen in Data Integration definierte Periode.

 **Note:**

Das Format zur Periodenbenennung {Month#Year} wird für Planning, Planning Modules, Financial Consolidation and Close und Tax Reporting unterstützt. Mit dieser Konvention können Sie Elementnamen für die Year- und Scenario-Dimensionen anstelle der Data Integration-Periodennamen angeben. Dieser Ansatz ist nützlich, wenn der Befehl aus einem Groovy-Skript ausgelöst wird, indem die Namen von Jahres- und Periodenelementen erfasst werden.

Die Namenskonvention mit Substitutionsvariablen {Jan#&CurYr} {&FcstMonth#&CurYr} ist eine Erweiterung der oben genannten Namenskonvention für Perioden. Sie können Substitutionsvariablen anstelle der Jahres- und Monatelementnamen angeben, wenn Sie diesen Befehl für Planning, Planning Modules, Financial Consolidation and Close und Tax Reporting ausführen. Elementnamen können auch mit Substitutionsvariablen kombiniert werden.

Die oben genannten Namenskonventionen mit Periodennamen und Substitutionsvariablen funktionieren nur, wenn bereits Anwendungsperiodenzuordnungen oder globale Periodenzuordnungen mit Jahr und Monat in den Zielwerten in Data Integration vorhanden sind.

- `inputFileName` gibt bei dateibasierten Dataloads den Namen der Datei (in der Inbox verfügbar) an, aus der Daten importiert werden sollen. Wenn Sie den Verzeichnisnamen in der Integrationsdefinition angeben, übergeben Sie nur den Dateinamen. Wenn Sie keinen Verzeichnisnamen in der Integrationsdefinition angeben, verwenden Sie das Format `inbox/DIR_NAME/FILE_NAME`. Beispiel: `inbox/GLBALANCES.txt` oder `inbox/EBSGL/GLBALANCES.txt`. Wenn die Datei in den Standardspeicherort der Umgebung hochgeladen wurde, verwenden Sie die Konvention `#epminbox/FILE_NAME`, z.B. `#epminbox/GLBALANCES.txt`, um die Eingabedatendatei anzugeben. Dieser Parameter ist nur bei nativen dateibasierten Dataloads anwendbar. Wenn Sie diesen Parameterwert für dateibasierte Dataloads nicht angeben, importiert dieser Befehl Daten aus der in der Integrationsdefinition angegebenen Datei. Wenn Sie diesen Parameterwert für Dataloads angeben, die nicht dateibasiert sind, ignoriert dieser Befehl ihn.
- `PARAMETERS` identifiziert optional Laufzeitparameter im Format `PARAMETER_NAME="PARAMETER"`. Parameter beinhalten sowohl Quellfilter als auch Zieloptionen.

 **Note:**

Der einzige Parameter, den Sie derzeit für Zielanwendungen vom Typ "Dimension" (Metadaten) verwenden können, ist `"Refresh Database"=Yes|No`.

Beispiele

- Import für einzelne Periode:

```
epmAutomate runIntegration VisionDataLoad importMode=Replace  
exportMode=Merge period="{Mar-15}"
```

- **Import mit mehreren Perioden:**
epmAutomate runIntegration VisionDataLoad importMode=Replace
exportMode=Merge period="{Mar-15} {Jun-15}"
- **Inkrementelle dateibasierte Datenintegration:**
epmAutomate runIntegration IncrementalFileLoad importMode=Replace
exportMode=Merge period="{Jan-20}{Mar-20}" inputFile=File1.txt

runMatchingReport

Generiert einen Bericht, der im Modul Transaktionsabgleich definiert ist.

Eine Liste vordefinierter Berichte im Modul "Transaktionsabgleich" und der Parameter zu ihrer Erstellung finden Sie unter Vordefinierte Berichte im Modul "Transaktionsabgleich" generieren in der Dokumentation *Account Reconciliation verwalten*.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

```
epmautomate runMatchingReport FILE_NAME GROUP_NAME REPORT_NAME REPORT_FORMAT  
[Param=value] wobei:
```

- **FILE_NAME** ein eindeutiger Dateiname für den Bericht ist, der generiert wird. Wenn ein Bericht mit diesem Namen auf dem Server vorhanden ist, wird er überschrieben. Verwenden Sie den Befehl [downloadFile](#), um diesen Bericht auf einen lokalen Computer herunterzuladen.
- **GROUP_NAME** der Name der Gruppe ist, der der Bericht zugeordnet ist.
- **REPORT_NAME** ein eindeutiger Name für den Bericht ist, der generiert werden soll.
- **REPORT_FORMAT** eines der folgenden Formate für den Bericht ist:
 - PDF
 - HTML (für Grafiken und Diagramme nicht unterstützt)
 - XLSX (für Grafiken nicht unterstützt)
 - CSV
 - CSV2

 **Hinweis:**

`REPORT_FORMAT CSV` lässt die Formatierung von Daten anhand einer Vorlage nicht zu, `CSV2` hingegen lässt dies zu. Das Generieren eines `CSV2`-formatierten Berichts ist im Vergleich zur `CSV`-Ausgabe zeitaufwendiger.

- `Param=Value` optional die erforderlichen Parameter zum Generieren des Berichts identifiziert. Beispiel: Geben Sie für den Bericht `Match Type Configuration`, der den Parameter `status` mit dem Wert `approved` erfordert, den Parameter und den Wert als `status=Approved` an.

Beispiel

```
epmautomate runMatchingReport Example_FileName "Transaction Matching"
"Match Type Configuration" HTML "status=Approved"
```

runPlanTypeMap

Die Daten aus einer Block Storage-Datenbank werden in eine Aggregate Storage-Datenbank oder in eine andere Block Storage-Datenbank kopiert. Dazu werden die in einem Job vom Typ `plan type map` (Plantypzuordnung) festgelegten Einstellungen verwendet.

Gilt für

Planning, Planning Modules, FreeForm, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate runPlanTypeMap JOB_NAME [clearData=true|false] wobei:
```

- `JOB_NAME` der in der Anwendung für einen Job vom Typ `plan type map` (Plantypzuordnung) definierte Name ist.
- `clearData` einer optionalen Einstellung entspricht, die angibt, ob die Daten in der Zieldatenbank vor dem Kopiervorgang entfernt werden müssen. Wenn dieser Parameterwert nicht festgelegt ist, wird der Standardwert `true` verwendet.

Parameterwerte (`true` oder `false`) müssen in Kleinbuchstaben angegeben werden.

Beispiel

```
epmautomate runPlanTypeMap CampaignToReporting clearData=false
```

runRuleSet

Startet ein Geschäftsregelset.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser (wenn Regelausführungszugriff erteilt wurde)

Verwendung

`epmautomate runRuleSet RULESET_NAME [PARAMETER=VALUE]` wobei Folgendes gilt:

- `RULESET_NAME` entspricht exakt dem Namen eines Geschäftsregelsets, so wie er in der Umgebung definiert wurde.
- `PARAMETER=VALUE` gibt die optionalen Laufzeitparameter und ihre Werte an, die für die Ausführung des Regelsets erforderlich sind.

Hinweis:

Das Regelset wird für den Plantyp ausgeführt, für den es bereitgestellt wurde.

Mit `PARAMETER=VALUE`-Paaren können Sie so viele Runtime Prompts angeben wie für das Regelset erforderlich sind. Das folgende Beispiel verwendet zwei Runtime Prompts (`Period` und `Entity`) und deren Werte (`Q1` und `USA`).

Standardwerte werden verwendet, wenn Sie keine Werte für Laufzeitparameter angeben. Der Befehl ignoriert Runtime Prompts, die keine genauen Übereinstimmungen mit den im Regelset definierten Prompts darstellen.

Informationen zum Eingeben mehrerer Werte für einen Parameter finden Sie unter [Mehrere Werte für Parameter angeben](#).

Beispiel

```
epmautomate runRuleSet RollupUSSales Period=Q1 Entity=USA
```

runSupplementalDataReport

Generiert relationale Berichte, die Daten aus Supplemental Data Manager anzeigen.

Supplemental Data-Berichte werden in Financial Consolidation and Close und Tax Reporting als Nicht-Konsolidierungsberichte gruppiert. Eine Liste der Berichte, die Sie generieren können, und der zum Generieren erforderlichen Parameter finden Sie im Abschnitt mit der Liste vordefinierter Berichte und Parameter unter Bericht für Financial Consolidation and Close und Tax Reporting generieren in der Dokumentation *REST-API für Oracle Enterprise Performance Management Cloud*.

Gilt für

Financial Consolidation and Close, und Tax Reporting.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Verwendung

`epmautomate runSupplementalDataReport FILE_NAME GROUP_NAME REPORT_NAME REPORT_FORMAT [Param=value]` wobei:

- `FILE_NAME` ein eindeutiger Dateiname für den Bericht ist.
- `GROUP_NAME` der Name der Gruppe ist, der der Bericht zugeordnet ist.
- `REPORT_NAME` ein eindeutiger Name für den Bericht ist, der generiert werden soll.
- `REPORT_FORMAT` eines der folgenden Formate für den Bericht ist:
 - PDF
 - HTML (für Grafiken und Diagramme nicht unterstützt)
 - XLSX (für Grafiken nicht unterstützt)
 - CSV
 - CSV2

`REPORT_FORMAT CSV` lässt die Formatierung von Daten anhand einer Vorlage nicht zu, `CSV2` hingegen lässt dies zu. Das Generieren eines `CSV2`-formatierten Berichts ist im Vergleich zur `CSV`-Ausgabe zeitaufwendiger.

- `Param=value` optional die erforderlichen Parameter zum Generieren des Berichts identifiziert. Beispiel: Um den Bericht `Gefährdete Aufgaben` zu generieren, der einen `schedule name` mit dem Wert `monthly` und eine `period` mit dem Wert `Jan` enthalten soll, geben Sie `"schedule name"=monthly period=Jan an`.

Beispiel

```
epmautomate runSupplementalDataReport Example_File_name Group1 "At Risk
Tasks" html "schedule name"=monthly period=Jan
```

runTaskManagerReport

Generiert relationale Berichte, die Daten aus Task Manager anzeigen.

Task Manager-Berichte werden in Financial Consolidation and Close und Tax Reporting als Nicht-Konsolidierungsberichte gruppiert.

Eine Liste der Berichte, die Sie generieren können, und der zum Generieren erforderlichen Parameter finden Sie im Abschnitt mit der Liste vordefinierter Berichte und Parameter unter Bericht für Financial Consolidation and Close und Tax Reporting generieren in der Dokumentation *REST-API für Oracle Enterprise Performance Management Cloud*.

Gilt für

Financial Consolidation and Close, und Tax Reporting.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Verwendung

`epmautomate runTaskManagerReport FILE_NAME GROUP_NAME REPORT_NAME REPORT_FORMAT [Param=value]` wobei:

- `FILE_NAME` ein eindeutiger Dateiname für den Bericht ist.
- `GROUP_NAME` der Name der Gruppe ist, der der Bericht zugeordnet ist.
- `REPORT_NAME` ein eindeutiger Name für den Bericht ist, der generiert werden soll.
- `REPORT_FORMAT` eines der folgenden Formate für den Bericht ist:
 - PDF
 - HTML (für Grafiken und Diagramme nicht unterstützt)
 - XLSX (für Grafiken nicht unterstützt)
 - CSV
 - CSV2

Hinweis:

`REPORT_FORMAT CSV` lässt die Formatierung von Daten anhand einer Vorlage nicht zu, `CSV2` hingegen lässt dies zu. Das Generieren eines `CSV2`-formatierten Berichts ist im Vergleich zur `CSV`-Ausgabe zeitaufwendiger.

- `Param=value` optional die erforderlichen Parameter zum Generieren des Berichts identifiziert. **Beispiel:** Um den Bericht `Vorzeitige Aufgaben` zu generieren, der einen `schedule name` mit dem Wert `monthly` und eine `period` mit dem Wert `Jan` enthalten soll, geben Sie `"schedule name"=monthly period=Jan` an.

Beispiel

```
epmautomate runTaskManagerReport Example_File_name Group1 "Early Tasks" PDF
"schedule name"=monthly period=Jan
```

sendMail

Sendet eine E-Mail mit der Option, Dateien aus Oracle Enterprise Performance Management Cloud anzuhängen.

Sie können diesen Befehl in Skripte einfügen, um Benutzer über verschiedene Bedingungen zu benachrichtigen oder um Berichte zu senden.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Narrative Reporting, Oracle Enterprise Data Management Cloud, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate sendMail ToAddress Subject [Body="MessageBody"] [Attachments=FILE1,FILE2]`, wobei Folgendes gilt:

- `ToAddress` gibt durch Semikolon getrennte E-Mail-Adressen von Empfängern in doppelten Anführungszeichen an. Beispiel:
"jdoe@example.com;jane.doe@example.com".
- `Subject` gibt den Betreff der E-Mail an.
- `Body="MessageBody"` (optional) ist der Inhalt der E-Mail. Wenn dieser Parameter nicht angegeben ist, enthält die E-Mail keinen Text.

Note:

Verwenden Sie zum Formatieren des Nachrichtentexts gültige HTML-Tags, um das gewünschte E-Mail-Format zu erstellen. Der gesamte Nachrichtentext (einschließlich HTML-Tags) muss als eine Zeile angegeben sein und darf keine Zeilenvorschubzeichen enthalten. Informationen hierzu finden Sie unter [Beispiele](#).

- `Attachments` (optional) gibt eine durch Komma getrennte Liste der in EPM Cloud verfügbaren Dateien an, die an die E-Mail angehängt werden sollen. Beispiel:
outbox/errorFile.txt,inbox/users.csv.

Note:

- Verwenden Sie das Sternchen (*) als Platzhalter für ein Zeichen im Dateinamen. Beispiel: Geben Sie `outbox/user*.csv` an, um alle Dateien in der Outbox anzuhängen, die Dateinamen mit fünf Buchstaben haben und dem Muster entsprechen.
- Sie können abgesehen von Snapshots alle Dateien, die mit dem Befehl [listFiles](#) aufgeführt werden, als E-Mail-Anhang anhängen. Die Größe des Anhangs darf 10 MB nicht überschreiten.

Beispiele

- **Nicht formatierte E-Mail:** `epmautomate sendMail "jdoe@example.com;jane.doe@example.com" "Data Load Process Failed" Body="Data Load 1 Failed" Attachments=outbox/Errorfile.txt,outbox/Errofile2.txt`
- **Formatierte E-Mail:** `epmautomate sendMail jdoe@example.com "Send Formatted Email" "Body=<!DOCTYPE html><html><body><h1>EpmAutomate Email Formatting</h1><p>Hi,</p><p>Test Allocation Rules, Volume, and SPT data were loaded into FY22_Feb_Actual_Version POV.</p><p>Check the`

```
attachment for details.</p></body></html>" Attachments=outbox/  
loadResults.txt
```

setApplicationAdminMode

Versetzt die Anwendung in den Administratormodus, sodass der Zugriff auf die Anwendung auf Serviceadministratoren beschränkt ist.

Dieser Befehl ist nützlich, um zu verhindern, dass Benutzer an der Anwendung arbeiten, wenn Serviceadministratoren administrative Vorgänge durchführen. Die Anwendung bleibt im Administrationsmodus, bis Sie sie wieder zurücksetzen, damit alle Benutzer darauf zugreifen können.



Note:

Dieser Befehl ersetzt den Befehl [applicationAdminMode](#), der veraltet ist, aber noch nicht aus EPM Automate entfernt wurde.

Verwenden Sie den Befehl [getApplicationAdminMode](#), um den aktuellen Status der Umgebung zu prüfen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Account Reconciliation, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate setApplicationAdminMode true|false
```

Geben Sie bei diesem Befehl `true` an, um die Anwendung in den Administrationsmodus zu versetzen, und `false`, um sie in den Normalmodus zurückzusetzen, damit alle Benutzer darauf zugreifen können.

Beispiele

- Anwendung in den Administrationsmodus versetzen:

```
epmautomate setApplicationAdminMode true
```
- Anwendung in den normalen Betrieb zurückversetzen:

```
epmautomate setApplicationAdminMode false
```

setDailyMaintenanceStartTime

Legt die Zeit (UTC oder eine andere Zeitzone) fest, zu der die tägliche Wartung der Umgebung gestartet wird.

Um sicherzustellen, dass die Verwendung dieses Befehls nicht mit der Anforderung von Oracle zur Erstellung von Backups in Konflikt steht, wird die Startzeit für die Wartung durch

diesen Befehl nicht geändert, wenn die tägliche Wartung in den letzten 36 Stunden nicht ausgeführt wurde.



Hinweis:

Serviceadministratoren, die derzeit über einen Browser bei der Umgebung angemeldet sind, sehen die neue Startzeit der täglichen Wartung nur, wenn sie sich abmelden und erneut anmelden.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate setDailyMaintenanceStartTime StartTime`, wobei *StartTime* die Zeit (im Format HH:00 mit 24 Stunden) ist, zu der die Wartung gestartet werden soll. Optional kann eine Zeitzone angegeben werden. Der zulässige Wertebereich für die Startzeit beträgt 00:00 - 23:00. Wenn die Startzeit nicht in UTC festgelegt werden soll, geben Sie eine gültige Standardzeitzone an. Beispiel: "14:00 America/Los_Angeles" für 14:00 Uhr Pazifische Normalzeit.

Beispiele

- Startzeit für tägliche Wartung auf 14:00 UTC festlegen:
`epmautomate setDailyMaintenanceStartTime 14:00`
- Startzeit für tägliche Wartung auf 14:00 Pazifische Normalzeit festlegen:
`epmautomate setDailyMaintenanceStartTime "14:00 America/Los Angeles"`

setDemoDates

Aktualisiert interne Demodaten von Oracle bei Bedarf.

Verwenden Sie diesen Befehl nur beim Installationssetup mit internen Demodaten von Oracle.

Nur Account Reconciliation: Dieser Befehl setzt die Datumsangaben für alle Abstimmungen zurück, die den `Demo Code`-Attributwerten von `setdemodates` oder `setdemodatesnostatuschange` zugeordnet wurden. Dieser Befehl führt Abstimmungen für bis zu 12 Perioden aus, eine aktuelle Periode und 11 vorherige (historische) Perioden. Wenn Abstimmungen aus mehr als zwei Perioden mit dem Attribut `Demo Code` getaggt sind, behandelt der Befehl diese Perioden als vorherige Periode. Abstimmungen, die diesen Attributwert nicht enthalten, sind nicht betroffen.

- Wenn der Wert `setdemodates` lautet, setzt der Befehl die Datumsangaben für die Abstimmung auf Grundlage des angegebenen Datums und eines zufälligen Status zurück.

- Wenn der Wert `setdemodatesnostatuschange` lautet, setzt der Befehl die Datumsangaben für die Abstimmung auf Grundlage des angegebenen Datums zurück, ohne den Abstimmungsstatus zu ändern.

Nur bei Financial Consolidation and Close und Tax Reporting: Dieser Befehl setzt das Start- und Enddatum von Aufgaben und andere zugehörige Datumsinformationen zurück, damit die Aufgaben für eine Demo ansprechend aussehen. Er berechnet die neuen Datumsangaben für Aufgaben basierend auf dem Wert des Attributs `SETDEMOMDATES-`, das im Zeitplan der Aufgabe festgelegt ist, zusammen mit dem Wert von `Demo Date`, den Sie angeben. Wenn der `Demo Date`-Wert nicht angegeben ist, verwendet der Befehl das heutige Datum, um die neuen Aufgabendaten zu berechnen.

 **Hinweis:**

Aufgaben in Zeitplänen, die keinen `SETDEMOMDATES-`-Wert haben, sind nicht betroffen.

Basierend auf dem von Ihnen angegebenen `Demo Date` schiebt dieser Befehl alle mit der Aufgabe verknüpften Datumsangaben nach vorne. Dies umfasst die Kernlaufzeitdaten (Startdatum, Enddatum usw.) sowie zusätzliche Datumsangaben, einschließlich Historiendaten, individuellen Workflowfähigkeitsdaten und Startdatum (aktuell). Der Aufgabenstatus ist nicht betroffen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Sales Planning und Strategische Personalplanung

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

`epmautomate setDemoDates [DEMO_DATE]`, wobei `DEMO_DATE` ein optionales Datum im Format `YYYY-MM-DD` ist. Wenn Sie diesen Wert nicht angeben, werden die Abstimmungen auf das aktuelle Datum zurückgesetzt.

Beispiel

```
epmautomate setDemoDates 2020-02-15
```

setEJJournalStatus

Legt in Financial Consolidation and Close das Aktivierungsergebnis für Enterprise-Journale aus dem ERP-System fest. Mit diesem Befehl können Sie den Aktivierungsstatus von Journalen mit dem Status `Aktivierung in Verarbeitung` unabhängig von ihrem Workflowstatus aktualisieren.

Dieser Befehl verwendet eine CSV-Datei zum Identifizieren des Status des Imports in das ERP-System. Mit dem Befehl [uploadFile](#) können Sie die Importdatei in die Umgebung hochladen. Die CSV-Datei weist das folgende Format auf:

```
Year,Period,Journal ID,Posting Status,Message
2020,Dec,1000001021,Posted,"SUCCESS"
2020,Dec,1000001022,Failed,"Row Header No: 2,10000415 - Linked value 6
does not exist Application-defined or object-defined error 65171"
2020,Dec,1000001022,Failed,"Row Header No: 7,10000415 - Z_ECS_MSG
(001)Enter a valid account number"
2020,Dec,1000001022,Failed,"Row Header No: 7,10000415 - Z_ECS_MSG
(002) Enter a valid cost center"
```

Die Spalte "Message" ist optional und kann ausgelassen werden.

Mit diesem Befehl werden keine Enterprise-Journaldaten aus Financial Consolidation and Close exportiert bzw. in das ERP-System importiert.

Gilt für

Financial Consolidation and Close

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate setEJJournalStatus FILE_NAME.csv`, wobei *FILE_NAME* die CSV-Datei angibt, die den Status des Imports in das ERP-System enthält.

Beispiel

```
epmautomate setEJJournalStatus JournalStatus.csv
```

setEncryptionKey

Legt einen benutzerdefinierten Verschlüsselungsschlüssel für den Datenbankzugriff fest.

Bei Verwendung dieses Befehls wird eine BYOK-(Bring Your Own Key-)Lösung für Kunden bereitgestellt, mit der sie Oracle Enterprise Performance Management Cloud in ihr Standard-Schlüsselmanagementsystem einschließen können.

Der benutzerdefinierte Verschlüsselungsschlüssel wird nach der nächsten täglichen Wartung der Umgebung wirksam. Sie können ihn sofort aktivieren, indem Sie den Befehl [resetService](#) ausführen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate setEncryptionKey key=key` wobei `key` eine benutzerdefinierte Zeichenfolge beliebiger Länge ist, die als Verschlüsselungsschlüssel verwendet werden soll.

Beispiele

- Verschlüsselungsschlüssel festlegen: `epmautomate setEncryptionKey key=se!m+a2J`
- Verschlüsselungsschlüssel entfernen: `epmautomate setEncryptionKey key=`

setEssbaseQryGovExecTime

Legt die maximale Zeit (in Sekunden) fest, die eine Oracle Essbase-Abfrage zum Abrufen und Ausgeben von Informationen verwenden darf, bevor die Abfrage beendet wird.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Profitability and Cost Management, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate setEssbaseQryGovExecTime TIME`, wobei `TIME` die Anzahl der Sekunden angibt, nach denen Essbase-Abfragen beendet werden. Dieser Wert muss eine ganze Zahl bis zu einem Höchstwert von 70.000 sein.

Oracle empfiehlt, diesen Wert nicht auf 0 (null) zu setzen, um zu verhindern, dass Essbase-Abfragen unbegrenzt ausgeführt werden.

Beispiel

```
epmautomate setEssbaseQryGovExecTime 600
```

setIdleSessionTimeout

Ändert das Sessiontimeout (in Minuten) der Umgebung Oracle Enterprise Performance Management Cloud. Das neue Sessiontimeout wird nach der nächsten täglichen Wartung der Umgebung aktiv. Mit diesem Befehl können Sie einen anderen Wert für das standardmäßige Sessiontimeout (75 Minuten) festlegen. Wenn eine Session für die mit diesem Befehl festgelegte Zeitdauer inaktiv ist, werden Benutzer zur Anmeldeseite umgeleitet.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate setIdleSessionTimeout MINUTES`, wobei "MINUTES" das neue Sessiontimeout nach Inaktivität (Minimum 15 und Maximum 150) in Minuten angibt.

Beispiel

```
epmautomate setIdleSessionTimeout 30
```

setIPAllowlist

Konfiguriert für OCI-Umgebungen (2. Generation) die Ausnahmeliste von IP-Adressen und Classless Inter-Domain Routings (CIDRs), die auf Oracle Enterprise Performance Management Cloud zugreifen dürfen. Mit diesem Befehl können Sie IPv4-Adressen und CIDRs hinzufügen oder entfernen.

Dieser Befehl bietet eine Selfservicemethode zum Konfigurieren einer Ausnahmeliste für EPM Cloud-Umgebungen, die auf OCI (2. Generation) gehostet werden.



Note:

Dieser Befehl kann nicht dazu verwendet werden, Ausnahmelisten in Classic-Umgebungen zu konfigurieren. Bei Classic-Umgebungen können Sie auf dem Bildschirm "Servicedetails" von "Meine Services (Classic)" Ausnahmelisten- oder Ablehnungslistenregeln erstellen, um den Benutzerzugriff auf EPM Cloud-Umgebungen zu steuern.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmAutomate setIPAllowlist add|remove FILE_NAME.txt`. Dabei gilt:

- `add` fügt die IP-Adressen und CIDRs, die in einer Textdatei aufgelistet sind, zur Ausnahmeliste hinzu.
- `remove` löscht die IP-Adressen und CIDRs, die in einer Textdatei aufgelistet sind, aus der Ausnahmeliste.
- `FILE_NAME` ist der Name einer Textdatei, in der die IP-Adressen und CIDRs aufgelistet sind, die der Ausnahmeliste hinzugefügt oder daraus entfernt werden sollen. Die Einträge in der Datei müssen durch Zeilenvorschubzeichen getrennt

sein. Verwenden Sie den Befehl `uploadFile`, um diese Datei in die Umgebung hochzuladen. Jede Zeile in der Datei muss eine IPv4-Adresse oder CIDR im folgenden Format sein:

```
xxx.xxx.xxx.xxx  
xxx.xxx.xxx.xxx/n
```

 **Note:**

- Es werden nur IPv4-IP-Adressen unterstützt.
- Verwenden Sie das CIDR-Format anstelle von einzelnen IP-Adressen, um einen kontinuierlichen Bereich von IP-Adressen festzulegen.
- Um zu verhindern, dass eine Ausnahmeliste den Zugriff von jeder IP-Adresse zulässt, schreiben Sie mit dem Befehl `getIPAllowlist` alle vorhandenen IP-Adressen und CIDRs in eine Datei. Laden Sie die Datei in die Umgebung hoch, und führen Sie dann diesen Befehl mit der Option `remove` aus, wie in diesem Beispiel gezeigt:

```
epmAutomate getIPAllowlist > myRemoveList.txt  
epmAutomate uploadFile myRemoveList.txt  
epmAutomate setIPAllowlist remove myRemoveList.txt
```

Beispiele

- Einige IP-Adressen und CIDRs zu einer Ausnahmeliste hinzufügen:

```
epmAutomate setIPAllowlist add myAddList.txt
```

- Einige IP-Adressen von einer Ausnahmeliste entfernen:

```
epmAutomate setIPAllowlist remove myRemoveList1.txt
```

setManualDataAccess

Gibt an, ob Oracle dazu berechtigt ist, in Notfällen manuell auf die relationalen und Oracle Essbase -Datenbanken einer Umgebung zuzugreifen, wenn eine Umgebung nicht reagiert und noch keine Serviceanforderung des Kunden zur Untersuchung und Bereitstellung der Umgebung vorliegt.

Im Notfall greift Oracle auf einen internen Prozess zurück, bei dem ein leitender Entwickler der oberen Führungsebene nach einem unabhängigen Prüfprozess den manuellen Zugriff auf die relationalen und Essbase-Datenbanken erlaubt. Mit diesem Befehl können Sie den Zugriff von Oracle auf die Datenbanken ohne explizite Genehmigung stoppen. Zusätzlich haben Sie die Option, zu verbieten, dass Oracle in Notfallsituationen manuell auf die relationalen und Essbase-Datenbanken zugreift, auch wenn eine Serviceanforderung offen ist.

Die Einstellung, die Sie mit diesem Befehl angeben, wird sofort wirksam.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate setManualDataAccess Allow|Revoke [disableEmergencyAccess=true|false]`, wobei `disableEmergencyAccess` optional angibt, ob Sie den gesamten manuellen Zugriff auf die relationalen und Essbase-Datenbanken verbieten möchten. Wenn Sie diesen Eigenschaftswert auf `true` setzen, wird verhindert, dass Oracle manuell auf diese Datenbanken zugreift, auch wenn eine Serviceanforderung offen ist. Der Standardwert ist `false`.

Oracle empfiehlt, dass `disableEmergencyAccess=true` nicht festgelegt wird, da Oracle nicht auf die relationalen und Essbase-Datenbanken zugreifen kann, wenn der Zugriff erforderlich ist, um Fehler in einer inaktiven Umgebung zu beheben. Wenn die Umgebung inaktiv ist, können Sie diesen Befehl nicht ausgeben, um Oracle zu ermöglichen, manuell auf diese Datenbanken zuzugreifen.

Beispiele

- Die Berechtigung widerrufen, die für den manuellen Zugriff auf relationale und Essbase-Datenbanken ohne explizite Genehmigung in Notsituationen erteilt wurde:
`epmautomate setManualDataAccess revoke`
- Manuellen Zugriff auf relationale und Essbase-Datenbanken in Notfällen zulassen:
`epmautomate setManualDataAccess allow`
- Manuellen Zugriff auf relationale und Essbase-Datenbanken verhindern, auch wenn eine Serviceanforderung offen ist:
`epmautomate setManualDataAccess revoke disableEmergencyAccess=true`

setPeriodStatus

Legt einen bestimmten Status für eine Periode fest.

Gilt für

Account Reconciliation

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer
Benutzer mit den vordefinierten Rollen Poweruser, Benutzer und Viewer benötigen möglicherweise zusätzliche Anwendungsrollen.

Verwendung

`epmautomate setPeriodStatus PERIOD STATUS` wobei:

- *PERIOD* der Name einer Periode ist.
- *STATUS* OPEN, CLOSED oder LOCKED lautet.

Beispiel

```
epmautomate setPeriodStatus "January 2015" OPEN
```

setSubstVars

Erstellt oder aktualisiert Substitutionsvariablen auf Anwendungs- oder Cube-Ebene.

Sie können diesen Befehl nicht zum Festlegen von mehreren Variablen und/oder Funktionen für Substitutionsvariablen verwenden.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

epmautomate setSubstVars *CUBE_NAME* *SUBSTVAR=VALUE* [*SUBSTVAR=VALUE*] wobei Folgendes gilt:

- *CUBE_NAME* ist der Cube (z.B. Plan1, Plan2), für den die Substitutionsvariable erstellt oder aktualisiert wird. Verwenden Sie anstelle eines Cube-Namens den Wert ALL, um Substitutionsvariablen auf Anwendungsebene festzulegen oder zu aktualisieren.
- *SUBSTVAR* ist der Name der Substitutionsvariable, für die ein Wert festgelegt oder aktualisiert werden soll.
- *VALUE* ist der neue Wert für die Substitutionsvariable.

Beispiele

- Eine Substitutionsvariable auf Anwendungsebene erstellen oder aktualisieren:
epmautomate setSubstVars ALL CurYear=2015 CurPeriod=Jan
- Substitutionsvariablen auf Cube-Ebene erstellen oder aktualisieren: epmautomate setSubstVars Plan2 CurYear=2013 CurPeriod=Jan

setVirusScanOnFileUploads

Ermöglicht in OCI-Umgebungen (2. Generation) das Virenschannen von Dateien vor dem Hochladen in Oracle Enterprise Performance Management Cloud.

Alle OCI-Umgebungen (2. Generation) werden durch ein Antivirusprogramm geschützt. Dieser Befehl bietet zusätzliche Sicherheit, da er das Aktivieren eines Virenschans für Uploaddateien ermöglicht. Das Scannen von Dateien vor dem Hochladen schützt gegen das Hochladen von Viren in die Umgebung.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate setVirusScanOnFileUploads true|false
```

Standardmäßig ist der Virenskan nicht aktiviert (auf `false` gesetzt). Wenn dieser Wert auf `true` gesetzt wird, scannt EPM Cloud alle Uploaddateien. Wenn eine Datei mit einem Virus infiziert ist, wird sie nicht in die Umgebung hochgeladen.

Beispiel

- Virenskan aktivieren: `epmautomate setVirusScanOnFileUploads true`
- Virenskan deaktivieren: `epmautomate setVirusScanOnFileUploads false`

simulateConcurrentUsage

Führt verschiedene gleichzeitige Vorgänge in einer Umgebung aus, wobei Benutzer simuliert werden.

Mit diesem Befehl kann die Performance der Umgebung validiert werden, um zu prüfen, ob die Antwortzeit akzeptabel ist, wenn der Service während bestimmter Vorgänge belastet ist, die von einer bestimmten Anzahl von Benutzern ausgeführt werden. Dieser Befehl kann zum Beispiel verwendet werden, um die Performance zu messen, wenn 50 Benutzer unter Verwendung verschiedener POVs gleichzeitig ein Formular öffnen. Damit werden Selfservice-Belastungstests für Umgebungen ermöglicht.

Mit diesem Befehl wird die Simulation ausgeführt, indem die angegebenen Vorgänge für eine bestimmte Anzahl von Benutzern und Iterationen ausgeführt werden. Dabei werden mehrere Iterationen ausgeführt, um die Mindestdauer, die Höchstdauer und die Durchschnittsdauer eines bestimmten Vorgangs zu berechnen. Der Befehl unterstützt die folgenden Vorgänge zur Ausführung von Belastungstests für die gleichzeitige Verwendung:

- Formulare öffnen
- Formulare speichern
- Geschäftsregeln ausführen
- Datenregeln ausführen
- Ad-hoc-Raster öffnen
- Berichte ausführen
- Bücher ausführen

 **Note:**

Dieser Befehl unterstützt keine Financial Reporting-Berichte und -Bücher, sondern nur Bücher und Berichte aus der Funktion "Berichte" (früherer Name: "Managementberichte").

Dieser Befehl akzeptiert als Eingabe eine ZIP-Datei, die bereits in die Inbox der Umgebung hochgeladen worden sein muss. Die ZIP-Datei muss eine `requirement.csv`-Datei und die Eingabedateien enthalten, die die in der `requirement.csv`-Datei enthaltenen Anwendungsfälle unterstützen. Der Befehl simuliert anschließend die Anwendungsfälle und erstellt einen Bericht, der per E-Mail an einen oder mehrere Empfänger gesendet werden kann.

Verwendungsszenario 1: Akzeptanztest der Anwendungsperformance für 50 Benutzer, die ein Formular gleichzeitig öffnen.

Lösung:

1. Erstellen Sie die Datei `requirement.csv` mit Einträgen ähnlich den folgenden. Es wird angenommen, dass Sie ein Formular namens `Exchange Rates` öffnen möchten, das im Ordner `Library/Global Assumption/` gespeichert ist:

```
# Type of Operation,Artifact Name,Number of Users,Input File,Additional
Info
Open Form, Library/Global Assumption/Exchange
Rates,50,open_form_input.csv,
```

2. Erstellen Sie die Datei `open_form_input.csv` mit dem unter [Eingabedatei für Formulare öffnen](#) angegebenen Format. Die Datei enthält einen Eintrag, der 50 Mal verwendet wird. Wenn Sie dasselbe Formular mit verschiedenen POVs öffnen möchten, muss die Datei so viele Einträge wie die Anzahl der zu verwendenden POVs enthalten.
3. Erstellen Sie die Datei `userVarMemberMapping.csv` mit dem unter [Datei "UserVarMemberMapping.csv" erstellen](#) angegebenen Format.
4. Erstellen Sie eine ZIP-Datei, die die Dateien aus den vorherigen Schritten enthält, und laden Sie sie in die Inbox hoch.
5. Führen Sie den Befehl `simulateConcurrentUsage` aus, wobei Sie die in den vorherigen Schritten erstellte ZIP-Datei als Eingabedatei verwenden.

Verwendungsszenario 2: Simulation der Performance bei Anstieg der saisonalen Verwendung, z.B. am Ende des Finanzjahrs. Annahme: 100 Benutzer speichern ein Formular mit einer Verzögerung von sechzig Sekunden zwischen den einzelnen Benutzern.

Lösung:

1. Erstellen Sie die Datei `requirement.csv` mit Einträgen ähnlich den folgenden. Es wird angenommen, dass Sie ein Formular namens `Accessories Revenue` speichern möchten, das im Ordner `Library/Dashboards/` gespeichert ist:

```
# Type of Operation,Artifact Name,Number of Users,Input File,Additional
Info
Save Form, Library/Dashboards/Accessories Revenue,100,save_form_input.csv,
```

2. Erstellen Sie die Datei `save_form_input.csv` mit dem unter [Eingabedatei für Formulare speichern](#) angegebenen Format.
3. Erstellen Sie die Datei `userVarMemberMapping.csv` mit dem unter [Datei "UserVarMemberMapping.csv" erstellen](#) angegebenen Format.
4. Erstellen Sie eine ZIP-Datei, die die Dateien aus den vorherigen Schritten enthält, und laden Sie sie in die Inbox hoch.
5. Führen Sie den Befehl `simulateConcurrentUsage` aus, wobei Sie die in den vorherigen Schritten erstellte ZIP-Datei als Eingabedatei und die folgenden Eigenschaftswerte verwenden: `iteration=1` und `lagTime=6`.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Strategic Workforce Planning und Sales Planning.

Erforderliche Rollen

Serviceadministrator. Zur Verwendung der `testModes` 0, 1 und 2 benötigen Sie außerdem die Rolle Identitätsdomainadministrator.

Verwendung

```
epmautomate simulateConcurrentUsage INPUT_FILE.zip [iterations=COUNT]
[notificationEmails="EMAIL_ADDRESS"] [testMode=0|1|2|3]
[lagTime=LAG_TIME], wobei Folgendes gilt:
```

- `INPUT_FILE.zip` ist der Name einer ZIP-Datei, in der Ihre Anwendungsfälle identifiziert werden. Bevor Sie diesen Befehl ausführen, verwenden Sie den Befehl [uploadFile](#) (Beispielbefehlssyntax: `epmautomate uploadFile "C:/uploads/INPUT_FILE.zip" inbox`), um diese Datei in die Inbox hochzuladen. Diese ZIP-Datei muss die folgenden Dateien enthalten:
 - Eine CSV-Datei mit den Anwendungsfällen namens `requirement.csv`. In jeder Zeile dieser CSV-Datei werden der Typ des auszuführenden Vorgangs, der Artefaktnamen, die Anzahl gleichzeitiger Benutzer, die Eingabedatei mit den Details zum Vorgang und zusätzliche Informationen zu jedem Anwendungsfall angegeben. Informationen hierzu finden Sie unter [Datei "requirement.csv" erstellen](#).
 - Die Eingabedateien mit Details zu den einzelnen Vorgängen. Informationen hierzu finden Sie in den folgenden Themen:
 - * [Eingabedatei für Formulare öffnen](#)
 - * [Eingabedatei für Formulare speichern](#)
 - * [Eingabedatei für Geschäftsregeln ausführen](#)
 - * [Eingabedatei für Datenregeln ausführen](#)
 - * [Eingabedatei für Ad-hoc-Raster](#)
 - * [Eingabedatei für Bücher ausführen](#)
 - * [Eingabedatei für Berichte ausführen](#)
 - * [Datei "UserVarMemberMapping.csv" erstellen](#)
- `iterations` ist eine positive Zahl, die die Anzahl der Wiederholungen angibt, für die ein in der Datei `requirement.csv` angegebener Anwendungsfall ausgeführt

werden soll, um die Antwortzeit zu messen. Wenn kein Wert angegeben ist, wird der Vorgang nur einmal ausgeführt.

- `notificationEmails` (optional) gibt die E-Mail-Adressen an, an die die Ergebnisse dieser Befehle gesendet werden. Bei Angabe mehrerer E-Mail-Adressen trennen Sie diese durch ein Semikolon. Schließen Sie die Adressliste außerdem in doppelte Anführungszeichen ein. Wenn kein Wert angegeben ist, werden die Ergebnisse nur an den Benutzer gesendet, der den Befehl initiiert hat. Ausführliche Informationen zu diesem Bericht finden Sie unter [Beispiel für einen Bericht zur Simulation der gleichzeitigen Verwendung](#).
- `[testMode]` (optional) gibt den Simulationsmodus für die gleichzeitige Verwendung an. Der Standardwert ist 0. Führen Sie den Befehl anschließend mit Modus 3 aus, um die Simulation so oft wie nötig auszuführen. Führen Sie den Befehl schließlich mit Modus 2 aus, um die simulierten Benutzer zu löschen.

Folgende Werte werden akzeptiert:

- 0: Der Standardsimulationsmodus, in dem der Umgebung simulierte Benutzer hinzugefügt werden, denen die Rolle Serviceadministrator zugewiesen wird. Anschließend wird die Simulation ausgeführt, und die simulierten Benutzer werden gelöscht. Dieser Modus ist nützlich, wenn Sie den Test nur einmal durchführen möchten.
Die simulierten Benutzer haben die folgenden Eigenschaften:
Vorname: testuser1, testuser2 usw.
Nachname: testuser1, testuser2 usw.
E-Mail-Adresse: testuser1@discard.oracle.com, testuser2@discard.oracle.com usw.
Benutzername: testuser1, testuser2 usw.
- 1: Der Umgebung werden simulierte Benutzer hinzugefügt, denen die Rolle Serviceadministrator zugewiesen wird. Die Simulation wird nicht ausgeführt, und die Benutzer werden nicht gelöscht.
- 2: Simulierte Benutzer werden gelöscht. Es werden keine Benutzer erstellt, und die Simulation wird nicht ausgeführt.
- 3: Die Simulation wird unter Verwendung der bereits vorhandenen simulierten Benutzer ausgeführt. Es werden keine Benutzer hinzugefügt oder gelöscht.

Wenn Sie die gleichzeitige Verwendung nur einmal ausführen möchten, verwenden Sie `testMode=0`. So führen Sie eine Reihe von Tests aus:

- Führen Sie den Befehl zunächst mit `testMode=1` aus, um die simulierten Benutzer hinzuzufügen, und weisen Sie ihnen die Rolle Serviceadministrator zu.
- Führen Sie den Befehl anschließend mit `testMode=3` aus, um die Simulation so oft wie nötig auszuführen.
- Führen Sie den Befehl schließlich mit `testMode=2` aus, um die simulierten Benutzer zu löschen.
- `[lagTime]` (optional) gibt die Anzahl der Sekunden an (mindestens 5 Sekunden), die der Befehl zwischen zwei Ausführungen jedes in der Datei `requirement.csv` aufgeführten Anwendungsfalls warten soll. Der Standardwert beträgt 5 Sekunden. Verwenden Sie keine negativen Zahlen (wie -1), Brüche (wie 1/2) oder Dezimalwerte. Nachdem ein Benutzer die Ausführung eines Anwendungsfalls aus der Datei `requirement.csv` initiiert hat, wartet der Befehl für die in diesem Parameter angegebene

Anzahl von Sekunden, bevor er die Ausführung des Anwendungsfalls durch den nächsten Benutzer initiiert. Da Benutzeraktivitäten in der Regel nicht gleichzeitig initiiert werden, hilft die Festlegung dieses Parameters, eine realistischere Simulation der Last einer Umgebung zu erstellen.

Beispiel

```
epmAutomate simulateConcurrentUsage test_simulation.zip iterations=5  
notificationEmails="jane.doe@example.com;john.doe@example.com;example@e  
xample.com" lagTime=6
```

skipUpdate

Fordert an, dass Oracle die monatlichen Updates einer Umgebung für maximal drei aufeinanderfolgende Zyklen überspringt, oder entfernt alle Anforderungen zum Überspringen von Updates, die zuvor mit diesem Befehl eingegeben wurden, sodass die Umgebung entsprechend der Mainline upgedatet wird.

Mit diesem Befehl können Sie auch die Anforderungen zum Überspringen von Updates auflisten, die derzeit für eine Umgebung angegeben sind. Der Status zu übersprungenen Updates erscheint im Aktivitätsbericht (unter Betriebskennzahlen), der erstellt wird, nachdem Sie diesen Befehl zum Überspringen von Updates für eine Umgebung verwendet haben. Informationen hierzu finden Sie unter Betriebskennzahlen in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*.

Wöchentliche Patches und Notfallpatches für den aktuellen Monat, werden, falls verfügbar, weiterhin für die Umgebung angewendet. Für die Monate, für die die Upgradeverzögerung angefordert wird, werden keine Updates vorgenommen.

Updates für eine Umgebung, die ein One-off-Patch ist, können nicht übersprungen werden. Darüber hinaus können keine monatlichen Updates übersprungen werden, die länger als drei Monate nach dem aktuellen Update der Umgebung liegen. Beispiel: Wenn die Umgebung auf Version 23.12 upgedatet wurde, können die monatlichen Updates 24.01, 24.02 und 24.03 übersprungen werden, jedoch nicht 24.04. Ausführliche Informationen dazu, wie Upgradeverzögerungen funktionieren, finden Sie unter Upgradeverzögerungen für Produktionsumgebungen anfordern in der *Dokumentation zu Vorgängen in Oracle Enterprise Performance Management Cloud*.

 **Hinweis:**

Wenn Sie das Update für nur eine Umgebung überspringen (z.B. das Update für die Produktionsumgebung, aber nicht für die Testumgebung), unterscheiden sich die Umgebungen drei Monate lang um drei Versionen. In solch einem Szenario können Sie möglicherweise Snapshots nicht zwischen diesen Umgebungen migrieren. Angenommen, Ihre Test- und Ihre Produktionsumgebung haben derzeit z.B. die Version 23.12, und Sie überspringen nur für die Produktionsumgebung die Updates für die Versionen 24.01, 24.02 und 24.03. Sobald Version 24.03 verfügbar ist, weist Ihre Testumgebung die Version 24.03 auf, während die Produktionsumgebung immer noch Version 23.12 aufweist. In diesem Fall wird die Migration zwischen Ihrer Test- und Ihrer Produktionsumgebung nicht unterstützt,

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate skipUpdate add|remove|list [version=UPDATE_NUMMER  
comment="KOMMENTAR"] wobei Folgendes gilt:
```

- **add** legt eine Anforderung zum Überspringen eines Updates für ein bestimmtes monatliches Update fest. Sie müssen die folgenden Parameter angeben:
 - **version:** Das monatliche Update, das übersprungen werden soll. Sie können ein, zwei oder drei anstehende monatliche Updates überspringen. Wenn die Umgebung sich z.B. im monatlichen Update 23.12 befindet, können Sie die Updates 24.01, 24.02 und/oder 24.03 überspringen. Um drei monatliche Updates zu überspringen, führen Sie den Befehl dreimal aus. Geben Sie jedes Mal ein bestimmtes Update an, das übersprungen werden soll, z.B. `version=24.01`, `version=24.02` und `version=24.03`. In diesem Szenario wird das Update für die Umgebung auf die Mainline beim monatlichen Zyklus 24.04 durchgeführt.

Wenn zwischen dem monatlichen Zyklus, für den das Überspringen angefordert wird, und dem aktuellen monatlichen Zyklus ein Abstand besteht, updatet Oracle die Umgebung planmäßig und überspringt die Updates im angegebenen monatlichen Zyklus. Beispiel: Die Umgebung befindet sich derzeit im monatlichen Update 23.12, und Sie geben an, dass das Update auf die Versionen 24.02 und 24.03 übersprungen werden soll. In diesem Fall wird die Umgebung auf 24.01 upgedatet, und die Updates 24.02 und 24.03 werden übersprungen. Die Umgebung wird beim Update 24.04 auf die Mainline aktualisiert.
 - **comment:** Dieser Text beschreibt den Grund, aus dem ein Update übersprungen werden soll. Kommentare müssen in doppelte Anführungszeichen gesetzt werden.
- **remove** entfernt alle Anforderungen zum Überspringen von Updates, die für die Umgebung angegeben sind, sodass sie bei der nächsten täglichen Wartung auf die Mainline aktualisiert werden kann. Wenn für eine Umgebung mehrere Anforderungen

zum Überspringen von Updates festgelegt wurden, werden mit diesem Befehl alle Anforderungen entfernt.

- `list` zeigt Anforderungen zum Überspringen von Updates an (die Anmelde-ID des Benutzers, der die Anforderung zum Überspringen von Updates eingegeben hat, Kommentar, Version, für die Updates übersprungen werden sollen, und das Datum, an dem die Anforderung eingegeben wurde), die derzeit für die Umgebung festgelegt sind, wie in der folgenden Abbildung dargestellt:

```
c:\Oracle\EPM Automate\bin>epmautomate skipupdate add version=24.01 comment="Some Comment"
skipupdate completed successfully

c:\Oracle\EPM Automate\bin>epmautomate skipupdate add version=24.02 comment="Some Comment"
skipupdate completed successfully

c:\Oracle\EPM Automate\bin>epmautomate skipupdate add version=24.03 comment="Some Comment"
skipupdate completed successfully

c:\Oracle\EPM Automate\bin>epmautomate skipupdate list
skipupdate completed successfully

1] User:  example@example.com | Version: 24.03 | Comments: Some Comment | Timestamp: 2023-11-15T19:17:09Z
2] User:  example@example.com | Version: 24.02 | Comments: Some Comment | Timestamp: 2023-11-15T19:17:01Z
3] User:  example@example.com | Version: 24.01 | Comments: Some Comment | Timestamp: 2023-11-15T19:16:51Z

c:\Oracle\EPM Automate\bin>epmautomate skipupdate remove
skipupdate completed successfully
```

Beispiele

- Überspringen eines Updates anfordern: `epmautomate skipUpdate add version=24.01 comment="We are in the process of closing the quarter"`
- Details zum Überspringen von Updates anzeigen: `epmautomate skipUpdate list`
- Alle Anforderungen zum Überspringen von Updates entfernen: `epmautomate skipUpdate remove`

snapshotCompareReport

Vergleicht zwei Snapshots und erstellt einen Bericht zu diesem Vergleich, in dem Unterschiede bei Berechnungsregeln und Regelsets sowie Eingabefeldern in den Snapshots ermittelt werden. Sie können diesen Bericht zur Behebung von Problemen verwenden, wie z.B.:

- Kürzliche Verminderung der Performance in einer Umgebung. Sie können den vorherigen Snapshot mit dem aktuellen Snapshot vergleichen, um Unterschiede zu ermitteln, die die Verminderung der Performance verursacht haben könnten.
- Sie bemerken Unterschiede im Verhalten oder bei der Performance von zwei Umgebungen, die erwartungsgemäß eigentlich ein identisches funktionales Verhalten oder eine identische Performance aufweisen müssten. In diesem Fall können Sie die Snapshots der beiden Umgebungen vergleichen, um die Unterschiede zwischen ihnen zu verstehen.
- Sie vermuten, dass einige Regeln oder Formulare aus einer Umgebung verschwunden sind. Verwenden Sie diesen Bericht, um die Artefakte, die zuvor vorhanden waren, mit den aktuellen Artefakten zu vergleichen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmAutomate snapshotCompareReport SOURCE_SNAPSHOT TARGET_SNAPSHOT [reportName=REPORT_NAME.html]` wobei:

- `SOURCE_SNAPSHOT` der Name des Snapshots ist, mit dem der Vergleich ausgeführt werden soll. Dieser Bericht enthält Daten zu den Unterschieden bei Regeln, Formularen, Dimensionen und Dimensionselementen in diesem Snapshot.
- `TARGET_SNAPSHOT` ist der Name des Snapshots, der verglichen werden soll.

Note:

- Namen von Snapshots können mit oder ohne ZIP-Erweiterung angegeben werden.
 - Beide Snapshots müssen in der Umgebung verfügbar sein. Verwenden Sie den Befehl [uploadFile](#), [copyFromObjectStorage](#) oder [copySnapshotFromInstance](#), um sie in die Umgebung hochzuladen.
- `REPORT_NAME` optional der Name der Berichtsdatei ist. Standardmäßig lautet der Name des Berichts `SnapshotCompare.html`. Verwenden Sie den Befehl [downloadFile](#), um den Bericht herunterzuladen.

Beispiele

- `epmAutomate snapshotCompareReport "Artifact Snapshot" Backup_22-09-08.zip reportName=Snapshot_Diffs.html`
- `epmAutomate snapshotCompareReport backup_snapshot_22-Aug-08.zip backup_Snapshot_22-Sep-08.zip reportName=Sep_22_snapshot_compare_report.html`

sortMember

Sortiert Elemente der Entity-, Account-, Scenario- und Versions-Dimension von Custom-Dimensionen.

Dieser Befehl ist nützlich, um Dimensionselemente nach dem Laden von Elementen in die Anwendung zu sortieren.

 **Hinweis:**

Sie können diesen Befehl nicht verwenden, um Elemente der Period-, Years- und Currency-Dimensionen zu sortieren.

Gilt für

Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate sortMember Member [type=children|descendants]`
`[order=ascending|descending]` wobei Folgendes gilt:

- *Member* ist der Name des übergeordneten Elements, dessen abhängige oder untergeordnete Elemente sortiert werden sollen.
- *type* gibt optional die zu sortierenden Elemente an. Folgende Werte werden akzeptiert:
 - *descendants* sortiert alle Unterelemente (untergeordnete und abhängige Elemente) des übergeordneten Elements, die Sie als Wert von *Member* angeben.
 - *children*, der Standardwert, sortiert nur Elemente in der Ebene unmittelbar unterhalb des übergeordneten Elements, das Sie als Wert von *Member* angeben.
- *order* gibt optional eine Sortierreihenfolge an. Folgende Werte werden akzeptiert:
 - *aufsteigend*, dies ist die Standardsortierreihenfolge.
 - *descending*

Beispiele

- Sortiert die untergeordneten Elemente der Entity-Dimension in aufsteigender Reihenfolge: `epmautomate sortMember Entity`
- Sortiert alle Unterelemente der Entity-Dimension in absteigender Reihenfolge: `epmautomate sortMember Entity type=descendants order=descending`

unassignRole

Entfernt eine Rolle, die derzeit den Benutzern zugewiesen ist, einschließlich des Benutzers, der diesen Befehl ausführt, deren Anmelde-IDs in der ANSI- oder UTF-8-codierten CSV-Datei enthalten sind, die mit diesem Befehl verwendet wird. Mit diesem Befehl können Sie die Zuweisung einer vordefinierten Rolle oder Anwendungsrolle entfernen.

 **Hinweis:**

Sie können diesen Befehl nicht verwenden, um Anwendungsrollenzuweisungen aus Account Reconciliation- und Profitability and Cost Management-Anwendungen zu entfernen, die die Zuweisung von Anwendungsrollen zu Benutzern nicht unterstützen.

Die CSV-Datei weist das folgende Format auf:

```
User Login
jane.doe@example.com
jdoe
```

Verwenden Sie den Befehl `uploadFile`, um die Datei in die Umgebung hochzuladen.

 **Hinweis:**

Setzen Sie Rollennamen, die Leerzeichen enthalten, in doppelte Anführungszeichen.

Wenn die Befehlsausführung abgeschlossen ist, gibt EPM Automate Informationen zu jedem nicht erfolgreichen Eintrag an die Konsole aus. Lesen Sie diese Informationen, um zu verstehen, warum die Befehlsausführung für einige Einträge in der CSV-Datei nicht erfolgreich war.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

So entfernen Sie vordefinierte Rollenzuordnungen:

- Classic-Umgebungen: Identitätsdomainadministrator und alle vordefinierten Rollen (Serviceadministrator, Poweruser, Benutzer oder Leseberechtigter)
- OCI-Umgebungen: Serviceadministrator oder Identitätsdomainadministrator und alle vordefinierten Rollen (Serviceadministrator, Poweruser, Benutzer oder Leseberechtigter)

So entfernen Sie Anwendungsrollenzuweisungen: Serviceadministrator oder Zugriffskontrollmanager

Verwendung

`epmautomate unassignRole FILE_NAME ROLE wobei:`

- `FILE_NAME` der Name einer CSV-Datei mit den Anmelde-IDs der Benutzer ist, deren Rollenzuweisung entzogen werden soll. Geben Sie die Erweiterung "CSV" in Kleinbuchstaben an.

Bei Benutzeranmeldewerten wird nicht zwischen Groß- und Kleinschreibung unterschieden. Beispiel: `jane.doe@example.com` wird genauso wie `Jane.Doe@Example.com` oder jede andere Variation der Groß-/Kleinschreibung behandelt.

- `ROLE` eine der folgenden Rollen angibt. Bei Rollennamen wird nicht zwischen Groß- und Kleinschreibung unterschieden.
 - Wenn Sie die Zuordnung von Benutzern zu vordefinierten Rollen entfernen, muss `ROLE` eine vordefinierte Rolle angeben, die für den Service gültig ist. Informationen hierzu finden Sie unter Erläuterungen zu vordefinierten Rollen in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*.
 - Wenn Sie die Zuweisung von Benutzern zu Anwendungsrollen entfernen, muss `ROLE` eine Rolle angeben, die zur Anwendung in der aktuellen Umgebung gehört. Anwendungsrollen sind auf der Registerkarte **Rollen** unter Zugriffskontrolle aufgelistet. Beschreibungen der Anwendungsrollen für jeden Geschäftsprozess finden Sie in diesen Themen in der Dokumentation *Zugriffskontrolle für Oracle Enterprise Performance Management Cloud verwalten*:
 - * Planning, Financial Consolidation and Close und Tax Reporting
 - * Enterprise Profitability and Cost Management
 - * Oracle Enterprise Data Management
 - * Profitability and Cost Management
 - * Account Reconciliation

Beispiele

- So heben Sie die Zuweisung für Benutzer von einer vordefinierten Rolle der Identitätsdomain auf:
`epmautomate unassignRole remove_roles.csv "Service Administrator"`
- So heben Sie die Zuweisung für Benutzer von einer Anwendungsrolle auf:
`epmautomate unassignRole example_file.csv "Task List Access Manager"`

updateUsers

Ändert Attribute wie E-Mail-Adresse, Vorname und Nachname von Oracle Enterprise Performance Management Cloud-Benutzern mit den neuen Werten in einer ANSI- oder UTF-8-codierten CSV-Datei, die in die Umgebung hochgeladen wurde.

Verwenden Sie den Befehl [uploadFile](#), um Dateien in eine Umgebung hochzuladen. Alle Spalten in der CSV-Datei sind obligatorisch. Jede Spalte muss einen gültigen Eintrag enthalten. Dieser Befehl validiert diese obligatorischen Werte in jeder Definition und zeigt Fehlermeldungen an, die jeden fehlenden oder ungültigen Wert angeben. Das Eingabedateiformat lautet wie folgt:

```
First Name,Last Name,Email,User Login
Jane,Doe,jane.doe@example.com,jdoe
John,Doe,john.doe@example.com,john.doe@example.com
```

Wenn der Wert für die Benutzeranmeldung in der CSV-Datei mit einem Konto übereinstimmt, das in der Identitätsdomain vorhanden ist, ändert der Befehl den

Benutzeraccount, sodass dieser mit den Werten in der Eingabedatei übereinstimmt. Da Benutzeraccounts für alle Umgebungen gleich sind, die eine Identitätsdomain unterstützen, stehen aktualisierte Benutzerinformationen in allen Umgebungen zur Verfügung, die dieselbe Identitätsdomain verwenden. Auf vordefinierte und anwendungsspezifische Rollen, die dem Benutzer zugewiesen sind, hat dieser Befehl keine Auswirkungen.

 **Note:**

- Mit diesem Befehl können Sie keine Benutzeranmeldewerte ändern.
- Sie sind nicht berechtigt, Ihre eigenen Accountattribute zu ändern.
- Die Eingabedatei mit Multibyte-Zeichen muss die UTF-8-Zeichenkodierung verwenden. Die Verwendung der ANSI-Codierung verursacht Probleme bei der Anzeige von Benutzerinformationen in den Fenstern "Meine Services".

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Strategische Personalplanung und Sales Planning.

Erforderliche Rollen

Identitätsdomainadministrator und alle vordefinierten Rollen (Serviceadministrator, Poweruser, Benutzer oder Leseberechtigter)

Verwendung

`epmautomate updateUsers FILE_NAME`, wobei `FILE_NAME` der Name einer CSV-Datei mit den zu ändernden Benutzerinformationen ist.

Beispiel

```
epmautomate updateUsers update_user_info.csv
```

upgrade

Lädt die neueste Version von EPM Automate automatisch herunter und installiert sie im Hintergrund.

Nachdem Sie den Befehl `login` ausgeführt haben, um eine Session zu initiieren, identifiziert EPM Automate die aktuell installierte Version. Wenn es sich bei der installierten Version nicht um die neueste verfügbare Version handelt, werden Sie von EPM Automate darüber informiert, dass eine neuere Version verfügbar ist.

 **Hinweis:**

Für ein von einem Windows-Administrator bereitgestelltes EPM Automate kann nur dann ein Upgrade durchgeführt werden, wenn der angemeldete Benutzer ein Windows-Administrator ist.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, Benutzer, Viewer

Verwendung

```
epmautomate upgrade
```

Beispiel

```
epmautomate upgrade
```

uploadFile

Lädt eine Datei vom lokalen Computer in den Service hoch. Verwenden Sie diesen Befehl, um Dateien mit Daten, Metadaten, Regeldefinitionen, Dimensionsdefinitionen, zugeordneten Transaktionen, Vorlagen und Backup-Snapshots hochzuladen.

Dieser Befehl überschreibt nicht die vorhandenen Dateien in der Umgebung. EPM Automate zeigt einen Fehler an, wenn der Name der Datei, die hochgeladen wird, mit dem Namen einer Datei im Speicherort zum Hochladen übereinstimmt.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator, Poweruser, die der Anwendungsrolle Migrationsadministrator zugewiesen sind

Verwendung

```
epmautomate uploadFile "FILE_NAME" [UPLOAD_LOCATION], wobei Folgendes gilt:
```

- *FILE_NAME* ist der Name der Datei, einschließlich des absoluten Pfades, wenn sich die Datei nicht in dem Verzeichnis befindet, aus dem Sie EPM Automate ausführen.
- *UPLOAD_LOCATION* (optional) entspricht dem Oracle Enterprise Performance Management Cloud-Speicherort, an den Sie die Datei hochladen möchten. Geben Sie keinen Speicherort zum Hochladen an, wenn Sie die Dateien an den Standardspeicherort zum Hochladen hochladen möchten. Ausführliche Informationen finden Sie unter [Standarddateispeicherorte](#). Unterstützte Werte:
 - `inbox` zum Hochladen von Dateien in die Inbox. Mit Ausnahme von Profitability and Cost Management suchen EPM Cloud-Geschäftsprozesse an diesem Speicherort nach zu verarbeitenden Dateien.

- `profitinbox` zum Hochladen von Dateien, die mit Profitability and Cost Management verarbeitet werden sollen.
- `to_be_imported` zum Hochladen eines Narrative Reporting-Snapshots, der während der nächsten täglichen Wartung der Umgebung importiert werden soll.
- `inbox/Verzeichnisname` zum Hochladen von Dateien in ein Verzeichnis in der Inbox für die Verarbeitung durch das Datenmanagement.
- `outbox` zum Hochladen von Dateien, die von anderen Geschäftsprozessen als Profitability and Cost Management-Prozessen verwendet werden, in die Outbox.
- `profitoutbox` zum Hochladen von Dateien, die von Profitability and Cost Management verwendet werden, in die Outbox.

Beispiele

- Snapshot in den Standardspeicherort hochladen:
`epmautomate uploadFile "C:/snapshots/backup_snapshot.zip"`
- Dateien in die Datenmanagement-Inbox hochladen:
`epmautomate uploadFile "C:/pbcsdata/quarterlydata.csv" inbox`
- Dateien in einen Ordner in der Inbox (für Datenmanagement) hochladen:
`epmautomate uploadFile "C:/fdmee_data/data.zip" inbox/dm_folder`
- Dateien in die "profitinbox" (Profitability and Cost Management) hochladen:
`epmautomate uploadFile "C:/profitability_data/data.zip" profitinbox`
- Narrative Reporting-Snapshot aus dem Verzeichnis `C:\temp` in den Speicherort "to_be_imported" hochladen:
`epmautomate uploadFile "C:\temp\EPRCS_Backup.tar.gz" to_be_imported`

userAuditReport

Generiert einen Benutzerauditbericht (CSV-Datei) und speichert ihn am Standardspeicherort für Downloads.

Der Benutzerauditbericht enthält Informationen zu den Benutzern, die sich in einem bestimmten Zeitraum (in den letzten maximal 120 Tagen) bei einer Umgebung angemeldet haben. Er listet die Anmelde-ID des Benutzers, die IP-Adresse des Computers, auf dem sich der Benutzer angemeldet hat, sowie das Datum und die Uhrzeit (z.B. 28. Juli 2022 18:43:21 UTC) des Benutzerzugriffs auf die Umgebung auf.

Hinweis:

Der Benutzerauditbericht listet für einen Benutzer, der sich innerhalb eines Zeitraums von fünf Minuten mehrmals bei der Oracle Enterprise Performance Management Cloud-Umgebung angemeldet hat, nur einen Anmeldeeintrag auf.

	A	B	C
1	Type Of Report	User Audit Report	
2	Report Generated Date	6/1/2020	
3			
4	User Name	IP Address	Access Date and Time
5			
6	jdoe@example.com	111.22.23.6	June 1, 2020 22:28:00 UTC
7	jane.doe@example.com	111.22.23.6	June 1, 2020 21:40:56 UTC

Verwenden Sie den Befehl [downloadFile](#), um den generierten Bericht auf Ihren Computer herunterzuladen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate userAuditReport FROM_DATE TO_DATE REPORT_NAME` wobei:

- `FROM_DATE` für das Startdatum der Periode (im Format YYYY-MM-DD) steht, für die der Auditbericht generiert werden soll.
- `TO_DATE` für das Enddatum der Periode (im Format YYYY-MM-DD) steht, für die der Auditbericht generiert werden soll.
- `REPORT_NAME` der Name der Berichtsdatei ist.

Hinweis:

Dieser Bericht kann nur für die letzten 120 Tage generiert werden.

Beispiel

```
epmautomate userAuditReport 2016-10-15 2016-12-15 myAuditReport.CSV
```

userGroupReport

Generiert einen Bericht (CSV-Datei), in dem die Gruppen aufgelistet sind, denen Benutzer in der Zugriffskontrolle zugewiesen sind, und speichert ihn am Standardspeicherort für Downloads.

Der Bericht gibt an, ob die Benutzerzuweisung zu einer Gruppe direkt (als Mitglied einer Gruppe) oder indirekt (als Mitglied einer Gruppe, die einer verschachtelten Gruppe untergeordnet ist) erfolgt.

In dem Bericht werden der Anmeldename, der Vorname, der Nachname, die E-Mail-Adresse, die zugewiesene Gruppe und der Zuweisungstyp im folgenden Format

identifiziert. Er ist identisch mit der CSV-Version des Berichts, der in der Registerkarte "Benutzergruppenbericht" in Zugriffskontrolle erstellt wird. Beispiel: Angenommen, der Benutzer `jdoo` ist ein Mitglied der Gruppe `Test1`, die der verschachtelten Gruppe `Test2` untergeordnet ist. In diesem Szenario werden im Bericht die folgenden Informationen für `jdoo` angezeigt:

```
User Login,First Name,Last Name,Email,Direct,Group
jdoo, John, Doe, jdoo@example.com, Yes, test1
jdoo, John, Doe, jdoo@example.com, No, test2
```

Verwenden Sie den Befehl [downloadFile](#), um den generierten Bericht auf Ihren Computer herunterzuladen.

Gilt für

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Enterprise Data Management Cloud, Narrative Reporting, Sales Planning und Strategische Personalplanung.

Erforderliche Rollen

Serviceadministrator oder Zugriffskontrollmanager

Verwendung

`epmautomate userGroupReport REPORT_NAME`, wobei `REPORT_NAME` der Name der Berichtsdatei ist.

Beispiel

```
epmautomate userGroupReport UsrGrpReport.CSV
```

validateConsolidationMetadata

Validiert die Metadaten der Umgebung, um eine fehlerfreie Datenbankaktualisierung und -konsolidierung sicherzustellen.

Führen Sie diesen Befehl nach dem Importieren von Metadaten mit dem Befehl [importMetadata](#) aus, um die Metadaten zu validieren und eine fehlerfreie Datenbankaktualisierung sicherzustellen, wenn Sie den Befehl [refreshCube](#) ausführen. Wenn Ihre Konsolidierungsmetadaten nicht korrekt sind, kann Ihre Konsolidierung unter Umständen auch fehlschlagen.

Durch diesen Befehl wird 0 (Null) oder die Anzahl der Validierungsfehler in der Konsole angezeigt, über die er ausgeführt wird. Er schreibt Validierungsfehler in eine CSV-Datei, die Sie verwenden können, um die Metadatenfehler zu beheben. Verwenden Sie den Befehl [downloadFile](#), um die resultierende CSV-Datei auf einen lokalen Server herunterzuladen.

Gilt für

Financial Consolidation and Close

Erforderliche Rollen

Serviceadministrator

Verwendung

`epmautomate validateConsolidationMetadata LOG_FILE_NAME`, wobei `LOG_FILE_NAME` der Name der Datei ist, die die Informationen zu den durch diesen Befehl identifizierten Fehlern enthält.

Beispiele

```
epmautomate validateConsolidationMetadata validation_error.csv
```

validateModel

Validiert ein Enterprise Profitability and Cost Management-Modell und schreibt die Validierungsausgabe in eine Datei.

Gilt für

Enterprise Profitability and Cost Management

Erforderliche Rollen

Serviceadministrator

Verwendung

```
epmautomate validateModel "modelName" FILE_NAME.txt [messageType=All|Warning|Error], wobei Folgendes gilt:
```

- `modelName` ist der Name des zu validierenden Enterprise Profitability and Cost Management-Modells. Dieser Wert muss in doppelten Anführungszeichen angegeben werden.
- `FILE_NAME` ist ein eindeutiger Name für eine Textdatei, in die EPM Automate die Modellvalidierungsausgabe schreiben soll. Diese Datei wird in der Outbox erstellt und kann mit dem Befehl [downloadFile](#) heruntergeladen werden.
- `messageType`, optional, ist der Status der Informationen, die in die Modellvalidierungsausgabe aufgenommen werden sollen. Mögliche Parameterwerte sind:
 - `Alle`. Mit diesem Wert werden Fehler und Warnungen in die Ausgabedatei der Validierung geschrieben.
 - `Fehler`. Mit diesem Wert werden nur Fehler in die Ausgabedatei der Validierung geschrieben. Das ist der Standardwert.
 - `Warnung`. Mit diesem Wert werden nur Modellvalidierungswarnungen in die Ausgabedatei der Validierung geschrieben.

Beispiel

```
epmautomate validateModel "10 Actuals Allocation Process" results.txt  
messageType=All
```

Exit-Codes

EPM Automate gibt einen Exit-Code und eine Meldung zur Angabe des Vorgangstatus zurück. Exit-Codes werden unter fünf Codenummern zusammengefasst, wobei jeder Code eine Vielzahl an Fehlerbedingungen angeben kann. Prüfen Sie die zugehörige Meldung, um die bestimmte Bedingung zu identifizieren, die den Fehler verursachte.

Außerdem erstellt EPM Automate für jede nicht erfolgreiche Befehlsausführung eine Logdatei (*COMMANDNAME_TIMESTAMP.log*, z.B. *uploadfile_16_11_2016_11_27_10.log*). Logdateien werden auf dem Computer erstellt, auf dem EPM Automate ausgeführt wird.

Fehler zu Exit-Code 1

Command failed to execute (Befehl konnte nicht ausgeführt werden.) EPM Automate verwendet diesen Exit-Code, um Meldungen zu den HTTP-Statuscodes 200 und 400 anzuzeigen. Diese Codes werden von den REST-APIs zurückgegeben, die von EPM Automate verwendet werden.

Insufficient privileges to perform the operation (Unzureichende Berechtigungen zur Ausführung des Vorgangs) Dieser Fehler wird angezeigt, wenn der Benutzer, dessen Zugangsdaten zum Anmelden beim Service verwendet werden, über keine ausreichenden Berechtigungen verfügt, um den beabsichtigten Vorgang auszuführen.

Melden Sie sich mit einem Konto an, das über ausreichende Berechtigungen zum Ausführen des Vorgangs verfügt. Üblicherweise sind nur Serviceadministratoren zur Ausführung von Vorgängen im Service berechtigt.

Resource does not exist (Ressource ist nicht vorhanden.) Dieser Fehler wird angezeigt, wenn die Datei oder der Snapshot, die bzw. der gelöscht oder heruntergeladen werden soll, im Service nicht vorhanden ist.

Verwenden Sie den Befehl `listfiles`, um den Namen der Datei und ihren Speicherort zu prüfen.

Invalid snapshot *SNAPSHOT* (Ungültiger Snapshot *SNAPSHOT*) Dieser Fehler wird angezeigt, wenn der Service den Snapshot nicht validieren kann, der für den Export- oder Importvorgang angegeben wurde.

Prüfen Sie, ob Sie einen gültigen Snapshot verwenden.

Internal server error. Unable to delete file: *FILE_NAME* Please issue "Provide Feedback" with details (Interner Serverfehler. Datei kann nicht gelöscht werden: *DATEINAME*. Setzen Sie "Feedback geben" mit Details ab.) Dieser Fehler wird angezeigt, wenn die Datei oder der Snapshot aufgrund eines Serverfehlers nicht im Service gelöscht werden kann.

Melden Sie Oracle dieses Problem über den Feedbackbefehl oder die Funktion "Feedback geben".

Invalid file: *FILE_NAME* (Ungültige Datei: *DATEINAME*) Dieser Fehler wird angezeigt, wenn die Datei oder der Snapshot, die bzw. der gelöscht oder heruntergeladen werden soll, im Service nicht vorhanden ist oder wenn der Dateiname nicht das erforderliche Format aufweist.

Verwenden Sie den Befehl `listfiles`, um den Namen der Datei und ihren Speicherort zu prüfen.

Recreate is running for a long time. Please contact support (Neuerstellung wird lange Zeit ausgeführt. Wenden Sie sich an den Support.) Dieser Fehler wird angezeigt, wenn der von Ihnen eingeleitete Vorgang zur Neuerstellung nicht innerhalb einer Stunde abgeschlossen ist.

Melden Sie Oracle dieses Problem über den Feedbackbefehl oder die Funktion "Feedback geben".

Reset service is running for a long time. Please contact support (Zurücksetzen des Service wird lange Zeit ausgeführt. Wenden Sie sich an den Support.) Dieser Fehler wird angezeigt, wenn der von Ihnen eingeleitete Vorgang zum Zurücksetzen des Service nicht innerhalb einer Stunde abgeschlossen ist.

Melden Sie Oracle dieses Problem über den Feedbackbefehl oder die Funktion "Feedback geben".

Cannot perform operation. Another instance is in progress. Please try after some time (Vorgang kann nicht durchgeführt werden. Eine andere Instanz wird ausgeführt. Wiederholen Sie den Versuch später.) Dieser Fehler wird angezeigt, wenn Sie versuchen, den Befehl `copysnapshotfrominstance` auszuführen, während eine andere Instanz des Befehls aktiv ist.

Warten Sie, bis der Befehl `copysnapshotfrominstance` beendet ist, bevor Sie versuchen, den Befehl erneut auszuführen.

Cannot perform operation. Another maintenance script is in progress. Please try after some time (Vorgang kann nicht durchgeführt werden. Ein anderes Wartungsskript wird ausgeführt. Wiederholen Sie den Versuch später.) Dieser Fehler wird angezeigt, wenn Sie versuchen, den Befehl `copysnapshotfrominstance`, `recreate` oder `resetservice` auszuführen, während der tägliche Prozess zur Wartung oder zum Zurücksetzen des Service ausgeführt wird.

Führen Sie den Vorgang erneut durch, wenn der Wartungs- oder Zurücksetzungsprozess beendet ist.

Login to source instance failed: SOURCE_URL (Anmelden bei Quellinstanz nicht erfolgreich: QUELL-URL) Dieser Fehler wird angezeigt, wenn sich EPM Automate nicht bei der Quellumgebung anmelden kann, um den Befehl `copysnapshotfrominstance` zu initiieren.

Prüfen Sie, ob die Zugangsdaten, die Identitätsdomain und die URL gültig sind, die für den Zugriff auf die Quellumgebung verwendet werden.

Internal server error. Copy snapshot from source instance failed. Please issue "Provide Feedback" with details (Interner Serverfehler. Kopieren von Snapshot aus Quellinstanz nicht erfolgreich. Setzen Sie "Feedback geben" mit Details ab.) Dieser Fehler wird angezeigt, wenn während der Ausführung des Prozesses `copysnapshotfrominstance` ein unerwartetes Problem bei EPM Automate auftritt.

Melden Sie Oracle dieses Problem über den Befehl `feedback` oder die Funktion "Feedback geben".

Internal server error. Please issue "Provide Feedback" with details (Interner Serverfehler. Setzen Sie "Feedback geben" mit Details ab.) Dieser Fehler wird angezeigt, um anzugeben, dass eine Vielzahl an internen Serverbedingungen Korrekturmaßnahmen von Oracle erfordern.

Melden Sie Oracle dieses Problem über den Befehl `feedback` oder die Funktion "Feedback geben".

Snapshot `SNAPSHOT_NAME` already exists. Please delete the snapshot and rerun the command (Snapshot `SNAPSHOTNAME` bereits vorhanden. Löschen Sie den Snapshot, und führen Sie den Befehl erneut aus.) Dieser Fehler wird angezeigt, wenn ein Snapshot von einem Speicherort heruntergeladen oder an einen Speicherort hochgeladen wird, an dem ein anderer Snapshot mit identischem Namen vorhanden ist.

Löschen oder verschieben Sie den vorhandenen Snapshot, und wiederholen Sie den Befehl.

Error in extracting the snapshot. Please retry with a proper snapshot (Fehler beim Extrahieren des Snapshots. Wiederholen Sie den Vorgang mit einem ordnungsgemäßen Snapshot.) Dieser Fehler wird angezeigt, wenn EPM Automate Snapshot-Inhalte beim Ausführen des Befehls `importsnapshot` nicht extrahieren kann.

Prüfen Sie, ob der Snapshot gültig ist.

Internal server error. Unable to open file for write. Please issue "Provide Feedback" with details (Interner Serverfehler. Datei kann zum Schreiben nicht geöffnet werden. Setzen Sie "Feedback geben" mit Details ab.) Dieser Fehler wird angezeigt, wenn Fehler das Erstellen oder Aktualisieren von CSV-Dateien verursachen, z.B. beim Generieren des Auditberichts.

Melden Sie Oracle dieses Problem über den Befehl `feedback` oder die Funktion "Feedback geben".

No matching records found, please select a different date range (Keine passenden Datensätze gefunden. Wählen Sie einen anderen Datumsbereich aus.) Dieser Fehler wird angezeigt, wenn Sie den Befehl `userauditreport` ausführen, um den Auditbericht für einen Datumsbereich zu generieren, für den keine Auditdaten verfügbar sind.

Geben Sie einen gültigen Datumsbereich an, und führen Sie dann den Befehl `userauditreport` erneut aus. Beachten Sie, dass der Service nur die Audithistorie der letzten 365 Tage beibehält.

File with same name exists: `FILE_NAME`, please choose a different filename (Es ist bereits eine Datei mit demselben Namen vorhanden: `DATEINAME`. Wählen Sie einen anderen Dateinamen aus.) Dieser Fehler wird angezeigt, wenn ein Bericht mit dem von Ihnen angegebenen Auditberichtsnamen bereits im Service vorhanden ist.

Löschen Sie im Service die vorhandene Datei, oder geben Sie einen anderen Dateinamen an, und führen Sie dann den Befehl `userauditreport` erneut aus.

Operation failed with status \$1. Please issue "Provide Feedback" (Vorgang nicht erfolgreich. Status: `$1`. Setzen Sie "Feedback geben" ab.) Diese Meldung gibt einen internen Serverfehler an, der bewirkt, dass der Prozess zum Zurücksetzen oder zum Neuerstellen des Service nicht erfolgreich ist.

Melden Sie Oracle dieses Problem über den Befehl `feedback` oder die Funktion "Feedback geben".

EPMCSS-20643: Failed to add users. File `FILE_NAME.csv` is not found. Please provide a valid file name (EPMCSS-20643: Benutzer konnten nicht hinzugefügt werden. Datei `DATEINAME.csv` wurde nicht gefunden. Geben Sie einen gültigen Dateinamen an.) Dieser Fehler wird angezeigt, wenn die angegebene CSV-Datei mit Informationen zu Benutzern, die hinzugefügt werden sollen, in der Inbox nicht verfügbar ist.

Verwenden Sie den Befehl `listfiles`, um den Namen der Datei und ihren Speicherort zu prüfen. Wenn sich die Datei nicht in der Inbox befindet, verwenden Sie den Befehl `uploadFile`, um die Datei hochzuladen.

EPMCSS-20644: Failed to remove users. File *FILE_NAME.csv* is not found. Please provide a valid file name (EPMCSS-20644: Benutzer konnten nicht entfernt werden. Datei DATEINAME.csv wurde nicht gefunden. Geben Sie einen gültigen Dateinamen an.) Dieser Fehler wird angezeigt, wenn die angegebene CSV-Datei mit Informationen zu Benutzern, die gelöscht werden sollen, in der Inbox nicht verfügbar ist.

Verwenden Sie den Befehl `listfiles`, um den Namen der Datei und ihren Speicherort zu prüfen. Wenn sich die Datei nicht in der Inbox befindet, verwenden Sie den Befehl `uploadFile`, um die Datei hochzuladen.

20645: Failed to assign role for users. Invalid role name *role*. Please provide a valid role name (20645: Rolle für Benutzer konnte nicht zugeordnet werden. Ungültiger Rollename "Rolle". Geben Sie einen gültigen Rollennamen an.) Dieser Fehler wird angezeigt, wenn die in der CSV-Datei angegebene Rolle nicht unterstützt wird.

Stellen Sie sicher, dass der in der Datei angegebene Rollename `Serviceadministrator`, `Poweruser`, `Benutzer` oder `Leseberechtigter` ist.

Verwenden Sie den Befehl `listfiles`, um den Namen der Datei und ihren Speicherort zu prüfen. Wenn sich die Datei nicht in der Inbox befindet, verwenden Sie den Befehl `uploadFile`, um die Datei hochzuladen.

Fehler zu Exit-Code 6

Service Unavailable (Service nicht verfügbar) Service ist aufgrund von HTTP-Fehler 404 nicht verfügbar.

Prüfen Sie die Serviceverfügbarkeit, indem Sie den Service über einen Browser auf dem Computer aufrufen, auf dem Sie EPM Automate ausführen. Wenn der Service aus einem beliebigen Grund nicht aktiv ist, warten Sie, und versuchen Sie es erneut, oder wenden Sie sich an den Oracle-Support.

Read/Write timeout (Timeout bei Lesen/Schreiben) Dieser Fehler wird angezeigt, wenn während eines Lese-/Schreibvorgangs aufgrund eines langsamen Netzwerkes oder aufgrund von Firewallproblemen ein Timeout für den Client-Socket auftritt (Socket-Timeout liegt bei 15 Minuten).

Führen Sie den nicht erfolgreichen Befehl erneut aus, wenn der Netzwerkdurchsatz hoch ist. Wenn der Fehler aufgrund von Firewall-Einstellungen auftritt, wenden Sie sich an Ihren Netzwerkadministrator.

Fehler zu Exit-Code 7

EPM Automate zeigt diesen Fehler an, wenn ein Befehl nicht ausgeführt werden kann. Die Fehlermeldung, z.B. `Invalid command` (Ungültiger Befehl), gibt an, warum der Fehler aufgetreten ist.

Unable to open password file *FILE_NAME* (Kennwortdatei DATEINAME kann nicht geöffnet werden.) Ungültig verschlüsselte Kennwortdatei, z.B. `PASSWORD_FILE.EPW`. EPM Automate hat die Datei nicht an dem von Ihnen angegebenen Speicherort gefunden, oder die Datei weist nicht das erforderliche Format auf.

Prüfen Sie den Dateinamen und -pfad. Wenn die Datei aufgrund eines ungültigen Formats nicht geparkt werden kann, verwenden Sie den Befehl [encrypt](#), um die Datei erneut zu erstellen.

Unable to parse password file *FILE_NAME* (Kennwortdatei *DATEINAME* kann nicht geparkt werden.) EPM Automate konnte die verschlüsselte Kennwortdatei aufgrund eines ungültigen Formats oder aufgrund von Beschädigung nicht parsen.

Verwenden Sie den Befehl [encrypt](#), um die Datei erneut zu erstellen.

Unable to connect to *URL*. Root cause *MESSAGE* (Verbindung mit *URL* nicht möglich. Ursache: *MELDUNG*.) Dieser Fehler wird angezeigt, wenn eine Verbindung aufgrund einer falschen *URL* nicht hergestellt werden kann. Die Meldung, die als Ursache angezeigt wird, erläutert den zugrunde liegenden Fehler, der sich aus der Verwendung einer nicht ordnungsgemäßen *URL* ergibt.

- Prüfen Sie, ob Sie eine gültige *URL* verwenden.
- Wenn gemäß Ihrer Proxyeinstellungen die Authentifizierung beim Proxyserver erforderlich ist, um eine Verbindung zum Internet herzustellen, geben Sie zum Anmelden einen Benutzernamen, eine Domain und ein Kennwort für den Proxyserver ein (oder verwenden Sie eine verschlüsselte Kennwortdatei, die das Kennwort für den Proxyserver enthält), um sich beim Proxyserver zu authentifizieren. Wenden Sie sich an Ihren Netzwerkadministrator, wenn Sie Unterstützung benötigen.

Unable to connect to *URL* Unsupported Protocol (Mit *URL* kann keine Verbindung hergestellt werden. Nicht unterstütztes Protokoll.) Der Anmeldebefehl war nicht erfolgreich, da die angegebene *URL* ein nicht unterstütztes Protokoll verwendet. Die zugehörige Fehlermeldung gibt das nicht unterstützte Protokoll an.

Stellen Sie sicher, dass die von Ihnen mit dem Anmeldebefehl verwendete *URL* das sichere Protokoll HTTPS verwendet.

Session is not authenticated. Please execute the "login" command before executing any other command (Session wurde nicht authentifiziert. Führen Sie den Befehl "login" aus, bevor andere Befehle ausgeführt werden.) Sie haben versucht, einen Befehl auszuführen, bevor eine Session mit dem Befehl `login` erstellt wurde.

Führen Sie den Befehl [login](#) aus, um eine sichere Verbindung zu einer Umgebung herzustellen, bevor Sie versuchen, andere Befehle auszuführen.

Invalid parameter (Ungültiger Parameter) Diese Meldung gibt einen Verwendungsfehler in einem Befehl an, der durch eine falsche Abfolge von Befehlsparametern oder durch Fehlen einiger erforderlicher Befehlsparameterwerte verursacht wurde.

Prüfen und korrigieren Sie die Befehlsparameter und die Abfolge, in der sie angegeben sind.

COMMAND_NAME* command is not supported by *SERVICE_TYPE (Befehl *BEFEHLSNAME* wird von *SERVICETYP* nicht unterstützt.) EPM Automate konnte den Befehl nicht für den Service ausführen, mit dem Sie verbunden sind, da der Service den Befehl nicht unterstützt.

Unter [Befehle](#) finden Sie Listen der Befehle, die von jedem Geschäftsprozess unterstützt werden.

File does not exist in location: *PATH* (Datei ist nicht vorhanden an Speicherort: *PFAD*.) EPM Automate konnte die Datei, die verarbeitet werden soll, z.B. mit dem Befehl `upload` oder `replay`, nicht finden.

Stellen Sie sicher, dass der Dateiname und -pfad richtig sind.

Unable to open file for read: *PATH* (Datei kann zum Lesen nicht geöffnet werden: PFAD) EPM Automate konnte aus der angegebenen Datei nicht lesen.

Stellen Sie sicher, dass die Datei das erforderliche Format aufweist. Prüfen Sie, ob der Benutzer, der EPM Automate ausführt, über Lesezugriff auf die Datei verfügt.

Unable to open file for write: *PATH* (Datei kann zum Schreiben nicht geöffnet werden: PFAD) EPM Automate konnte nicht in die angegebene Datei schreiben.

Stellen Sie sicher, dass die Datei nicht durch einen anderen Prozess gesperrt ist. Prüfen Sie, ob der Benutzer, der EPM Automate ausführt, über Schreibzugriff auf die Datei verfügt.

Invalid command (Ungültiger Befehl) EPM Automate hat einen nicht unterstützten Befehl festgestellt.

Prüfen Sie, ob der Befehl von EPM Automate unterstützt wird. Stellen Sie außerdem sicher, dass der Befehlsname richtig geschrieben ist.

Invalid date format (Ungültiges Datumsformat) Das Tool hat ein ungültiges Datumsformat festgestellt.

Machen Sie die Datumsangaben für die Berichterstellung in einem unterstützten Datumsformat.

FROMDATE *DATE* cannot be greater than TODATE *DATE* (ANFANGSDATUM DATUM darf nicht nach ENDDATUM DATUM liegen.) EPM Automate hat ein Enddatum festgestellt, das vor dem Anfangsdatum liegt.

Stellen Sie sicher, dass der Wert für *to date* in einem bestimmten Datumsbereich ein späteres Datum ist als der Wert für *from date*.

Exceeded maximum number of feedbacks (6) for a day (Maximale Anzahl Feedbacks (6) für einen Tag überschritten.) Dieser Fehler wird angezeigt, wenn Sie die Anzahl Feedbacks überschreiten, die Sie mit dem Befehl *feedback* übergeben können.

File with the same name already exists in the download path *PATH*. Please delete the file and rerun the command (Datei mit demselben Namen ist bereits im Downloadpfad PFAD vorhanden. Löschen Sie die Datei, und führen Sie den Befehl erneut aus.) Dieser Fehler wird angezeigt, wenn Sie versuchen, eine Datei an einen Speicherort herunterzuladen, an dem bereits eine Datei vorhanden ist, deren Name mit dem Namen der Datei übereinstimmt, die heruntergeladen wird.

Löschen oder verschieben Sie die vorhandene Datei, oder benennen Sie sie um, und führen Sie den Befehl dann erneut aus.

File is empty: *PATH* (Datei ist leer: PFAD) Dieser Fehler wird angezeigt, wenn die Wiedergabedatei keinen Inhalt aufweist.

Stellen Sie sicher, dass die Wiedergabedatei (CSV-Datei) die Zugangsdaten (Benutzername und Kennwort) sowie den Namen der HAR-Dateien aufführt, die zum Ausführen des Befehls *replay* verwendet werden.

Unable to encrypt the password as localhost cannot be resolved. Ensure that hostnames are mapped properly to IP addresses (Kennwort kann nicht verschlüsselt werden, da localhost nicht aufgelöst werden kann. Stellen Sie sicher, dass Hostnamen ordnungsgemäß IP-Adressen zugeordnet sind.)

Dieser Fehler wird angezeigt, wenn EPM Automate die localhost-Definition zu einer MAC-Adresse nicht auflösen kann, da die Hostdatei auf Ihrem Computer für die Adresse 127.0.0.1 einen Servernamen anstelle von localhost enthält.

Stellen Sie sicher, dass die Hostdatei localhost als Servernamen für 127.0.0.1 angibt.

Snapshot Name is invalid (Snapshot-Name ist ungültig.) Dieser Fehler wird angezeigt, wenn Sie den Namen des umzubenennenden Snapshots nicht angeben.

Geben Sie den Namen eines Snapshots an, der in der Umgebung verfügbar ist.

New Snapshot Name is invalid (Neuer Snapshot-Name ist ungültig.) Dieser Fehler wird angezeigt, wenn Sie keinen neuen Namen für den Snapshot angeben.

Geben Sie den neuen Namen für den Snapshot an.

Invalid snapshot name: {0}. Characters \\/*?"<>| are not allowed (Ungültiger Snapshot-Name: {0}. Die Zeichen \\/*?"<>| sind nicht zulässig.) Dieser Fehler wird angezeigt, wenn der Snapshot-Name Sonderzeichen wie Leerzeichen, \ (umgekehrter Schrägstrich), / (Schrägstrich), * (Sternchen), ? (Fragezeichen), " (Anführungszeichen), < (Kleiner-als-Zeichen) und > (Größer-als-Zeichen) enthält.

Geben Sie einen neuen Snapshot-Namen an, der keines dieser Sonderzeichen enthält.

Unable to rename snapshot : {0}. There could be another process accessing it. Please try after sometime (Snapshot kann nicht umbenannt werden: {0}. Möglicherweise greift ein anderer Prozess darauf zu. Versuchen Sie es später erneut.) Dieser Fehler wird angezeigt, wenn EPM Automate keine exklusive Sperre für den Snapshot erhalten kann, da dieser von einem anderen Prozess verwendet wird.

Warten Sie, bis der aktuelle Vorgang, der den Snapshot verwendet, abgeschlossen ist, und versuchen Sie es dann erneut.

Snapshot {0} already exists. Please delete the snapshot and re-run the command (Snapshot {0} ist bereits vorhanden. Löschen Sie den Snapshot, und führen Sie den Befehl erneut aus.) Dieser Fehler wird angezeigt, wenn der neue Snapshot-Name mit dem Namen eines vorhandenen Snapshots in der Umgebung identisch ist.

Verwenden Sie einen anderen Snapshot-Namen, oder löschen Sie den vorhandenen Snapshot mit dem Befehl `deletefile`.

Fehler zu Exit-Code 9

Invalid credentials (Ungültige Zugangsdaten) Dieser Fehler wird angezeigt, wenn der mit dem Befehl `login` verwendete Benutzername oder das Kennwort falsch ist.

Geben Sie gültige Zugangsdaten für die Umgebung an, mit der Sie eine Verbindung herstellen möchten.

Authentication failed while executing command. Please retry (Authentifizierung während der Ausführung des Befehls nicht erfolgreich. Versuchen Sie es erneut.) Dieser Fehler wird angezeigt, wenn die Basisauthentifizierung während der Ausführung eines anderen Befehls als `login` nicht erfolgreich ist. Dieser Fehler kann auch bei HTTP-Aufrufen auftreten, wenn eine Befehlsausführung wiederholt versucht wird (bis zu drei Mal).

Fehler zu Exit-Code 11

Internal server error. Due to manual reset service, your Oracle EPM Cloud Service environment is currently unavailable. (Interner Serverfehler. Aufgrund einer manuellen Zurücksetzung ist Ihre Oracle EPM Cloud Service-Umgebung derzeit nicht verfügbar.) Dieser Fehler wird angezeigt, wenn EPM Automate-Befehle aufgeführt werden, während die Umgebung zurückgesetzt wird.

Internal server error MESSAGE (Interner Serverfehler: MELDUNG) Dieser Fehler wird angezeigt, wenn EPM Automate unbekannte Ausnahmen feststellt, die nicht mit HTTP-Verbindungen zusammenhängen. Enthält Serverfehler 503 und 500.

Unable to connect to URL: MESSAGE (Verbindung mit URL nicht möglich: MELDUNG) Dieser Fehler wird angezeigt, wenn der Server nicht verfügbar ist. Die Fehlermeldung gibt die Ausnahme an, die bewirkt, dass der Befehl nicht erfolgreich ist.

Wenn der Server nicht verfügbar ist, wenden Sie sich an den Oracle-Support. Wenn die Meldung Probleme mit der URL angibt, prüfen Sie, ob die von Ihnen verwendete URL gültig ist.

3

Beispielszenarios für die Befehlsausführung

EPM Automate kann zur Automatisierung vieler gängiger Oracle Enterprise Performance Management Cloud-Verwaltungsaufgaben verwendet werden.

- [Beispielszenarios für alle Services](#)
- [Beispielszenarios für Planning, Consolidation, Tax Reporting und Enterprise Profitability and Cost Management](#)
- [Beispielszenarios für Account Reconciliation](#)
- [Beispielszenarios für Profitability and Cost Management](#)
- [Beispielszenarios für Oracle Enterprise Data Management Cloud](#)

Beispielskripte kopieren

Kopieren Sie keine Beispielskripte aus der PDF-Version dieses Dokuments. Um zu verhindern, dass Skripte wegen Zeilenumbrüchen und Fußzeileninformationen unbrauchbar werden, empfiehlt Oracle, die Beispielskripte aus der HTML-Version von [Mit EPM Automate für Oracle Enterprise Performance Management Cloud arbeiten](#) zu kopieren.

Beispielszenarios für alle Services

In diesen Szenarios ist eine typische Abfolge von Befehlen dargestellt, mit denen bestimmte Vorgänge in Oracle Enterprise Performance Management Cloud-Umgebungen durchgeführt werden können.

Siehe auch:

- [Anwendungs-Snapshots auf einem Computer sichern](#)
In diesem Szenario wird erläutert, wie der Sicherungsprozess des Snapshots, der während der täglichen Servicewartung auf einem lokalen Computer erstellt wird, automatisiert wird.
- [Benutzer über den Abschluss der täglichen Wartung informieren](#)
Die tägliche Wartung von Oracle Enterprise Performance Management Cloud-Umgebungen dauert in der Regel deutlich weniger als die dafür vorgesehene eine Stunde.
- [Snapshot in oder aus Oracle Object Storage kopieren](#)
- [Benutzer erstellen und vordefinierten Rollen zuweisen](#)
Verwenden Sie die Skripte in diesem Abschnitt, um Benutzer in der Identitätsdomain zu erstellen und sie vordefinierten Rollen zuzuweisen.
- [Anzahl der lizenzierten Benutzer zählen \(Benutzer, die Rollen zugewiesen sind\)](#)
Verwenden Sie das Skript in diesem Abschnitt, um den Bericht zur Rollenzuweisung zum Zählen der Anzahl an Benutzern für eine Umgebung zu generieren.

- [Auditberichte zu Benutzern erstellen, die Rollen zugewiesen sind](#)
Verwenden Sie die Skripte in diesem Abschnitt, um den Prozess zum Erstellen eines Auditberichts für Benutzer, die vordefinierten Rollen in einer Umgebung zugewiesen sind, zu automatisieren und den Bericht optional per E-Mail an einen Empfänger zu senden.
- [Auditberichte zu Rollenzuweisungen und zur Aufhebung von Zuweisungen](#)
Verwenden Sie das PowerShell-Skript in diesem Abschnitt, um den Prozess zum Erstellen eines Auditberichts zu automatisieren, in dem die Rollenzuweisung und die Rollenwiderrufung in einer Umgebung dargelegt sind.
- [Zugriffslogs und Aktivitätsberichte zur Einhaltung von Datenschutzgesetzen maskieren](#)
Verwenden Sie die Skripte in diesem Abschnitt, um den Prozess zum Maskieren von Informationen im Aktivitätsbericht oder in Zugriffslogs den Datenschutzgesetzen entsprechend zu automatisieren und den Bericht optional per E-Mail an einen Empfänger zu senden.
- [Downloads von Aktivitätsberichten auf einen lokalen Computer automatisieren](#)
Verwenden Sie das Skript in diesem Abschnitt, um das Herunterladen von Aktivitätsberichten aus einer Umgebung auf einen lokalen Computer zu automatisieren.
- [Zugriffslogs aus einer Umgebung herunterladen](#)
Verwenden Sie das Skript in diesem Abschnitt, um den Prozess zum Herunterladen von Zugriffslogs aus einer Umgebung auf einen lokalen Computer zu automatisieren.
- [Klonen von Umgebungen automatisieren](#)
Verwenden Sie das Skript in diesem Abschnitt, um das Klonen von Umgebungen zu automatisieren.
- [Unnötige Dateien aus einer Umgebung entfernen](#)
Verwenden Sie diese Skripte, um unnötige Dateien aus einer Umgebung zu entfernen.
- [Dateien aus einer Umgebung suchen und herunterladen](#)
Verwenden Sie das Beispielskript in diesem Abschnitt, um den Prozess zum Herunterladen von Dateien aus einer Oracle Enterprise Performance Management Cloud-Umgebung zu automatisieren. Verwenden Sie dabei eine Textzeichenfolge als Platzhalter.
- [Alte EPM Cloud-Umgebungen für Audits neu erstellen](#)
Verwenden Sie das Skript in diesem Abschnitt, um eine Selfservicelösung zu erstellen, mit der eine aktuelle Bibliothek von Snapshots für Ihre Oracle Enterprise Performance Management Cloud-Umgebung verwaltet werden kann. Sie benötigen eine Umgebung, die speziell für die Aktualisierung und Wartung einer Bibliothek mit aktuellen Snapshots eingerichtet ist.
- [Audit und Compliance des Datenbankzugriffs automatisieren](#)
Verwenden Sie die PowerShell- und Bash Shell-Skripte in diesem Abschnitt, um EPM Automate-Befehle zum Erfassen von Audit- und Compliance-Daten im Zusammenhang mit dem manuellen Datenbankzugriff zu nutzen.
- [Benutzer und vordefinierte Rollenzuweisungen replizieren](#)
Mit den Skripten in diesem Abschnitt können Sie Benutzer und vordefinierte Rollenzuweisungen aus einer Umgebung in eine andere migrieren.
- [Vierteljährliches Upgradeintervall für EPM Cloud erstellen](#)
Verwenden Sie diese Skripte, um eine Selfservice-Lösung zum Überspringen von Updates zu erstellen, sodass Oracle Enterprise Performance Management Cloud-

Umgebungen quartalsweise mit einem zweiwöchigen Testzyklus upgedatet werden. In diesem Fall werden Produktionsumgebungen zwei Wochen später als Testumgebungen upgedatet.

- **Vierteljährliches Upgradeintervall für EPM Cloud mit sechswöchigen Testzyklen erstellen**
Verwenden Sie das Skript in diesem Abschnitt, um eine Selfservice-Lösung zum Überspringen von Updates zu erstellen, sodass Oracle Enterprise Performance Management Cloud-Umgebungen quartalsweise mit einem sechswöchigen Testzyklus upgedatet werden. In diesem Fall werden Produktionsumgebungen sechs Wochen später als die Testumgebungen upgedatet.

Anwendungs-Snapshots auf einem Computer sichern

In diesem Szenario wird erläutert, wie der Sicherungsprozess des Snapshots, der während der täglichen Servicewartung auf einem lokalen Computer erstellt wird, automatisiert wird.

- Anwendungs-Snapshot (Artifact Snapshot), der im Wartungsfenster erstellt wurde, herunterladen
- Heruntergeladenen Snapshot durch Anhängen des Zeitstempels umbenennen
- Zehn Backups behalten und bei Bedarf das älteste löschen

Note:

- Dieses Skript kann nicht zum Erstellen eines Backups von Narrative Reporting verwendet werden.
- Wenn Sie dieses Skript an Ihre Zwecke anpassen möchten, ändern Sie nach Bedarf die Werte für die Laufzeitparameter (`url`, `user`, `password` und `NumberOfBackups`).

Informationen zum Planen von Skripten mit der Windows-Aufgabenplanung finden Sie unter [Skriptausführung automatisieren](#).

Windows-Beispielskript

Um Snapshot-Downloads zu automatisieren, erstellen Sie eine Batch- (.bat) oder Shell-Datei (.sh) mit einem Skript ähnlich dem folgenden.

```
@echo off
rem Sample script to download and maintain 10 maintenance backups
rem Update the following parameters

SET url=https://example.oraclecloud.com
SET user=ServiceAdmin
SET password=Example.epw
SET SnapshotName="Artifact Snapshot"
SET NumberOfBackups=10

rem EPM Automate commands
call epmautomate login %user% %password% %url%
IF %ERRORLEVEL% NEQ 0 goto :ERROR
call epmautomate downloadfile %SnapshotName%
```

```

        IF %ERRORLEVEL% NEQ 0 goto :ERROR
            call epmautomate logout
        IF %ERRORLEVEL% NEQ 0 goto :ERROR

rem Rename downloaded Artifact Snapshot, keep the last 10 backups
Set Timestamp=%date:~4,2%_%date:~7,2%_%date:~10,2%%
Set Second=%time:~0,2%%time:~3,2%
ren %SnapshotName%.zip %SnapshotName%_%Timestamp%_%Second%.zip

SET Count=0
FOR %%A IN (%SnapshotName%*.*) DO SET /A Count += 1
IF %Count% gtr %NumberOfBackups% FOR %%A IN (%SnapshotName%*.*) DO del
"%%A" && GOTO EOF
:EOF

echo Scheduled Task Completed successfully
exit /b %errorlevel%
:ERROR
echo Failed with error #%errorlevel%.
exit /b %errorlevel%

```

Linux-/UNIX-Beispielskript

Um Snapshot-Downloads zu automatisieren, erstellen Sie eine Shell-Datei (.sh) mit einem ähnlichen Skript wie das folgende. Wenn Ihr Kennwort Sonderzeichen enthält, finden Sie unter [Sonderzeichen verarbeiten](#) weitere Informationen.

```

#!/bin/sh
# Sample script to download and maintain 10 maintenance backups
# Update the following seven parameters

url=https://example.oraclecloud.com
user=serviceAdmin
password=/home/user1/epmautomate/bin/example.epw
snapshotname="Artifact Snapshot"
numberofbackups=10
epmautomatescript=/home/user1/epmautomate/bin/epmautomate.sh
javahome=/home/user1/jdk1.8.0_191/

export JAVA_HOME=${javahome}

printResult()
{
    op="$1"
    opoutput="$2"
    returncode="$3"

    if [ "${returncode}" -ne 0 ]
    then
        echo "Command failed. Error code: ${returncode}. ${opoutput}"
    else
        echo "${opoutput}"
    fi
}

```

```

processCommand()
{
    op="$1"
    date=`date`

    echo "Running ${epmautomatescript} ${op}"
    operationoutput=`eval "$epmautomatescript $op"`
    printResult "$op" "$operationoutput" "$?"
}

op="login ${user} ${password} ${url}"
processCommand "${op}"

op="downloadfile \"${snapshotname}\""
processCommand "${op}"

op="logout"
processCommand "${op}"

# Renames the downloaded artifacts, keeps the last 10 backups
timestamp=`date +%m_%d_%Y_%I%M`
mv "${snapshotname}.zip" "${snapshotname}_${timestamp}.zip"

((numberofbackups+=1))
ls -tp ${snapshotname}*.zip | grep -v '/$' | tail -n +${numberofbackups} |
xargs -d '\n' -r rm --

```

Benutzer über den Abschluss der täglichen Wartung informieren

Die tägliche Wartung von Oracle Enterprise Performance Management Cloud-Umgebungen dauert in der Regel deutlich weniger als die dafür vorgesehene eine Stunde.

Die tatsächliche Dauer der täglichen Wartung der Umgebung wird als Wert der Kennzahl für die Dauer der täglichen Wartung in Minuten im Abschnitt zu operativen Metriken des Aktivitätsberichts erfasst. Wenn Sie nicht eine Stunde warten möchten, bis Sie die Umgebung verwenden können, verwenden Sie eine benutzerdefinierte Version dieses Skripts, um Benutzer darüber zu informieren, dass die tägliche Wartung abgeschlossen ist, sodass sie ihre Aktivitäten fortsetzen können.

Windows-Skript

Erstellen Sie die Datei `daily_maintenance_completed.ps1`, indem Sie das folgende PowerShell-Skript kopieren. Informationen zum Aktualisieren des Skripts für die eigene Verwendung finden Sie unter [Skripte ausführen](#).

```

# Daily Maintenance Completed Notification script
#
# Update the following parameters
# -----
$emailaddresses=user1@oracle.com,user2@oracle.com
# -----

$username=$args[0]
$password=$args[1]

```



```

$url=$args[2]

if ($($args.count) -ne 3) {
    echo "Usage: ./daily_maintenance_completed.ps1 <USERNAME>
<PASSWORD> <URL>"
    exit 1
}

$amw_time=""

function getDailyMaintenanceStartTime {
    $amwstring=$(epmautomate.bat getDailyMaintenanceStartTime)
    $elements=$amwstring.split(' ')
    $amwtime=$elements[0]
    return $amwtime
}

function goToSleep ($amw_time){
    $current_mdy=Get-Date -AsUTC -UFormat "%m/%d/%Y"
    $current_date_time=Get-Date -AsUTC -UFormat "%m/%d/%Y %H:%M:%S"
    $current_epoch=Get-Date -Date $current_date_time -UFormat "%s"
    $target_date_time=[DateTime]"${current_mdy} ${amw_time}"
    $target_epoch=Get-Date -Date $target_date_time -UFormat "%s"
    $sleep_seconds=$target_epoch - $current_epoch

    # Today's AMW start time has already passed, so add 24 hours to
sleep_seconds
    if ($sleep_seconds -lt 0) {
        $sleep_seconds=$sleep_seconds + 86400
    }

    $sleep_ts=New-TimeSpan -Seconds ${sleep_seconds}
    $sleep_hms="${sleep_ts}" -replace '^\\d+?\\.\\.'

    echo "Current time is ${current_date_time}. Sleeping for $
{sleep_hms}, until daily maintenance start time of ${amw_time}."
    Start-Sleep -Seconds $sleep_seconds
}

function attemptLogin {
    $serverdown=$False
    while ($true) {
        epmautomate.bat login ${username} ${password} ${url}
        if ($?) { # login succeeded
            if ($serverdown) { # server has been brought down
                echo "Daily maintenance processing has completed ..."
                break
            } else { # server has not yet been brought down
                echo "Daily maintenance processing has not yet
started. Sleeping for 2 minutes before the next check ..."
                Start-Sleep -Seconds 120
            }
        } else { # login failed
            if ($serverdown) { # server has been brought down
                echo "Waiting for daily maintenance processing to

```

```

complete. Sleeping for 2 minutes before the next check ..."
        Start-Sleep -Seconds 120
    } else { # server has not yet been brought down
        echo "Daily maintenance processing is now beginning.
Sleeping for 2 minutes before the next check ..."
        Start-Sleep -Seconds 120
        $serverdown=$True
    }
}
}

function sendNotification {
    $servername=$url.split("/") [2];
    $subject="Daily maintenance processing has completed"
    $formattedmessage="Daily maintenance processing has completed for
server ${servername}"
    $emailaddresses=${emailaddresses}.replace(',',';')

    echo "Mailing report"
    epmautomate.bat sendmail "${emailaddresses}" "${subject}" Body="$
{formattedmessage}"
}

echo "Beginning daily maintenance completion notification script."
echo "Logging into server ..."
epmautomate.bat login ${username} ${password} ${url}
$amwtime=getDailyMaintenanceStartTime
goToSleep ($amwtime)
attemptLogin
sendNotification
echo "Logging out of server ..."
epmautomate.bat logout
echo "Script processing has completed."

```

Linux-/UNIX-Skript

Erstellen Sie die Datei `daily_maintenance_completed.sh`, indem Sie das folgende Skript kopieren. Informationen zum Aktualisieren des Skripts für die eigene Verwendung finden Sie unter [Skripte ausführen](#).

```

#!/bin/bash
# Update the following parameters
# -----
epmautomatescript="LOCATION_EPM_AUTOMATE_EXECUTABLE"
javahome="LOCATION_JAVA_HOME"
emailaddresses=EMAIL_ADDRESS_1,EMAIL_ADDRESS_2,EMAIL_ADDRESS_N
# -----

username="$1"
password="$2"
url="$3"

export JAVA_HOME=${javahome}

```

```

if [ "$#" -ne 3 ]; then
    echo "Usage: ./daily_maintenance_completed.sh <USERNAME>
<PASSWORD> <URL>"
    exit 1
fi

amw_time=""

getDailyMaintenanceStartTime() {
    amw_time=$((${epmautomatescript} getDailyMaintenanceStartTime | cut
-d' ' -f1)
}

goToSleep() {
    current_mdy=$(date -u +%m/%d/%Y)
    current_date_time=$(date -u)
    current_epoch=$(date +%s)
    target_epoch=$(date -d "${current_mdy} ${amw_time}" +%s)
    sleep_seconds=$((target_epoch - $current_epoch))

    # Today's AMW start time has already passed, so add 24 hours to
sleep_seconds
    if [[ ${sleep_seconds} -lt 0 ]]
    then
        sleep_seconds=$((sleep_seconds + 86400))
    fi

    sleep_hms=$(date -d@${sleep_seconds} -u +%H:%M:%S)

    echo "Current time is ${current_date_time}. Sleeping for $
${sleep_hms}, until daily maintenance start time of ${amw_time}."
    sleep $sleep_seconds
}

attemptLogin() {
    local serverdown=1
    while true
    do
        ${epmautomatescript} login ${username} ${password} ${url}
        if [[ $? -eq 0 ]] # login succeeded
        then
            if [[ ${serverdown} -eq 0 ]] # server has been brought down
            then
                echo "Daily maintenance processing has completed"
                break
            else # server has not yet been brought down
                echo "Daily maintenance processing has not yet
started. Sleeping for 2 minutes before the next check ..."
                sleep 120
            fi
        else # login failed
            if [[ ${serverdown} -eq 0 ]] # server has been brought down
            then
                echo "Waiting for daily maintenance processing to
complete. Sleeping for 2 minutes before the next check ..."

```

```

        sleep 120
    else # server has not yet been brought down
        echo "Daily maintenance processing is now beginning.
Sleeping for 2 minutes before the next check ..."
        sleep 120
        serverdown=0
    fi
done
}

sendNotification()
{
    local servername=$(echo "${url}" | cut -d '/' -f3- | rev | cut -d ':' -
f2- | rev)
    local subject="Daily maintenance processing has completed"
    local formattedmessage="Daily maintenance processing has completed for
server ${servername}"
    local emailaddresses=$(echo ${emailaddresses} | sed "s/,;/g")

    echo "Mailing report"
    ${epmautomatescript} sendmail "${emailaddresses}" "${subject}" Body="$
$formattedmessage"
}

echo "Beginning daily maintenance completion notification script."
echo "Logging into server ..."
${epmautomatescript} login ${username} ${password} ${url}
getDailyMaintenanceStartTime
goToSleep
attemptLogin
sendNotification
echo "Logging out of server ..."
${epmautomatescript} logout
echo "Script processing has completed."

```

Serverseitiges Groovy-Skript

Erstellen Sie das Groovy-Skript `daily_maintenance_completed`, indem Sie den folgenden Code kopieren. Informationen zum Aktualisieren des Skripts für die eigene Verwendung finden Sie unter [Skripte ausführen](#).

```

// Daily Maintenance Completed Notification script

// Update the following parameters
// -----
String username="USERNAME"
String password="PASSWORD"
String url="URL OF THE ENVIRONMENT"
String emailaddresses="EMAIL_ADDRESS_1,EMAIL_ADDRESS_2,EMAIL_ADDRESS_N"
// -----

def LogMessage(String message) {
    def date = new Date()
    def sdf = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss")

```

```

        println([' ' + sdf.format(date) + ' ] ' + message);
    }

    def LogOperationStatus(EpmAutomateStatus opstatus) {
        def returncode = opstatus.getStatus()
        if (returncode != 0){
            LogMessage(opstatus.getOutput())
        }
        LogMessage('return code: ' + returncode)
    }

    def getDailyMaintenanceStartTime(EpmAutomate automate) {
        LogMessage("Operation: getDailyMaintenanceStartTime")
        EpmAutomateStatus amwtimestatus =
        automate.execute('getDailyMaintenanceStartTime')
        LogOperationStatus(amwtimestatus)
        def amwstring=(amwtimestatus.getOutput())
        def elements=amwstring.split(' ')
        def amwtime=elements[0]
        return amwtime
    }

    def goToSleep(String amw_time){
        def date = new Date()
        def current_mdy = new SimpleDateFormat("MM/dd/yyyy")
        def current_date_time = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss")
        float current_epoch = date.getTime() / 1000
        def pattern = "MM/dd/yyyy HH:mm:ss"
        def input = current_mdy.format(date) + " " + amw_time + ":00"
        def target_date_time = Date.parse(pattern, input)
        float target_epoch = target_date_time.getTime() / 1000
        int sleep_seconds = Math.round(target_epoch - current_epoch)

        //Today's AMW start time has already passed, so add 24 hours to
        sleep_seconds
        if (sleep_seconds < 0) {
            sleep_seconds = sleep_seconds + 86400
        }

        def sleep_milliseconds = sleep_seconds * 1000
        LogMessage("Current time is " + current_date_time.format(date) +
        ". Sleeping until daily maintenance start time of " + amw_time +
        ":00.")
        sleep(sleep_milliseconds)
    }

    def attemptLogin(EpmAutomate automate, String username, String
    password, String url) {
        def serverdown=1
        while (true) {
            LogMessage("Operation: login " + username + " " + password + "
            " + url)
            EpmAutomateStatus status =
            automate.execute('login',username,password,url)
            def returncode = status.getStatus()

```

```

        if (returncode == 0) {
            if (serverdown == 0){
                LogMessage("Daily maintenance processing has completed ...")
                break
            } else {
                LogMessage("Daily maintenance processing has not yet
started. Sleeping for 2 minutes before the next check ...")
                sleep(120000)
            }
        } else {
            if (serverdown == 0){
                LogMessage("Waiting for daily maintenance processing to
complete. Sleeping for 2 minutes before the next check ...")
                sleep(120000)
            } else {
                LogMessage("Daily maintenance processing is now beginning.
Sleeping for 2 minutes before the next check ...")
                sleep(120000)
                serverdown=0
            }
        }
    }
}

def sendNotification(EpmAutomate automate, String url, String
emailaddresses) {
    def servername=url.tokenize("/")[-1];
    def subject="Daily maintenance processing has completed"
    def formattedmessage="Daily maintenance processing has completed for
server " + servername
    def emailaddressesformatted = emailaddresses.replaceAll(',',';')

    LogMessage("Operation: sendmail " + emailaddressesformatted + " " +
subject + " Body=" + formattedmessage)
    EpmAutomateStatus status =
automate.execute('sendmail',emailaddressesformatted,subject,'Body=' +
formattedmessage)
    LogOperationStatus(status)
}

LogMessage("Beginning daily maintenance completion notification script.")

EpmAutomate automate = getEpmAutomate()

LogMessage("Operation: login " + username + " " + password + " " + url)
EpmAutomateStatus status = automate.execute('login',username,password,url)
LogOperationStatus(status)

String amwtime = getDailyMaintenanceStartTime(automate)
goToSleep (amwtime)
attemptLogin(automate,username,password,url)
sendNotification(automate,url,emailaddresses)

LogMessage("Operation: logout ")
status = automate.execute('logout')

```

```
LogOperationStatus(status)
```

```
LogMessage ("Script processing has completed.")
```

Skripte ausführen

Windows und Linux/UNIX

1. Erstellen Sie die Datei `daily_maintenance_completed.ps1` oder `daily_maintenance_completed.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
2. Aktualisieren Sie das Skript:
 - **Windows:** Aktualisieren Sie den Wert von `emailaddresses` mit einer kommagetrennten Liste von E-Mail-Adressen, die benachrichtigt werden sollen, wenn die tägliche Wartung abgeschlossen ist.
 - **Linux/UNIX:** Aktualisieren Sie die folgenden Variablen:
 - `epmautomatescript` mit dem Speicherort der ausführbaren Datei für EPM Automate. Beispiel: `epmautomatescript="/home/utils/EPMAutomate/bin/epmautomate.sh"`
 - `javahome` mit dem Verzeichnis, in dem das von EPM Automate verwendete JDK installiert ist. Beispiel: `"/home/user1/jdk1.8.0_191"`
 - `emailaddresses` mit einer kommagetrennten Liste von E-Mail-Adressen, die benachrichtigt werden sollen, wenn die tägliche Wartung abgeschlossen ist. Beispiel: `jdoe@example.com,jane_doe@example.com`
3. Navigieren Sie in einem Befehlsfenster oder in einer Befehlskonsole zu dem Ordner, in dem das Skript `daily_maintenance_completed` gespeichert ist.
4. Führen Sie diesen Befehl aus:
 - **Windows:** `./daily_maintenance_completed.ps1 USERNAME PASSWORD URL`
 - **Linux/UNIX:** `./daily_maintenance_completed.sh USERNAME PASSWORD URL`, wobei Folgendes gilt:
 - `USERNAME` ist der Benutzername eines Serviceadministrators.
 - `PASSWORD` ist das Kennwort des Serviceadministrators.
 - `URL` ist die URL der EPM Cloud-Umgebung.

Serverseitiges Groovy-Skript:

1. Erstellen Sie das Groovy-Skript `daily_maintenance_completed.groovy`, indem Sie es aus einem vorhergehenden Abschnitt kopieren.
2. Aktualisieren Sie die folgenden Werte:
 - `username` mit dem Benutzernamen eines Serviceadministrators.
 - `password` mit dem Kennwort des Serviceadministrators
 - `url` mit der URL der EPM Cloud-Umgebung, für die die Benachrichtigung über den Abschluss der täglichen Wartung erfolgen muss. Beispiel: `. Beispiel: https://testExample-idDomain.pbcs.us1.oraclecloud.com`
 - `emailaddresses` mit einer kommagetrennten Liste von E-Mail-Adressen, die benachrichtigt werden sollen, wenn die tägliche Wartung abgeschlossen ist.

3. Verwenden Sie das Groovy-Fenster in einem EPM Cloud-Geschäftsprozess, oder automatisieren Sie die Skriptausführung mit [runBusinessRule](#). Weitere Informationen finden Sie in den folgenden Informationsquellen:
 - [Befehle ohne Installation von EPM Automate ausführen](#)
 - Groovy-Regeln verwenden in der Dokumentation *Planning verwalten*

Snapshot in oder aus Oracle Object Storage kopieren

Dieses Thema enthält Beispielskripte zum Abschließen der folgenden Aufgaben:

- Kopieren Sie den Parameter `Artifact Snapshot` (den Wartungs-Snapshot) aus Oracle Enterprise Performance Management Cloud in eine Oracle Object Storage-Gruppe und benennen Sie ihn um, indem Sie das Datum anhängen, an dem der Snapshot kopiert wurde.
- Kopieren Sie einen Backup-Snapshot aus einer Oracle Object Storage-Gruppe in EPM Cloud.

Für die Skripte in diesem Abschnitt wird vorausgesetzt, dass Sie bereits eine Gruppe in Oracle Object Storage erstellt haben, die den Snapshot enthalten soll. Bevor Sie diese Skripte ausführen, müssen Sie sie für Ihre Verwendung anpassen, indem Sie die folgenden Platzhalter aktualisieren:

Table 3-1 Parameter und deren Werte

Platzhalter	Erwarteter Wert
<code>JAVA_HOME</code>	Verzeichnis, in dem das von EPM Automate verwendete JDK installiert ist. Beispiel: <code>./home/JDK/bin</code>
<code>epmautomateExe</code>	Verzeichnis, in dem EPM Automate installiert ist. Beispiel: <code>./home/utills/EPMAutomate/bin</code>
<code>cloudServiceUser</code>	Benutzer-ID eines EPM Cloud-Serviceadministrators. Beispiel: <code>John.doe@example.com</code>
<code>cloudServicePassword</code>	Kennwort des Serviceadministrators oder der Speicherort der Kennwortdatei. Wenn das Kennwort Sonderzeichen enthält, finden Sie weitere Informationen unter Sonderzeichen verarbeiten . Beispiel: <code>ex_PWD_213</code>
<code>cloudServiceUrl</code>	URL der EPM Cloud-Umgebung, aus der der Parameter <code>Artifact Snapshot</code> kopiert werden soll. Beispiel: <code>https://test-cloud-id_Dom.pbcs.us1.oraclecloud.com</code>
<code>objectStorageUser</code>	Benutzer-ID eines Benutzers in Oracle Object Storage. Um einen Snapshot in Object Storage zu kopieren, muss dieser Benutzer über Schreibzugriff für die Gruppe verfügen, in die der Snapshot kopiert wird. Um einen Snapshot aus Object Storage zu kopieren, muss dieser Benutzer über Lesezugriff für die Gruppe verfügen, aus der der Snapshot kopiert wird. Beispiel: <code>jDoe</code>
<code>objectStoragePassword</code>	Kennwort des <code>objectStorageUser</code> . Beispiel: <code>example_PWD</code>

Table 3-1 (Cont.) Parameter und deren Werte

Platzhalter	Erwarteter Wert
objectStorageBucketUrl	<p>URL der Oracle Object Storage-Gruppe, in die der Snapshot kopiert werden soll. In den folgenden Quellen finden Sie Informationen zum URL-Format:</p> <ul style="list-style-type: none"> • copyToObjectStorage • copyFromObjectStorage <p>Beispiel: https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/axaxnpcrorw5/bucket-20210301-1359</p>
snapshot	<p>Name des Snapshots, den Sie aus der Oracle Object Storage-Gruppe kopieren möchten.</p> <p>Beispiel: Artifact Snapshot20210429.zip</p>

EPM Automate-Beispielskript zum Kopieren eines Snapshots aus EPM Cloud in Oracle Object Storage

Passen Sie dieses Skript an, und führen Sie es aus, um den Parameter `Artifact Snapshot` umzubenennen und aus EPM Cloud in eine Oracle Object Storage-Gruppe zu kopieren.

```
#!/bin/sh
export JAVA_HOME=<path_to_jdk>
epmautomateExe=<path_to_epmautomate_executable>
cloudServiceUser=<cloud_service_user>
cloudServicePassword=<cloud_service_password>
cloudServiceUrl=<cloud_service_url>
# User with write access to Object Storage bucket
objectStorageUser=<object_storage_user>
objectStoragePassword=<object_storage_password>
objectStorageBucketUrl=<object_storage_bucket>
currentDate=`date +%Y%m%d`
sourceSnapshot="Artifact Snapshot"
targetSnapshot="${sourceSnapshot} ${currentDate}"
$epmautomateExe login ${cloudServiceUser} ${cloudServicePassword} $
{cloudServiceUrl}
$epmautomateExe renamesnapshot "${sourceSnapshot}" "${targetSnapshot}"
$epmautomateExe copyToObjectStorage "${targetSnapshot}" $
{objectStorageUser} ${objectStoragePassword} "$
{objectStorageBucketUrl}/${targetSnapshot}"
$epmautomateExe logout
exit 0
```

EPM Automate-Beispielskript zum Kopieren eines Snapshots aus Oracle Object Storage in EPM Cloud

Passen Sie dieses Skript an, und führen Sie es aus, um den Parameter `Artifact Snapshot` eines bestimmten Datums aus einer Oracle Object Storage-Gruppe in EPM Cloud zu kopieren.

```
#!/bin/sh
export JAVA_HOME=<path_to_jdk>
epmautomateExe=<path_to_epmautomate_executable>
cloudServiceUser=<cloud_service_user>
cloudServicePassword=<cloud_service_password>
cloudServiceUrl=<cloud_service_url>
# User with read access to Object Storage bucket
objectStorageUser=<object_storage_user>
objectStoragePassword=<object_storage_password>
objectStorageBucketUrl=<object_storage_bucket>
snapshot=<desired_snapshot>
$epmautomateExe login ${cloudServiceUser} ${cloudServicePassword} $
{cloudServiceUrl}
$epmautomateExe copyFromObjectStorage ${objectStorageUser} $
{objectStoragePassword} "${objectStorageBucketUrl}/${snapshot}"
$epmautomateExe logout
exit 0
```

Benutzer erstellen und vordefinierten Rollen zuweisen

Verwenden Sie die Skripte in diesem Abschnitt, um Benutzer in der Identitätsdomain zu erstellen und sie vordefinierten Rollen zuzuweisen.

Diese Skripte verwenden EPM Automate-Befehle zur Ausführung der folgenden Aktivitäten:

- Melden Sie sich bei der Umgebung als Serviceadministrator mit der Rolle Identitätsdomainadministrator an.
- Exportieren Sie Gruppen- und Mitgliedschaftsinformationen aus der Umgebung, um einen von Ihnen angegebenen Snapshot neu zu generieren. Beispiel: `example_snapshot.zip`. Es wird angenommen, dass Sie Gruppen- und Mitgliedschaftsartefakte bereits zuvor mit der Option Migration exportiert haben, um diesen Snapshot zu erstellen.
- Laden Sie den Snapshot (`example_snapshot.zip`) in ein lokales Verzeichnis herunter.
- Löschen Sie den Snapshot (`example_snapshot.zip`) aus der Umgebung.
- Melden Sie sich von der Umgebung ab.
- Führen Sie Vorgänge aus, für die Sie benutzerdefinierten Code hinzugefügt haben. Beispiele für solche Vorgänge sind:
 - Inhalt der Datei `example_snapshot.zip` extrahieren.
 - Informationen zu neuen Benutzern im Format "Vorname,Nachname,E-Mail-Adresse,Benutzeranmeldedaten" an die Datei `HSS-Shared Services\resource\External Directory\Users.csv` anhängen. Ausführliche Informationen finden Sie unter Batch mit Benutzeraccounts importieren in der Dokumentation *Erste Schritte mit Oracle Cloud*.

- Informationen zu Rollenzuweisungen für neue Benutzer (im Format "Vorname,Nachname,E-Mail-Adresse,Benutzeranmeldedaten") an die entsprechenden Rollendateien in der Datei `HSS-Shared Services\resource\External Directory\Roles\` anhängen. Beispiel: Zuweisungen zur Rolle `Serviceadministrator` müssen an `<service_name> Service Administrator.csv` angehängt werden. Zuweisungen zur Rolle "Leseberechtigter" müssen hingegen an `HSS-Shared Services\resource\External Directory\Roles\<service_name> Viewer.csv` angehängt werden. Ausführliche Informationen finden Sie unter *Vielen Benutzern eine Rolle zuweisen* in der Dokumentation *Erste Schritte mit Oracle Cloud*.
 - Den Snapshot mit den aktualisierten Dateien durch Komprimieren des Verzeichnisses `HSS-Shared Services` und seines Inhalts neu erstellen.
- Melden Sie sich bei der Umgebung als `Serviceadministrator` an, der auch über die Rolle `Identitätsdomainadministrator` verfügt.
 - Laden Sie die geänderte Datei `example_snapshot.zip` in die Umgebung hoch.
 - Importieren Sie die Datei `example_snapshot.zip` in die Umgebung.
 - Löschen Sie die hochgeladene Datei `example_snapshot.zip` aus der Umgebung.
 - Melden Sie sich ab.

Windows-Beispielskript

Erstellen Sie eine Datei namens `createUsersAndAssignRoles.ps1`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./
input.properties -raw)
$username="$($inputproperties.username)"
$passwordfile="$($inputproperties.passwordfile)"
$serviceURL="$($inputproperties.serviceURL)"
$snapshotName="$($inputproperties.snapshotName)"
$userspassword="$($inputproperties.userspassword)"
$resetPassword="$($inputproperties.resetPassword)"

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate exportsnapshot ${snapshotName}
epmautomate downloadfile ${snapshotName}.zip
epmautomate deletefile ${snapshotName}.zip
epmautomate logout

# Add custom code to extract the contents of example_snapshot.zip
# Add custom code to append new user information to HSS-Shared
Services\resource\External Directory\Users.csv
# Optional: Add custom code to append role information to the
appropriate role file(s) in HSS-Shared Services\resource\External
Directory\Roles\
# Add custom code to zip HSS-Shared Services and its contents as
example_snapshot.zip

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile ${snapshotName}.zip
```

```
epmautomate importsnapshot ${snapshotName} userPassword=${userspassword}
resetPassword=${resetPassword}
epmautomate deletefile ${snapshotName}.zip
epmautomate logout
```

Linux-/UNIX-Beispielskript

Erstellen Sie eine Datei namens `createUsersAndAssignRoles.sh`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
#!/bin/bash

. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "${serviceURL}"
${epmautomatescript} exportsnapshot "${snapshotName}"
${epmautomatescript} downloadfile "${snapshotName}.zip"
${epmautomatescript} deletefile "${snapshotName}.zip"
${epmautomatescript} logout

# Add custom code to extract the contents of example_snapshot.zip
# Add custom code to append new user information to HSS-Shared
Services\resource\External Directory\Users.csv
# Optional: Add custom code to append role information to the appropriate
role file(s) in HSS-Shared Services\resource\External Directory\Roles\
# Add custom code to zip HSS-Shared Services and its contents as
example_snapshot.zip

${epmautomatescript} login "${username}" "${passwordfile}" "${serviceURL}"
${epmautomatescript} uploadfile "${snapshotName}.zip"
${epmautomatescript} importsnapshot "${snapshotName}" "userPassword=${
userspassword}" "resetPassword=${resetPassword}"
${epmautomatescript} deletefile "${snapshotName}.zip"
${epmautomatescript} logout
```

Beispiel für die Datei "input.properties"

Um die `createUsersAndAssignRoles`-Skripte auszuführen, erstellen Sie die Datei `input.properties`, und aktualisieren Sie sie mit den Informationen für Ihre Umgebung. Speichern Sie die Datei im selben Verzeichnis, in dem die Datei `createUsersAndAssignRoles.ps1` oder `createUsersAndAssignRoles.sh` gespeichert ist.

Windows

```
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
snapshotName=SNAPSHOT_NAME
userspassword=TEMP_IDM_PASSWORD
resetPassword=true
```

Linux/UNIX

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
snapshotName=SNAPSHOT_NAME
userspassword=TEMP_IDM_PASSWORD
resetPassword=true
```

Tabelle 3-2 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis <code>JAVA_HOME</code> . Nur für Linux/UNIX.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (<code>epmautomate.sh</code>). Nur für Linux/UNIX.
username	Benutzername eines Serviceadministrators, der auch über die Rolle Identitätsdomainadministrator verfügt.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
serviceURL	URL der Umgebung, aus der Sie den Snapshot generieren möchten.
snapshotName	Der Name des Snapshots, den Sie generieren möchten. Es wird angenommen, dass Sie Gruppen- und Mitgliedschaftsartefakte bereits zuvor mit der Option Migration exportiert haben, um diesen Snapshot zu erstellen.
userspassword	Das Anfangskennwort für neue Benutzer.
resetPassword	Angabe, ob neue Benutzer das Kennwort bei der ersten Anmeldung neu festlegen müssen. Setzen Sie diesen Wert auf <code>true</code> , um zu erzwingen, dass neue Benutzer ihr Kennwort bei ihrer ersten Anmeldung ändern.

Skripte ausführen

1. Erstellen Sie die Datei `createUsersAndAssignRoles.ps1` oder `createUsersAndAssignRoles.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
2. Fügen Sie benutzerdefinierten Code hinzu, um folgende Vorgänge auszuführen:
 - Inhalt des Snapshots extrahieren
 - Informationen zu neuen Benutzern an die Datei `HSS-Shared Services\resource\External Directory\Users.csv` anhängen
 - Optional: Informationen zu Rollenzuweisungen für neue Benutzer (im Format "Vorname,Nachname,E-Mail-Adresse,Benutzeranmeldedaten") an die entsprechenden Rollendateien in der Datei `HSS-Shared Services\resource\External Directory\Roles\` anhängen
 - Den Snapshot mit den aktualisierten Dateien neu erstellen

3. Erstellen Sie die Datei `input.properties`, und speichern Sie sie im selben Verzeichnis, in dem sich das Skript `createUsersAndAssignRoles` befindet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Informationen hierzu finden Sie unter [Beispiel für die Datei "input.properties"](#).
Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.
4. Starten Sie das Skript.
 - **Windows PowerShell:** Führen Sie die Datei `createUsersAndAssignRoles.ps1` aus.
 - **Linux/UNIX:** Führen Sie die Datei `./createUsersAndAssignRoles.sh` aus.

Anzahl der lizenzierten Benutzer zählen (Benutzer, die Rollen zugewiesen sind)

Verwenden Sie das Skript in diesem Abschnitt, um den Bericht zur Rollenzuweisung zum Zählen der Anzahl an Benutzern für eine Umgebung zu generieren.

Erstellen Sie `provisionedUsersCount.bat`, indem Sie das folgende Skript kopieren.

Hinweis:

- Die Eingabeparameter für die Ausführung von `provisionedUsersCount.bat` sind `username`, `password/password_file`, `service_url` und `report_file_name`.
Beispiel: Geben Sie in der Eingabeaufforderung einen Befehl ähnlich dem folgenden ein:

```
provisionedUsersCount.bat jdoe password.epw https://example.oraclecloud.com myRole_assign.CSV
```
- Wenn das Kennwort Sonderzeichen enthält, finden Sie unter [Sonderzeichen verarbeiten](#) weitere Informationen.

```
@echo off

set paramRequiredMessage=Syntax: provisionedUsersCount.bat USERNAME PASSWORD/
PASSWORD_FILE URL REPORT_FILE_NAME

if "%~1" == "" (
    echo User Name is missing.
    echo %paramRequiredMessage%
    exit /b 1
)

if "%~2" == "" (
    echo Password or Password_File is missing.
    echo %paramRequiredMessage%
    exit /b 1
)

if "%~3" == "" (
```

```

    echo URL is missing.
    echo %paramRequiredMessage%
    exit /b 1
)

if "%~4" == "" (
    echo Role Assignment Report File Name is missing.
    echo %paramRequiredMessage%
    exit /b 1
)

call epmautomate.bat login %~1 "%~2" %~3
REM call epmautomate.bat login %~1 "%~2" %~3

if %errorlevel% NEQ 0 exit /b 1
    call epmautomate.bat roleAssignmentReport "%5"
if %errorlevel% NEQ 0 exit /b 1
    call epmautomate.bat downloadFile "%5"
if %errorlevel% NEQ 0 exit /b 1

set filePath="%cd%\%4"

if exist %filePath% (
    SETLOCAL EnableDelayedExpansion
    set /a lineCount=0
    set /a userCount=0
    set userHeaderFound=false
    for /f "tokens=*" %%A in ( 'type %filePath%' ) do (
        set /a lineCount+=1
        set line=%%A

        REM Fetch username from role assignment information row
        if !userHeaderFound!==true (
            for /f "delims=" %%i in ("!line!") do (
                set userName=%%i
            )
            if NOT !userName! == "" (
                if !userCount! gtr 0 if NOT !userName! == !lastUserName! (
                    set /a userCount+=1
                    set users[!userCount!]=!userName!
                )
                if !userCount! == 0 (
                    set /a userCount+=1
                    set users[!userCount!]=!userName!
                )
            )
            set lastUserName=!userName!
        )
    )

    REM Check for headers of Role Assignment Report
    if "!line!"=="User Login,First Name,Last
Name,Email,Role,Granted through Group" (
        set userHeaderFound=true
    )
    if "!line!"=="User Login,First Name,Last Name,Email,Roles,Granted

```

```

Through" (
    set userHeaderFound=true
    )
)

    echo Number of Users: !userCount!
    for /l %%n in (1,1,!userCount!) do (
REM echo !users[%%n]!
    )
    endlocal

) else (
    echo Invalid provisioning report file path - %filePath%.
)

```

Auditberichte zu Benutzern erstellen, die Rollen zugewiesen sind

Verwenden Sie die Skripte in diesem Abschnitt, um den Prozess zum Erstellen eines Auditberichts für Benutzer, die vordefinierten Rollen in einer Umgebung zugewiesen sind, zu automatisieren und den Bericht optional per E-Mail an einen Empfänger zu senden.

Dieser Auditbericht zeigt die Benutzer, die vordefinierten Rollen oder Gruppen zugewiesen sind, die seit der letzten Berichtsgenerierung geändert wurden. Um einen täglichen Auditbericht zu erstellen, führen Sie dieses Skript täglich aus.

Erstellen Sie `provisioningAuditReport.bat`, indem Sie das folgende Skript kopieren. Dieses Wrapper-Batchskript ruft das PowerShell-Skript `provisioningAuditReport.ps1` auf, dessen Quellcode später in diesem Szenario bereitgestellt wird.

Hinweis:

- Eingabeparameter zum Ausführen von `provisioningAuditReport.bat`: `username`, `password` oder `password_file`, `service_url` und `report_email_to_address` (optional, nur erforderlich, wenn Sie den Bericht an eine E-Mail-Adresse senden möchten).
- Wenn das Kennwort Sonderzeichen enthält, finden Sie unter [Sonderzeichen verarbeiten](#) weitere Informationen.

```

@echo off
set paramRequiredMessage=Syntax: provisioningAuditReport.bat USERNAME
PASSWORD/PASSWORD_FILE URL [REPORT_EMAIL_TO_ADDRESS]

if "%~1" == "" (
    echo User Name is missing.
    echo %paramRequiredMessage%
    exit /b 1
)
if "%~2" == "" (
    echo Password or Password_File is missing.
    echo %paramRequiredMessage%

```



```

        exit /b 1
    )
    if "%~3" == "" (
        echo URL is missing.
        echo %paramRequiredMessage%
        exit /b 1
    )

```

```
PowerShell.exe -File provisioningAuditReport.ps1 %*
```

provisioningAuditReport.bat ruft das Skript provisioningAuditReport.ps1 auf, das Sie erstellen, indem Sie das folgende Skript kopieren.

provisioningAuditReport.ps1 erstellt den Auditbericht. Speichern Sie das Skript in demselben Verzeichnis, in dem sich provisioningAuditReport.bat befindet.

```

$username=$args[0]
$password=$args[1]
$url=$args[2]
$reportemailtoaddress=$args[3]

$date=$(get-date -f dd_MM_yy_HH_mm_ss)
$datedefaultformat=$(get-date)
$logdir="./logs/"
$logfile="$logdir/epmautomate-provisionauditreport-" + $date + ".log"
$reportdir="./reports/"
$provisionreport="provreport-auditest-" + $date + ".csv"
$provisionreporttemp="./provreport-auditest-temp.csv"
$provisionreportunique="./provreport-auditest-unique.csv"
$provisionreportbaselineunique="./provreport-auditest-baseline-unique.csv"

function EchoAndLogMessage
{
    $message=$args[0]
    echo "$message"
    echo "$message" >> $logfile
}

function Init
{
    $logdirexists=Test-Path $logdir
    if (!$logdirexists) {
        mkdir $logdir 2>&1 | out-null
    }

    $logfileexists=Test-Path $logfile
    if ($logfileexists) {
        rm $logfile 2>&1 | out-null
    }

    $reportdirexists=Test-Path $reportdir
    if (!$reportdirexists) {
        mkdir $reportdir 2>&1 | out-null
    }
}

```

```

    }
}

function PostProcess
{
    rm $provisionreporttemp
    mv -Force $provisionreportunique $provisionreportbaselineunique
}

function ProcessCommand
{
    $op=$args
    echo "EPM Automate operation: epmautomate.bat $op" >> $logfile
    epmautomate.bat $op >> $logfile 2>&1
    if ($LASTEXITCODE -ne 0) {
        echo "EPM Automate operation failed: epmautomate.bat $op. See $logfile
for details."
        exit
    }
}

function RunEpmAutomateCommands
{
    EchoAndLogMessage "Running EPM Automate commands to generate the
provisioning report."
    ProcessCommand login $username $password $url
    ProcessCommand provisionreport $provisionreport
    ProcessCommand downloadfile $provisionreport
    ProcessCommand deletefile $provisionreport
    ProcessCommand logout
}

function CreateProvisionReportTempFile
{
    # Loop through iteration csv file and parse
    Get-Content $provisionreport | ForEach-Object {
        $elements=$_.split(',')
        echo "$($elements[0]), $($elements[2])" >> $provisionreporttemp
    }
}

function CreateUniqueElementsFile
{
    gc $provisionreporttemp | sort | get-unique > $provisionreportunique
}

function CheckBaselineAndCreateAuditReport
{
    $provisionreportbaselineuniqueexists=Test-
Path $provisionreportbaselineunique
    if (!$provisionreportbaselineuniqueexists) {
        EchoAndLogMessage "No existing provisioning report, so comparison with a
baseline is not possible. Audit report will be created at the next test run."
    } else {
        CreateAuditReport
    }
}

```

```

    }
}

function EmailAuditReport
{
    $auditreport=$args[0]
    $elements=$auditreport.split('/')
    $auditreportname=$elements[2]

    if (${reportemailtoaddress} -match "@") {
        EchoAndLogMessage "Emailing audit report"
        ProcessCommand login $username $password $url
        ProcessCommand uploadFile $auditreport
        ProcessCommand sendMail $reportemailtoaddress "Provisioning
Audit Report" Body="Provisioning Audit Report is attached."
Attachments=$auditreportname
        ProcessCommand deleteFile $auditreportname
        ProcessCommand logout
    }
}

function CreateAuditReport
{
    $auditreport=$reportdir + "auditreport-"+ $date + ".txt"
    $additions = @()
    $deletions = @()

    EchoAndLogMessage "Comparing previous provisioning report with the
current report."
    $compare=compare-object (get-content $provisionreportunique) (get-
content $provisionreportbaselineunique)

    $compare | foreach {
        if ($_.sideindicator -eq '<=')
        {
            $additions += $_.inputobject
        } elseif ($_.sideindicator -eq '=>') {
            $deletions += $_.inputobject
        }
    }

    echo "Provisioning Audit Report for $datedefaultformat"
> $auditreport
    echo "-----"
>> $auditreport

    if ($additions.count -ne 0)
    {
        echo " " >> $auditreport
        echo "Additions:" >> $auditreport
        foreach($element in $additions) { echo "$element" >> $auditreport }
    }

    if ($deletions.count -ne 0)
    {

```

```

    echo " " >> $auditreport
    echo "Deletions:" >> $auditreport
    foreach($element in $deletions) { echo "$element" >> $auditreport }
}

if (($additions.count -eq 0) -and ($deletions.count -eq 0))
{
    echo " " >> $auditreport
    echo "No changes from last audit report." >> $auditreport
}

EchoAndLogMessage "Provisioning audit report has been
generated: $auditreport."
EmailAuditReport $auditreport
}

Init
EchoAndLogMessage "Starting EPMAutomate provisioning audit reporting"
RunEpmAutomateCommands
CreateProvisionReportTempFile
CreateUniqueElementsFile
CheckBaselineAndCreateAuditReport
PostProcess
EchoAndLogMessage "EPMAutomate provisioning audit reporting completed"

```

Auditberichte zu Rollenzuweisungen und zur Aufhebung von Zuweisungen

Verwenden Sie das PowerShell-Skript in diesem Abschnitt, um den Prozess zum Erstellen eines Auditberichts zu automatisieren, in dem die Rollenzuweisung und die Rollenwiderrufung in einer Umgebung dargelegt sind.

Erstellen Sie `AuditReportRoleAssignment.bat`, indem Sie das folgende Skript kopieren. Dieses Wrapper-Batchskript ruft das PowerShell-Skript `AuditReportRoleAssignment.ps1` auf, dessen Quellcode später in diesem Szenario bereitgestellt wird.

Hinweis:

- Die Eingabeparameter für die Ausführung von `AuditReportRoleAssignment.bat` sind `username`, `password` oder `password_file` und `service_url`.
- Wenn das Kennwort Sonderzeichen enthält, finden Sie unter [Sonderzeichen verarbeiten](#) weitere Informationen.

Skript: `AuditReportRoleAssignment.bat`

```

@echo off
set paramRequiredMessage=Syntax: AuditReportRoleAssignment.bat USERNAME
PASSWORD/PASSWORD_FILE URL

if "%~1" == "" (

```

```

        echo User Name is missing.
        echo %paramRequiredMessage%
        exit /b 1
    )
if "%~2" == "" (
    echo Password or Password_File is missing.
    echo %paramRequiredMessage%
    exit /b 1
)
if "%~3" == "" (
    echo URL is missing.
    echo %paramRequiredMessage%
    exit /b 1
)

PowerShell.exe -File AuditReportRoleAssignment.ps1 %*

```

Skript: AuditReportRoleAssignment.ps1

```

# EPM Automate Role Assignment Audit Report Script
$username=$args[0]
$password=$args[1]
$url=$args[2]

# Generic variables
$date=$(get-date -f dd_MM_yy_HH_mm_ss)
$datedefaultformat=$(get-date)
$logdir="./logs/"
$logfile="$logdir/epmautomate-provisionauditreport-" + $date + ".log"
$reportdir="./reports/"
$provisionreport="provreport-auditteest-" + $date + ".csv"
$provisionreporttemp="./provreport-auditteest-temp.csv"
$provisionreportunique="./provreport-auditteest-unique.csv"
$provisionreportbaselineunique="./provreport-auditteest-baseline-unique.csv"

function EchoAndLogMessage
{
    $message=$args[0]
    echo "$message"
    echo "$message" >> $logfile
}

function Init
{
    $logdirexists=Test-Path $logdir
    if (!$logdirexists) {
        mkdir $logdir 2>&1 | out-null
    }
    $logfileexists=Test-Path $logfile
    if ($logfileexists) {
        rm $logfile 2>&1 | out-null
    }
    $reportdirexists=Test-Path $reportdir
    if (!$reportdirexists) {

```

```

        mkdir $reportdir 2>&1 | out-null
    }
}

function PostProcess
{
    rm $provisionreporttemp
    mv -Force $provisionreportunique $provisionreportbaselineunique
}

function ProcessCommand
{
    $op=$args
    echo "EPM Automate operation: epmautomate.bat $op" >> $logfile
    epmautomate.bat $op >> $logfile 2>&1
    if ($LASTEXITCODE -ne 0) {
        echo "EPM Automate operation failed: epmautomate.bat $op.
See $logfile for details."
        exit
    }
}

function RunEpmAutomateCommands
{
    EchoAndLogMessage "Running EPM Automate commands to generate the audit
report."
    ProcessCommand login $username $password $url
    ProcessCommand provisionreport $provisionreport
    ProcessCommand downloadfile $provisionreport
    ProcessCommand deletefile $provisionreport
    ProcessCommand logout
}

function CreateProvisionReportTempFile
{
    # Loop through iteration csv file and parse
    Get-Content $provisionreport | ForEach-Object {
        $elements=$_split(',')
        echo "$($elements[0]), $($elements[2])" >> $provisionreporttemp
    }
}

function CreateUniqueElementsFile
{
    gc $provisionreporttemp | sort | get-unique > $provisionreportunique
}

function CheckBaselineAndCreateAuditReport
{
    $provisionreportbaselineuniqueexists=Test-
Path $provisionreportbaselineunique
    if (!$provisionreportbaselineuniqueexists) {
        EchoAndLogMessage "Could not find a baseline audit report to compare
with. Audit report will be created next time you run test."
    } else {
        CreateAuditReport
    }
}

```

```

    }
}

function CreateAuditReport
{
    $auditreport=$reportdir + "auditreport-"+ $date + ".txt"
    $additions = @()
    $deletions = @()
    EchoAndLogMessage "Comparing previous audit report with the
current one."
    $compare=compare-object (get-content $provisionreportunique) (get-
content $provisionreportbaselineunique)
    $compare | foreach {
        if ($_.sideindicator -eq '<=')
        {
            $additions += $_.inputobject
        } elseif ($_.sideindicator -eq '=>') {
            $deletions += $_.inputobject
        }
    }
    echo "Provisioning Audit Report for $datedefaultformat"
> $auditreport
    echo "-----"
>> $auditreport
    if ($additions.count -ne 0)
    {
        echo " " >> $auditreport
        echo "Additions:" >> $auditreport
        foreach($element in $additions) { echo "$element"
>> $auditreport }
    }
    if ($deletions.count -ne 0)
    {
        echo " " >> $auditreport
        echo "Deletions:" >> $auditreport
        foreach($element in $deletions) { echo "$element"
>> $auditreport }
    }
    if (($additions.count -eq 0) -and ($deletions.count -eq 0))
    {
        echo " " >> $auditreport
        echo "No changes from last audit report." >> $auditreport
    }
    EchoAndLogMessage "Role audit report generated: $auditreport."
}

Init
EchoAndLogMessage "Starting EPMAutomate role audit report generation"
RunEpmAutomateCommands
CreateProvisionReportTempFile
CreateUniqueElementsFile
CheckBaselineAndCreateAuditReport
PostProcess
EchoAndLogMessage "EPMAutomate role audit report completed"

```

Zugriffslogs und Aktivitätsberichte zur Einhaltung von Datenschutzgesetzen maskieren

Verwenden Sie die Skripte in diesem Abschnitt, um den Prozess zum Maskieren von Informationen im Aktivitätsbericht oder in Zugriffslogs den Datenschutzgesetzen entsprechend zu automatisieren und den Bericht optional per E-Mail an einen Empfänger zu senden.

Aufgrund der strengen Datenschutzgesetze einiger Länder müssen die in den Aktivitätsberichten und Zugriffslogs verfügbaren Informationen möglicherweise für Serviceadministratoren ausgeblendet werden, um private Daten von Benutzern zu schützen.

Verwenden Sie `anonymizeData.bat`, um Informationen in den Aktivitätsberichten oder Zugriffslogs zur Einhaltung von Datenschutzgesetzen zu maskieren und optional per E-Mail zu versenden. Zum Maskieren von Informationen planen Sie dieses Skript oder eine Variante des Skripts bei Verwendung von Windows Scheduler, sodass es jeden Tag ausgeführt wird, nachdem der tägliche Wartungsprozess für die einzelnen Umgebungen abgeschlossen ist.

Verwenden Sie diese Informationsquellen:

- Aktivitätsberichte und Zugriffslogs zum Überwachen der Verwendung verwenden in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*
- [Skriptausführung automatisieren](#)

Sie erstellen `anonymizeData.bat` manuell, indem Sie das im Folgenden angegebene Windows-Skript kopieren und mit Windows Scheduler planen. Sie können ähnliche plattformgeeignete Skripte erstellen und ausführen, wenn Sie die Planung nicht mit Windows vornehmen.

`anonymizeData.bat` ist ein Wrapper-Skript zur Ausführung des Skripts `anonymizeData.ps1`, das Sie wie im Folgenden beschrieben erstellen und aktualisieren.

Wenn das Kennwort, das Sie verwenden, Sonderzeichen enthält, finden Sie unter [Sonderzeichen verarbeiten](#) weitere Informationen.

1. Erstellen Sie eine Batchdatei (BAT) mit dem Namen `anonymizeData.bat`, die das folgende Skript enthält, und speichern Sie sie an einem entsprechenden Speicherort, z.B. `C:\automate_scripts`.

```
@echo off
set paramRequiredMessage=Syntax: anonymizeData.bat USERNAME PASSWORD/
PASSWORD_FILE URL [EMAIL_TO_ADDRESS]

if "%~1" == "" (
    echo User Name is missing.
    echo %paramRequiredMessage%
    exit /b 1
)
if "%~2" == "" (
    echo Password or Password_File is missing.
    echo %paramRequiredMessage%
    exit /b 1
)
if "%~3" == "" (
```



```

echo URL is missing.
echo %paramRequiredMessage%
exit /b 1
)

```

```
PowerShell.exe -File anonymizeData.ps1 %*
```

- Erstellen Sie ein PowerShell-Skript (PS1) mit dem Namen `anonymizeData.ps1`, das das folgende Skript enthält, und speichern Sie es an einem entsprechenden Speicherort, z.B. `C:\automate_scripts`.

```

# Anonymize data script

$username=$args[0]
$password=$args[1]
$url=$args[2]
$emailtoaddress=$args[3]

# Generic variables
$date=$(get-date -f dd_MM_yy_HH_mm_ss)
$datedefaultformat=$(get-date)
$logdir="./logs/"
$logfile="$logdir/anonymize-data-" + $date + ".log"
$filelist="filelist.txt"

function LogMessage
{
    $message=$args[0]

    echo "$message" >> $logfile
}

function EchoAndLogMessage
{
    $message=$args[0]

    echo "$message"
    echo "$message" >> $logfile
}

function Init
{
    $logdirexists=Test-Path $logdir
    if (!$logdirexists) {
        mkdir $logdir 2>&1 | out-null
    }

    $logfileexists=Test-Path $logfile
    if ($logfileexists) {
        rm $logfile 2>&1 | out-null
    }

    $filelistexists=Test-Path $filelist
    if ($filelistexists) {
        rm $filelist 2>&1 | out-null
    }
}

```

```

}

function ProcessCommand
{
    $op=$args
    echo "EPM Automate operation: epmautomate.bat $op" >> $logfile
    if ($op -eq 'listfiles') {
        epmautomate.bat $op | where {$? -like ' apr/*/access_log.zip'} |
Tee-Object -FilePath $filelist | Out-File $logfile -Append 2>&1
    } else {
        epmautomate.bat $op >> $logfile 2>&1
        if ($LASTEXITCODE -ne 0) {
            echo "EPM Automate operation failed: epmautomate.bat $op.
See $logfile for details."
            #exit
        }
    }
}

function RunEpmAutomateCommands
{
    EchoAndLogMessage "Running EPM Automate commands to anonymize data in
the access logs and activity reports."
    ProcessCommand login $username $password $url
    ProcessCommand listfiles
    ProcessFiles
    ProcessCommand logout
}

function ProcessActivityReport
{
    $activityreport=$args[0]
    $user=$args[1]

    $activityreportexists=Test-Path "$activityreport"
    if ($activityreportexists) {
        LogMessage "Removing User ID: $user from activity
report $activityreport"
        (Get-Content "$activityreport").replace("$user", 'XXXXX') | Set-
Content "$activityreport"
        $txt = [io.file]::ReadAllText("$activityreport") -replace
" `r`n", "`n"
        [io.file]::WriteAllText("$activityreport", $txt)
        #Get-ChildItem -File -Recurse | % { $x = get-content -raw -
path $activityreport; $x -replace "`r`n", "`n" | set-content -
path $activityreport }
    }
}

function AnonymizeData
{
    $aprrdir=$args[0]
    $datestampdir=$args[1]
    $path="$aprrdir/$datestampdir"
    $accesslogzipped="access_log.zip"

```

```

$accesslog="access_log.csv"
$accesslogupdated=$accesslog + ".tmp"
$activityreportfile="$datestampdir" + ".html"
$userArray = @()

    expand-Archive -Path "$path/$accesslogzipped" -
DestinationPath $path
    rm $path/$accesslogzipped 2>&1 | out-null
    $accesslogexists=Test-Path "$path/$accesslog"
    if ($accesslogexists) {
        EchoAndLogMessage "Processing access log: $path/$accesslog"
        Get-Content $path/$accesslog | ForEach-Object {
            $elements=[regex]::Split( $_ , ',(?=(?:
[^"]|"["^"]*"*)*$)' )
            $date=$elements[0]
            $time=$elements[1]
            $uri=$elements[2]
            $duration=$elements[3]
            $bytes=$elements[4]
            $ip=$elements[5]
            $user=$elements[6]
            $screen=$elements[7]
            $action=$elements[8]
            $object=$elements[9]
            if ($date -like 'Date') {
                echo "$_" >> $path/$accesslogupdated
            } else {
                if ($user -notlike '-') {
                    LogMessage "Removing instance of User ID: $user
from $path/$accesslog."
                    echo
"$date,$time,$uri,$duration,$bytes,$ip,XXXXX,$screen,$action,$object
" >> $path/$accesslogupdated
                    $userArray += $user
                } else {
                    echo
"$date,$time,$uri,$duration,$bytes,$ip,$user,$screen,$action,$object
" >> $path/$accesslogupdated
                }
            }
        }
        #Get-ChildItem -File -Recurse | % { $x = get-content -raw -
path $path/$accesslogupdated; $x -replace "`r`n","`n" | set-content
-path $path/$accesslogupdated }
        $txt = [io.file]::ReadAllText("$path/$accesslogupdated") -
replace "`r`n","`n"
        [io.file]::WriteAllText("$path/$accesslogupdated", $txt)
        mv -Force $path/$accesslogupdated $path/$accesslog
        Compress-Archive -
Path $path/$accesslog $path/$accesslogzipped
        rm $path/$accesslog 2>&1 | out-null
    }

    EchoAndLogMessage "Processing activity
report: $path/$activityreportfile"

```

```

    $userArray = $userArray | Select-Object -Unique
    foreach ($element in $userArray) {
        ProcessActivityReport "$path/$activityreportfile"
    }
}

function ProcessFiles
{
    # Loop through iteration csv file and parse
    Get-Content $filelist | ForEach-Object {
        $fullpath=$_trim()
        $elements=$fullpath.split('/')
        $aprdir=$elements[0]
        $datestampdir=$elements[1]
        $accesslogfile="access_log.zip"
        $activityreportfile="$datestampdir" + ".html"
        $datestampdirexists=Test-Path "$aprdir/$datestampdir"
        $accesslog="$aprdir/$datestampdir/$accesslogfile"
        $activityreport="$aprdir/$datestampdir/$activityreportfile"

        echo "fullpath: $fullpath" >> $logfile
        echo "aprdir: $aprdir, datestampdir: $datestampdir" >> $logfile
        if (!$datestampdirexists) {
            mkdir "$aprdir/$datestampdir" -ea 0 2>&1 | out-null
            ProcessCommand downloadfile "$accesslog"
            ProcessCommand downloadfile "$activityreport"
            mv "$accesslogfile" "$aprdir/$datestampdir"
            mv "$activityreportfile" "$aprdir/$datestampdir"
            AnonymizeData "$aprdir" "$datestampdir"
            ProcessCommand deletefile "$accesslog"
            ProcessCommand deletefile "$activityreport"
            ProcessCommand uploadfile "$accesslog" "$aprdir/$datestampdir"
            ProcessCommand uploadfile "$activityreport"
        }
        "$aprdir/$datestampdir"
    } else {
        EchoAndLogMessage "Files in directory $aprdir/$datestampdir
were processed earlier. Skipping these files."
    }
}

function callSendMail
{
    $elements=$logfile.split('/')
    $logfilefilename=$elements[3]

    if (${emailtoaddress} -match "@") {
        epmautomate.bat login ${username} ${password} ${url}
        epmautomate.bat uploadFile "$logfile"
        epmautomate.bat sendMail $emailtoaddress "Mask Access Logs and
Activity Reports results" Body="The results of running the Mask Access
Logs and Activity Reports script are attached." Attachments=$logfilefilename
        epmautomate.bat deleteFile "$logfilefilename"
        epmautomate.bat logout
    }
}

```

```
    }  
  }  
  
  Init  
  EchoAndLogMessage "Starting the anonymize data script"  
  RunEpmAutomateCommands  
  EchoAndLogMessage "Anonymize data script completed"  
  EchoAndLogMessage "Refer to logfile: $logfile for details."  
  callSendMail
```

3. Planen Sie `anonymizeData.bat` mit Windows Scheduler. Detaillierte Schritte finden Sie unter [Skriptausführung automatisieren](#)

Zur Ausführung von `anonymizeData.bat` müssen Sie die folgenden Parameterwerte angeben:

- Benutzername eines Serviceadministrators
- Kennwort des Serviceadministrators oder Speicherort, an dem die verschlüsselte Kennwortdatei verfügbar ist
- URL der Serviceumgebung, in der die Zugriffslogs und Aktivitätsberichte maskiert werden sollen
- **Optional:** Die E-Mail-Adresse, an die der Bericht gesendet werden soll. Der Bericht wird nur per E-Mail gesendet, wenn dieser Wert angegeben wurde.

Downloads von Aktivitätsberichten auf einen lokalen Computer automatisieren

Verwenden Sie das Skript in diesem Abschnitt, um das Herunterladen von Aktivitätsberichten aus einer Umgebung auf einen lokalen Computer zu automatisieren.

Verwenden Sie `syncAprReports.bat`, um Aktivitätsberichte herunterzuladen. Sie können die Batchdatei mit Windows Scheduler so planen, dass das Herunterladen von Aktivitätsberichten automatisiert wird. Detaillierte Informationen zu Aktivitätsberichten finden Sie unter *Aktivitätsberichte und Zugriffslogs zum Überwachen der Verwendung verwenden in Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*.

Sie erstellen `syncAprReports.bat` manuell, indem Sie das in der folgenden Prozedur bereitgestellte Skript kopieren und dann die Verbindungsparameter aktualisieren. Dieses Skript prüft die Umgebung und lädt nur die Berichte herunter, die neuer sind als die Berichte, die im Downloadverzeichnis auf dem lokalen Computer verfügbar sind.

 **Hinweis:**

- Dieses Skript kann nur auf einem Windows-Computer ausgeführt werden.
- Der Feedbackaktivitätsbericht, der generiert wird, wenn Benutzer Feedback weiterleiten, wird mit diesem Skript nicht heruntergeladen.
- Wenn das Kennwort, das Sie verwenden, Sonderzeichen enthält, finden Sie unter [Sonderzeichen verarbeiten](#) weitere Informationen.

1. Erstellen Sie eine Batchdatei (.BAT) namens `syncAprReports.bat`, die das folgende Skript enthält, und speichern Sie sie in einem entsprechenden Speicherort, beispielsweise `C:\automate_scripts`.

```
@echo off
title APR
setlocal DisableDelayedExpansion

REM To hardcode the values in the script replace %1, %2, %3, and %4, with
the actual values.
REM Example:
REM set apr_dir="C:\Oracle\apr"
REM set username="serviceAdmin"
REM set password="Ex@mp!e!"
REM set url="https://test-example.stg-pbcs.us1.oraclecloud.com"
set apr_dir=%1
set username=%2
set password=%3
set url=%4
setlocal EnableDelayedExpansion
set epmautomate_dir=%cd%
set lastfile=
set argC=0
for %%x in (*) do Set /A argC+=1
if %argC% neq 0 (
    if %argC% neq 3 (
        if %argC% neq 4 (
            goto :usage
        )
    )
)
goto :login
:usage
echo.
echo Invalid syntax. Please check the parameters.
echo Syntax:
echo 1) syncAprReports.bat APR_FolderPath_on_client username password url
echo      or
echo 2) set the parameters in the file and use below syntax
echo      syncAprReports.bat
goto :end

:login
```

```

setlocal DisableDelayedExpansion
for /f "delims=" %i in ('epmautomate login %username% %password%
%url%') do set result=%i
if "Login successful" neq "%result%" (
    echo Login Failed
    goto :end
)

if not exist %apr_dir% (
    echo.
    echo apr folder does not exist
    GOTO :end
)
cd /D %apr_dir%
for /f "delims=" %D in ('dir /a:d /b /o:-n') do (
    REM AFTER: for /f "delims=" %%D in ('dir /a-d /b /s /o:-n') do (
        set "lastFile=%%~nD"
        goto :next
    )

:next
setlocal EnableDelayedExpansion
echo.
echo Most Recent APR on client is %lastFile%

set "output_cnt=0"
cd /D %epmautomate_dir%
for /F "delims=" %f in ('epmautomate listfiles') do (

    cd /D !apr_dir!
    set "line=%f"
    for /f "tokens=* delims=" %a in ("!line!") do set line=%a
    if "!line:~0,3!" equ "apr" (

        if "!line:~4,8!" neq "Feedback" (

            set isValidFile=false
            if "!line:~-5!" equ ".html" set isValidFile=true
            if "!line:~-5!" equ ".json" set isValidFile=true

            if "!isValidFile!" equ "true" (

                if "%lastFile%" lss "!line:~4,19!" (

                    if "!line:~4,19!" neq "!dirname!" (

                        set apr_dir=!apr_dir:!="!
                        set /a output_cnt+=1
                        set "output[!output_cnt!]=!apr_dir!\!"
line:~4,19!"

                        set "dirname=!line:~4,19!"

                        REM start downloading
                        mkdir "!dirname!"
                        cd /D !dirname!

```

```

        echo downloading !line!
        set "downloadDir=!apr_dir!\!dirname!"

        cd /D %epmautomate_dir%
        for /f "delims=" %%i in ('epmautomate
downloadfile "!line!") do set result1=%%i
        move "!line:~24!" "!downloadDir!" > nul
        echo !result1!
        REM end downloading

    ) else (

        REM start downloading
        cd /D !dirname!
        echo downloading !line!
        set apr_dir=!apr_dir:"=!
        set "downloadDir=!apr_dir!\!dirname!"
        cd /D %epmautomate_dir%
        for /f "delims=" %%i in ('epmautomate
downloadfile "!line!") do set result1=%%i
        move "!line:~24!" "!downloadDir!" > nul
        echo !result1!
        REM end downloading

    )
) else (

    REM TO-DO

)
)
)
)
)
)
)
)

echo.
echo %output_cnt% APR's downloaded
for /L %%n in (1 1 !output_cnt!) DO echo !output[%%n]!
goto :end

:end
cd /D %epmautomate_dir%
endlocal

```

- Ändern Sie `syncAprReports.bat`, um die Werte für die Parameter in der folgenden Tabelle festzulegen. Mit diesen Werten wird auf die Umgebung zugegriffen, um Aktivitätsberichte herunterzuladen.

Tabelle 3-3 In `syncAprReports.bat` einzuschließende Parameterwerte

Parameter	Erwarteter Wert
<code>set apr_dir=%1</code>	Geben Sie ein vorhandenes Verzeichnis an, in das Aktivitätsberichte heruntergeladen werden sollen. Beispiel: <code>set apr_dir="C:\Oracle\apr"</code>

Tabelle 3-3 (Fortsetzung) In `syncAprReports.bat` einzuschließende Parameterwerte

Parameter	Erwarteter Wert
<code>set username=%2</code>	Ein Oracle Enterprise Performance Management Cloud-Benutzername, der für die Anmeldung bei der Umgebung verwendet werden muss, um Aktivitätsberichte herunterzuladen. Beispiel: <code>set username="ServiceAdmin"</code>
<code>set password=%3</code>	Der Name und Speicherort der Datei, in der das verschlüsselte Kennwort des Benutzers gespeichert wird, der mit der Variable <code>username</code> angegeben wird. Sie können auch das Nur-Text-Kennwort des Benutzers angeben (nicht empfohlen). Informationen zum Erstellen einer verschlüsselten Kennwortdatei finden Sie im Abschnitt zum Befehl encrypt . Beispiele: <code>set password="C:\mySecuredir\password.epw"</code> <code>set password="Ex@mp1e1"</code>
<code>set url=%4</code>	Die URL der Umgebung. Beispiel: <code>set url="https://test-example.stg-pbcs.us1.oraclecloud.com"</code>

3. Planen Sie `syncAprReports.bat` mit Windows Scheduler. Detaillierte Schritte finden Sie unter [Skriptausführung automatisieren](#)

Zugriffslogs aus einer Umgebung herunterladen

Verwenden Sie das Skript in diesem Abschnitt, um den Prozess zum Herunterladen von Zugriffslogs aus einer Umgebung auf einen lokalen Computer zu automatisieren.

Sie können `syncAccessLog.bat` mit Windows Scheduler so planen, dass das Herunterladen der Logdateien automatisiert wird. Prozeduren zum Herunterladen von Zugriffslogs mit dem Anwendungsmanagement finden Sie unter *Aktivitätsberichte und Zugriffslogs anzeigen und herunterladen* in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*.

Das folgende Skript prüft die Umgebung und lädt nur die Logdateien herunter, die neuer sind als die Berichte, die im Downloadverzeichnis auf dem lokalen Computer verfügbar sind. Es handelt sich hierbei um ein Windows-Skript; Sie können ein ähnliches Shell-Skript für Linux-/UNIX-Umgebungen erstellen.

1. Erstellen Sie eine Batchdatei (BAT) mit dem Namen `syncAccessLog.bat`, die das folgende Skript enthält, und speichern Sie sie an einem entsprechenden Speicherort, z.B. `C:\automate_scripts`.

Hinweis:

Wenn Ihr Kennwort Sonderzeichen enthält, finden Sie unter [Sonderzeichen verarbeiten](#) weitere Informationen.

```
@echo off
title APR
```

```

setlocal DisableDelayedExpansion

REM To hardcode the values in the script replace %1, %2, %3, and %4 with
the actual values.
REM Example:
REM set apr_dir="C:\Oracle\apr"
REM set username="serviceAdmin"
REM set password="C:\mySecuredir\password.epw"
REM set url="https://test-cloudpln.pbcs.us1.oraclecloud.com"
set apr_dir=%1
set username=%2
set password=%3
set url=%4

setlocal EnableDelayedExpansion
set epmautomate_dir=%cd%
set lastfile=
REM if [%1]==[] goto :usage
REM if [%2]==[] goto :usage
REM if [%3]==[] goto :usage

set argC=0
for %%x in (*) do Set /A argC+=1
if %argC% neq 0 (
    if %argC% neq 3 (
        if %argC% neq 4 (
            goto :usage
        )
    )
)
goto :login

:usage
echo.
echo Invalid syntax. Please check the parameters.
echo Syntax:
echo 1) syncAccessLog.bat APR_FolderPath_on_client username password url
echo     or
echo 2) set the parameters in the file and use below syntax
echo syncAccessLog.bat
goto :end

:login
setlocal DisableDelayedExpansion
REM for /f "delims=" %%i in ('epmautomate login %2 %3 %4') do set result=%%i
for /f "delims=" %%i in ('epmautomate login %username% %password% %url%')
do set result=%%i

if not exist %apr_dir% (
    echo.
    echo apr folder does not exist
    GOTO :end
)
cd /D %apr_dir%

```

```

for /f "delims=" %%D in ('dir /a:d /b /o:-n') do (
REM AFTER: for /f "delims=" %%D in ('dir /a-d /b /s /o:-n') do (
    set "lastFile=%%~nD"
    goto :next
)

:next
setlocal EnableDelayedExpansion
echo.
echo Most Recent Access Log on client is %lastFile%

set "output_cnt=0"
cd /D %epmautomate_dir%
for /F "delims=" %%f in ('epmautomate listfiles') do (

    cd /D !apr_dir!
    set "line=%%f"
    for /f "tokens=* delims=" %%a in ("!line!") do set line=%%a
    if "!line:~0,3!" equ "apr" (
        if "!line:~-4!" equ ".zip" (
            if "%lastFile%" lss "!line:~4,19!" (
                if "!line:~4,19!" neq "!dirname!" (
                    set apr_dir=!apr_dir:"=!
                    set /a output_cnt+=1
                    set "output[!output_cnt!]=!apr_dir!\!line:~4,19!"
                    set "dirname=!line:~4,19!"

                    REM start downloading
                    mkdir "!dirname!"
                    cd /D !dirname!
                    echo downloading !line!
                    set "downloadDir=!apr_dir!\!dirname!"
                    cd /D %epmautomate_dir%
                    for /f "delims=" %%i in ('epmautomate downloadfile "!"
line!"') do set result1=%%i
                    move "!line:~24!" "!downloadDir!" > nul
                    echo !result1!
                    REM end downloading

                ) else (
                    REM start downloading
                    cd /D !dirname!
                    echo downloading !line!
                    set apr_dir=!apr_dir:"=!
                    set "downloadDir=!apr_dir!\!dirname!"
                    cd /D %epmautomate_dir%
                    for /f "delims=" %%i in ('epmautomate downloadfile "!"
line!"') do set result1=%%i
                    move "!line:~24!" "!downloadDir!" > nul
                    echo !result1!
                    REM end downloading

                )
            ) else (
                REM TO-DO
            )
        )
    )
)

```

```

    )
  )
)

echo.
echo %output_cnt% access logs downloaded
for /L %%n in (1 1 !output_cnt!) DO echo !output[%%n]!
goto :end

:end
cd /D %epmautomate_dir%
endlocal

```

2. Ändern Sie `syncAccessLog.bat`, um die Werte für die Parameter in der folgenden Tabelle festzulegen. Mit diesen Werten wird auf die Umgebung zugegriffen, um Zugriffslogs herunterzuladen.

Tabelle 3-4 In `syncAccessLog.bat` einzuschließende Variablenwerte

Variable	Erwarteter Wert
<code>set apr_dir=%1</code>	Geben Sie das Verzeichnis an, in das Zugriffslogs heruntergeladen werden sollen. Beispiel: <code>set apr_dir="C:\Oracle\apr"</code>
<code>set username=%2</code>	Ein Oracle Enterprise Performance Management Cloud-Benutzername, der für die Anmeldung bei der Umgebung verwendet werden muss, um Zugriffslogs herunterzuladen. Beispiel: <code>set username="ServiceAdmin"</code>
<code>set password=%3</code>	Der Name und Speicherort der Datei, in der das verschlüsselte Kennwort des Benutzers gespeichert wird, der mit der Variable <code>username</code> angegeben wird. Sie können auch das Nur-Text-Kennwort des Benutzers angeben (nicht empfohlen). Informationen zum Erstellen einer verschlüsselten Kennwortdatei finden Sie im Abschnitt zum Befehl encrypt . Beispiele: <code>set password="C:\mySecuredir\password.epw"</code> <code>set password="P@ssword1"</code>
<code>set url=%4</code>	Die URL der Umgebung. Beispiel: <code>set url="https://test-cloudpln.pbc.us1.oraclecloud.com"</code>

3. Planen Sie `syncAccessLog.bat` mit Windows Scheduler. Detaillierte Schritte finden Sie unter [Skriptausführung automatisieren](#)

Klonen von Umgebungen automatisieren

Verwenden Sie das Skript in diesem Abschnitt, um das Klonen von Umgebungen zu automatisieren.

Um eine Umgebung zu klonen, erstellen Sie eine Batchdatei (`.bat`) oder eine Shell-Datei (`.sh`) mit einem ähnlichen Skript wie das folgende. Folgende Beispielskripte umfassen die folgenden Aktivitäten::

- Melden Sie sich bei der Quellumgebung an.

- Verwenden Sie den `Artifact Snapshot` (den während der letzten täglichen Wartung der Quellumgebung erstellten Snapshot) oder einen anderen in der Quellumgebung verfügbaren Snapshot, um die Zielumgebung als Klon der Quelle zu erstellen.
- Erstellen Sie optional Benutzer mit den zugehörigen vordefinierten und anwendungsbezogenen Rollenzuweisungen, die mit denen in der Quellumgebung übereinstimmen.
- Ändern Sie optional die Startzeit der täglichen Wartung so, dass sie mit der Startzeit der Quellumgebung übereinstimmt.
- Melden Sie sich ab.

Ausführliche Informationen zum Klonprozess finden Sie unter "[EPM Cloud-Umgebungen klonen](#)" in der Dokumentation *Migration für Oracle Enterprise Performance Management Cloud verwalten*.

Informationen zum Planen von Skripten mit der Windows-Aufgabenplanung finden Sie unter [Skriptausführung automatisieren](#).

Windows

1. Erstellen Sie eine Batchdatei (BAT) mit dem Namen `cloneEnvironment.bat`, die das folgende Skript enthält, und speichern Sie sie an einem geeigneten Speicherort, z.B. `C:\automate_scripts`.

```
@echo off
set paramRequiredMessage=Syntax: cloneEnvironment.bat "SOURCE
USERNAME" "SOURCE PASSWORD FILE" "SOURCE URL" "TARGET USERNAME"
"TARGET PASSWORD FILE" "TARGET URL"

set usersandpredefinedroles="false"
set snapshotname="Artifact Snapshot"
set dailymaintenancestarttime="true"
set dirpath=%~dp0
cd %dirpath:~0,-1%

if "%~1" == "" (
    echo Source User Name is missing.
    echo %paramRequiredMessage%
    exit /b 1
)
if "%~2" == "" (
    echo Source Password File is missing.
    echo %paramRequiredMessage%
    exit /b 1
)
if "%~3" == "" (
    echo Source URL is missing.
    echo %paramRequiredMessage%
    exit /b 1
)
if "%~4" == "" (
    echo Target User Name is missing.
    echo %paramRequiredMessage%
    exit /b 1
)
```

```

)
if "%~5" == "" (
    echo Target Password File is missing.
    echo %paramRequiredMessage%
    exit /b 1
)
if "%~6" == "" (
    echo Target URL is missing.
    echo %paramRequiredMessage%
    exit /b 1
)

```

```

PowerShell.exe -File cloneEnvironment.ps1 %~1 %~2 %~3 %~4 %~5 %~6
%usersandpredefinedroles% %snapshotname% %dailymaintenancestarttime%

```

2. Ändern Sie `cloneEnvironment.bat`, um die Werte für die folgenden Parameter festzulegen:

Tabelle 3-5 In "`cloneEnvironment.bat`" festzulegende Parameter:

Parameter	Beschreibung
<code>usersandpredefinedroles</code>	Setzen Sie den Wert dieses Parameters auf <code>true</code> , um Benutzer und die zugehörigen vordefinierten und anwendungsbezogenen Rollenzuweisungen zu klonen. Um Benutzer und Rollenzuweisungen zu klonen, muss der Benutzer, der das Skript ausführt, in der Zielumgebung über die Rollen <code>Serviceadministrator</code> und <code>Identitätsdomainadministrator</code> verfügen.
<code>snapshotname</code>	Name des Snapshots in der Quellumgebung, die zum Klonen verwendet werden soll:
<code>dailymaintenancestarttime</code>	Setzen Sie den Wert dieses Parameters auf <code>true</code> , um die Startzeit der täglichen Wartung der geklonten Umgebung in die Startzeit der Quellumgebung zu ändern. Setzen Sie diesen Wert auf <code>false</code> , um die aktuelle Startzeit der täglichen Wartung der geklonten Umgebung beizubehalten.

3. Erstellen Sie ein PowerShell-Skript mit dem Namen `cloneEnvironment.ps1`, das das folgende Skript enthält, und speichern Sie es im selben Verzeichnis, in dem Sie `cloneEnvironment.bat` gespeichert haben, z.B. `C:\automate_scripts`.

```

# Clone Environment script

$source_username=$args[0]
$source_password=$args[1]
$source_url=$args[2]
$target_username=$args[3]
$target_password=$args[4]
$target_url=$args[5]
$usersandpredefinedroles=$args[6]
$snapshotname=$args[7]
$dailymaintenancestarttime=$args[8]

```

```
epmautomate.bat login "${source_username}" "${source_password}" "$
{source_url}"
epmautomate.bat cloneEnvironment "${target_username}" "$
{target_password}" "${target_url}" UsersAndPreDefinedRoles="$
{usersandpredefinedroles}" SnapshotName="${snapshotname}"
DailyMaintenanceStartTime="${dailymaintenancestarttime}"
epmautomate.bat logout
```

4. Führen Sie cloneEnvironment.bat mit dem folgenden Befehl aus:

```
cloneEnvironment.bat "SOURCE USERNAME" "SOURCE PASSWORD FILE"
"SOURCE URL" "TARGET USERNAME" "TARGET PASSWORD FILE" "TARGET URL"
```

Beispiel:

```
cloneEnvironment.bat jdoe@example.com
C:\mySecuredir\example_pwd.epw https://
source_example.oraclecloud.com jdoe@example.com
C:\mySecuredir\example_pwd2.epw https://
target_example.oraclecloud.com.
```

Linux

1. Erstellen Sie ein Shell-Skript mit dem Namen cloneEnvironment.sh, das das folgende Skript enthält, und speichern Sie es an einem geeigneten Speicherort.

```
#!/bin/bash

# Update the following parameters
# -----
epmautomatescript=/home/user1/epmautomate/bin/epmautomate.sh
javahome=/home/user1/jdk1.8.0_191/
usersandpredefinedroles="false"
snapshotname="Artifact Snapshot"
dailymaintenancestarttime="true"
# -----

source_username="$1"
source_password="$2"
source_url="$3"
target_username="$4"
target_password="$5"
target_url="$6"

export JAVA_HOME=${javahome}

if [ "$#" -ne 6 ]; then
    echo "Usage: ./cloneEnvironment.sh <SOURCE USERNAME> <SOURCE
PASSWORD FILE> <SOURCE URL> <TARGET USERNAME> <TARGET PASSWORD
FILE> <TARGET URL>"
    exit 1
fi

${epmautomatescript} login "${source_username}" "$
```

```
{source_password}" "${source_url}"
${epmautomatescript} cloneEnvironment "${target_username}" "${
target_password}" "${target_url}" UsersAndPreDefinedRoles="${
usersandpredefinedroles}" SnapshotName="${snapshotname}"
DailyMaintenanceStartTime="${dailymaintenancestarttime}"
${epmautomatescript} logout
```

- Ändern Sie `cloneEnvironment.sh`, um die Werte für die folgenden Parameter festzulegen:

Tabelle 3-6 In "`cloneEnvironment.sh`" festzulegende Parameter

Parameter	Beschreibung
<code>epmautomatescript</code>	Der absolute Pfad der ausführbaren Datei für EPM Automate (<code>epmautomate.sh</code>).
<code>javahome</code>	Verzeichnis <code>JAVA_HOME</code> .
<code>usersandpredefinedroles</code>	Setzen Sie den Wert dieses Parameters auf <code>true</code> , um Benutzer und die zugehörigen vordefinierten und anwendungsbezogenen Rollenzuweisungen zu klonen. Um Benutzer und Rollenzuweisungen zu klonen, muss der Benutzer, der das Skript ausführt, in der Zielumgebung über die Rollen <code>Serviceadministrator</code> und <code>Identitätsdomainadministrator</code> verfügen.
<code>snapshotname</code>	Name des Snapshots in der Quellumgebung, die zum Klonen verwendet werden soll:
<code>dailymaintenancestarttime</code>	Setzen Sie den Wert dieses Parameters auf <code>true</code> , um die Startzeit der täglichen Wartung der geklonten Umgebung in die Startzeit der Quellumgebung zu ändern. Setzen Sie diesen Wert auf <code>false</code> , um die aktuelle Startzeit der täglichen Wartung der geklonten Umgebung beizubehalten.

- Führen Sie `cloneEnvironment.sh` aus.

```
./cloneEnvironment.sh "SOURCE USERNAME" "SOURCE PASSWORD FILE" "SOURCE
URL" "TARGET USERNAME" "TARGET PASSWORD FILE" "TARGET URL"
```

Beispiel:

```
./cloneEnvironment.sh jdoe@example.com ./home/secure/example_pwd.epw
https://source_example.oraclecloud.com jdoe@example.com ./home/secure/
example_pwd.epw2 https://target_example.oraclecloud.com.
```

Unnötige Dateien aus einer Umgebung entfernen

Verwenden Sie diese Skripte, um unnötige Dateien aus einer Umgebung zu entfernen.

Diese Skripte führen die folgenden Schritte aus:

- Anmeldung bei der Umgebung
- Auflistung der Dateien und Snapshots in der Umgebung

- Löschung der in der Datei `input.properties` angegebenen Dateien
- Abmeldung

Windows-Beispielskript

Erstellen Sie eine Datei namens `removeUnnecessaryFiles.ps1`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./
input.properties -raw)
$username="$($inputproperties.username) "
$passwordfile="$($inputproperties.passwordfile) "
$serviceURL="$($inputproperties.serviceURL) "
$file1="$($inputproperties.file1) "
$file2="$($inputproperties.file2) "

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate listfiles
epmautomate deletefile ${file1}
epmautomate deletefile ${file2}
epmautomate logout
```

Linux-/UNIX-Beispielskript

Erstellen Sie eine Datei namens `removeUnnecessaryFiles.sh`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "${
serviceURL}"
${epmautomatescript} listfiles
${epmautomatescript} deletefile "${file1}"
${epmautomatescript} deletefile "${file2}"
${epmautomatescript} logout
```

Datei "input.properties" erstellen

Um die `removeUnnecessaryFiles`-Skripte auszuführen, erstellen Sie die Datei `input.properties`, und aktualisieren Sie sie mit den Informationen für Ihre Umgebung. Speichern Sie die Datei im selben Verzeichnis, in dem die Datei `removeUnnecessaryFiles.ps1` oder `removeUnnecessaryFiles.sh` gespeichert ist.

Windows

```
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
file1=FILE_NAME
file2=FILE_NAME
```

Linux/UNIX

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
file1=FILE_NAME
file2=FILE_NAME
```

Tabelle 3-7 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis JAVA_HOME. Nur für Linux/UNIX.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (epmautomate.sh). Nur für Linux/UNIX.
username	Benutzername eines Serviceadministrators, der auch über die Rolle Identitätsdomainadministrator verfügt.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
serviceURL	URL der Umgebung, aus der Sie den Snapshot generieren möchten.
file1 und file2	Name einer Datei oder eines Snapshots, die oder der aus der Umgebung gelöscht werden soll. Wenn die Datei sich nicht in der Outbox befindet, geben Sie den Pfad und den Namen der Datei an.

Skripte ausführen

1. Erstellen Sie die Datei `removeUnnecessaryFiles.ps1` oder `removeUnnecessaryFiles.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
2. Erstellen Sie die Datei `input.properties`, und speichern Sie sie im selben Verzeichnis, in dem sich das Skript `removeUnnecessaryFiles` befindet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Informationen hierzu finden Sie unter [Datei "input.properties" erstellen](#).
Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.
3. Starten Sie das Skript.
 - **Windows PowerShell:** Führen Sie die Datei `removeUnnecessaryFiles.ps1` aus.
 - **Linux/UNIX:** Führen Sie die Datei `./removeUnnecessaryFiles.sh` aus.

Dateien aus einer Umgebung suchen und herunterladen

Verwenden Sie das Beispielskript in diesem Abschnitt, um den Prozess zum Herunterladen von Dateien aus einer Oracle Enterprise Performance Management Cloud-Umgebung zu automatisieren. Verwenden Sie dabei eine Textzeichenfolge als Platzhalter.

Mit dem folgenden Skript können Sie die Zeichenfolge, die Sie als Wert des Parameters `FILENAME` angeben, mit den Dateinamen abgleichen, die mit dem Befehl `listfiles` angezeigt

werden. Anschließend können Sie die Dateien, die mit der Zeichenfolge übereinstimmen, automatisch herunterladen.

Stellen Sie sicher, dass Sie dem Parameter `FILENAME` die entsprechende Suchzeichenfolge zuweisen. Beispiel: `FILENAME="Scheduler Output/epm"` gleicht die Zeichenfolge `Scheduler Output/epm` mit den Dateinamen in der Befehlsausgabe `listfiles` in Ihrer Umgebung ab, um die Dateien für den Download zu bestimmen.

Die Eingabeparameter für die Ausführung dieses Skripts sind `username`, `password` oder `password_file` und `service_url`.



Hinweis:

Wenn das Kennwort Sonderzeichen enthält, finden Sie unter [Sonderzeichen verarbeiten](#) weitere Informationen.

Windows

```
@echo off
    setlocal EnableExtensions EnableDelayedExpansion
    set USERNAME="username"
    set PASSWORD="password"
    set URL="url"

    call epmautomate login %USERNAME% %PASSWORD% %URL%
    set FILENAME="Scheduler Output/epm"
    for /f "tokens=*" %%i in ('epmautomate listfiles ^| findstr /b /r /
c:"^ *%FILENAME%" ') do (
        call epmautomate downloadfile "%%i"
    )
    call epmautomate logout
endlocal
```

Linux/UNIX

```
#!/bin/sh
    USERNAME="username"
    PASSWORD="password"
    URL="url"

    ./epmautomate.sh login $USERNAME $PASSWORD $URL
    FILENAME='Scheduler Output/epm'
    #echo $FILENAME
    ./epmautomate.sh listfiles | grep "^ $FILENAME" | while read -r line ;
do
    echo "Processing $line"
    ./epmautomate.sh downloadfile "$line"
done
./epmautomate.sh logout
```

Alte EPM Cloud-Umgebungen für Audits neu erstellen

Verwenden Sie das Skript in diesem Abschnitt, um eine Selfservicelösung zu erstellen, mit der eine aktuelle Bibliothek von Snapshots für Ihre Oracle Enterprise Performance Management Cloud-Umgebung verwaltet werden kann. Sie benötigen eine Umgebung, die speziell für die Aktualisierung und Wartung einer Bibliothek mit aktuellen Snapshots eingerichtet ist.

EPM Cloud unterstützt die Snapshot-Kompatibilität nur für einen monatlichen Zyklus. Sie können Wartungs-Snapshots von der Testumgebung in die Produktionsumgebung migrieren und umgekehrt. Die Auditing-Anforderungen einiger Kunden können jedoch die Wiederherstellung von Snapshots aus mehreren Jahren in der neuesten Umgebung und den Zugriff auf die Anwendung in kurzer Zeit erforderlich machen.

Dieses Skript sollte einmal im Monat ausgeführt werden, um die verfügbaren Snapshots zu konvertieren und sie mit der neuesten EPM Cloud-Patchebene kompatibel zu machen. Oracle empfiehlt, das Skript nach dem dritten Freitag im Monat auszuführen, um sicherzustellen, dass alle Probleme in der Produktionsumgebung behoben sind.

Hinweis:

Sie können dieses Skript nicht verwenden, um Snapshots für Narrative Reporting, Account Reconciliation und Oracle Enterprise Data Management Cloud zu aktualisieren.

Funktionsweise des Skripts

Für jeden vom Kunden gespeicherten Snapshot führt das Upgradeskript die folgenden Aufgaben mit EPM Automate aus:

1. Mit den Informationen in der Datei `input.properties` wird die Anmeldung in einer Umgebung durchgeführt.
2. Verwendet den Befehl `recreate`, um die Umgebung wiederherzustellen.
3. Importiert den Snapshot in die Umgebung.
4. Führt eine tägliche Wartung der Umgebung aus, was dazu führt, dass der Snapshot in das Format konvertiert wird, das mit der aktuellen EPM Cloud-Patchebene kompatibel ist.
5. Lädt `Artifact Snapshot` (Wartungs-Snapshot) in einen Ordner herunter. Wenn Sie eine 18.05-Umgebung durch das Hochladen von Snapshots aus `Snapshots/18.05` erstellt haben, wird `Artifact Snapshot` in `Snapshots/18.06` heruntergeladen.
6. Sendet die Ergebnisse der Neuerstellung alter Umgebungen per E-Mail an eine E-Mail-Adresse (falls angegeben).

Skripte ausführen

1. Erstellen Sie die Datei `input.properties`, und aktualisieren Sie sie mit Informationen für Ihre Umgebung. Speichern Sie die Datei in einem lokalen Verzeichnis. Hierbei handelt es sich um das Verzeichnis, das in dieser Erörterung als `parentsnapshotdirectory` referenziert wird. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem.

Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um Skripte ausführen zu können.

- Erstellen Sie das Skript `upgradeSnapshots.ps1` (Windows PowerShell) oder das Skript `upgradeSnapshots.sh` (Linux/UNIX), und speichern Sie es im Verzeichnis `parentsnapshotdirectory`, in dem sich `input.properties` befindet.
- Erstellen Sie unter `parentsnapshotdirectory` ein Unterverzeichnis, z.B. `snapshots`.
- Erstellen Sie innerhalb des Verzeichnisses, das Sie im vorherigen Schritt erstellt haben (`snapshots`) ein Unterverzeichnis für den monatlichen Snapshot, den Sie konvertieren möchten, um ihn mit der aktuellen EPM Cloud-Patchebene kompatibel zu machen. Benennen Sie das Verzeichnis im Format `YYY.MM`, z.B. `18.05` für das Verzeichnis zum Speichern der Snapshots vom Mai 2018.
- Kopieren Sie Snapshots in das entsprechende Unterverzeichnis. Beispiel: Kopieren Sie die Snapshots für Mai 2018 in `snapshots/18.05`.
- Starten Sie das Skript.
 - Linux/UNIX: Führen Sie das Skript `./upgradeSnapshots.sh` aus.
 - Windows PowerShell: Führen Sie das Skript `upgradeSnapshots.ps1` aus.

Windows

Erstellen Sie das Skript `input.properties` und `upgradeSnapshots.ps1` durch Kopieren der Skripte in diesem Abschnitt.

`input.properties` erstellen

```
username=exampleAdmin
userpassword=examplePassword
serviceurl=exapleURL
proxyserverusername=proxyServerUserName
proxyserverpassword=proxyPassword
proxyserverdomain=proxyDoamin
parentsnapshotdirectory=C:/some_directory/snapshots
emailtoaddress=exampleAdmin@oracle.com
```

`input.properties` aktualisieren



Hinweis:

Wenn authentication at proxy server für Ihre Windows-Netzwerkumgebung nicht aktiviert ist, entfernen Sie die Eigenschaften `proxyserverusername`, `proxyserverpassword` und `proxyserverdomain` aus der Datei `input.properties`.

Tabelle 3-8 Parameter für "input.properties"

Parameter	Beschreibung
username	Benutzername eines Serviceadministrators.
userpassword	Kennwort des Serviceadministrators.
serviceurl	URL der Umgebung, die für diese Aktivität verwendet wird.
proxyserverusername	Der Benutzername zum Authentifizieren einer sicheren Session auf dem Proxyserver, der den Zugriff auf das Internet steuert.
proxyserverpassword	Das Kennwort zur Authentifizierung des Benutzers beim Proxyserver.
proxyserverdomain	Der Name der Domain, die für den Proxyserver definiert ist.
parentsnapshotdirectory	Absoluter Pfad des Verzeichnisses, das als übergeordnetes Verzeichnis des Verzeichnisses verwendet werden soll, in dem die zu verarbeitenden Snapshots gespeichert werden. Verwenden Sie Schrägstriche (/) als Verzeichnistrennzeichen.
emailtoaddress	Optional, die E-Mail-Adresse, an die Ergebnisse der Neuerstellung alter Umgebungen gesendet werden sollen. Die Ergebnisse werden nur per E-Mail gesendet, wenn dieser Wert angegeben wurde. Beispiel: john.doe@example.com

 **Hinweis:**

Wenn Ihr Kennwort Sonderzeichen enthält, finden Sie unter [Sonderzeichen verarbeiten](#) weitere Informationen.

upgradeSnapshots.ps1 erstellen

Verwenden Sie dieses Beispielskript zum Erstellen von `upgradeSnapshots.ps1`.

```
# Script for recreating an old EPM Cloud environment

# read in key/value pairs from input.properties file
$inputproperties=ConvertFrom-StringData(Get-Content ./input.properties -raw)

# Global variables
$parentsnapshotdirectory="$($inputproperties.parentsnapshotdirectory) "
$username="$($inputproperties.username) "
$userpassword="$($inputproperties.userpassword) "
$serviceurl="$($inputproperties.serviceurl) "
$proxyserverusername="$($inputproperties.proxyserverusername) "
$proxyserverpassword="$($inputproperties.proxyserverpassword) "
$proxyserverdomain="$($inputproperties.proxyserverdomain) "
$emailtoaddress="$($inputproperties.emailtoaddress) "
$operationmessage="EPM Automate operation:"
$operationfailuremessage="EPM Automate operation failed:"
$operationsuccessmessage="EPM Automate operation completed successfully:"
$epmautomatescript="epmautomate.bat"
```

```

$workingdir="$pwd"
$logdir="$workingdir/logs/"
$logfile="$logdir/epmautomate-upgradesnapshots.log"

function LogMessage
{
    $message=$args[0]
    $_mydate=$(get-date -f dd_MM_yy_HH_mm_ss)

    echo "[$_mydate] $message" >> $logfile
}

function LogAndEchoMessage
{
    $message=$args[0]
    $_mydate=$(get-date -f dd_MM_yy_HH_mm_ss)

    echo "[$_mydate] $message" | Tee-Object -Append -FilePath $logfile
}

function LogOutput
{
    $_mydate=$(get-date -f dd_MM_yy_HH_mm_ss)
    $op=$args[0]
    $opoutput=$args[1]
    $returncode=$args[2]

    #If error
    if ($returncode -ne 0) {
        $failmessage="[$_mydate] $operationfailuremessage $op"
        LogMessage $failmessage
        LogMessage $opoutput
        LogMessage "return code: $returncode"
    } else {
        $successmessage="[$_mydate] $operationsuccessmessage $op"
        LogMessage $successmessage
        LogMessage $opoutput
        LogMessage "return code: $returncode"
    }
}

function ExecuteCommand
{
    $op=$args[0]
    $epmautomatecall="$epmautomatescript $op"
    $date=$(get-date -f dd_MM_yy_HH_mm_ss)

    LogMessage "$operationmessage $epmautomatecall"
    $operationoutput=iex "& $epmautomatecall" >> $logfile 2>&1
    LogOutput $op $operationoutput $LastExitCode
}

function ProcessCommand
{
    $command=$args[0]

```

```

$date=$(get-date -f dd_MM_yy_HH_mm_ss)

if (!(([string]::IsNullOrEmpty($command)) {
    if (!$command.StartsWith("#")) {
        ExecuteCommand $command
    }
})
}

function Init
{
    $logdirexists=Test-Path $logdir
    if (!$logdirexists) {
        mkdir $logdir 2>&1 | out-null
    }

    # removing existing epmautomate debug logs
    rm *.*.log

    $logfileexists=Test-Path $logfile
    # remove existing log file
    if ($logfileexists) {
        rm $logfile
    }
}

function GetNextDate
{
    $latestyearmonth=$args[0]
    LogMessage "latest year.month: $latestyearmonth"
    $latestyear,$latestmonth=$latestyearmonth.split('\. ')
    LogMessage "latest year: $latestyear"
    LogMessage "latest month: $latestmonth"
    $intlatestyear=[int]$latestyear
    $intlatestmonth=[int]$latestmonth

    if ($intlatestmonth -eq 12) {
        $intnextmonth=1
        $intnextyear=$intlatestyear+1
    } else {
        $intnextmonth=$intlatestmonth+1
        $intnextyear=$intlatestyear
    }

    $nextyear="{0:D2}" -f $intnextyear
    $nextmonth="{0:D2}" -f $intnextmonth

    echo "$nextyear.$nextmonth"
}

function ProcessSnapshot
{
    $snapshotfile=$args[0]
    LogMessage "snapshotfile: $snapshotfile"
    $nextdate=$args[1]

```



```

    LogMessage "nextdate: $nextdate"
    $snapshotfilename=$snapshotfile.split('/')[-1]
    LogMessage "snapshotfilename: $snapshotfilename"
    $snapshotname=$snapshotfilename.split('.')[0]
    LogMessage "snapshotname: $snapshotname"

    ProcessCommand
"login $username $userpassword $serviceurl $proxyserverusername $proxys
erverpassword $proxyserverdomain"
    ProcessCommand "recreate -f"
    ProcessCommand "uploadfile $snapshotfile"
    ProcessCommand "importsnapshot $snapshotname"
    ProcessCommand "runDailyMaintenance skipNext=true -f"
    ProcessCommand "downloadfile 'Artifact Snapshot'"
    ProcessCommand "deletefile $snapshotname"
    ProcessCommand "logout"

    $nextdatedirexists=Test-Path $parentsnapshotdirectory/$nextdate
    if (!$nextdatedirexists) {
        mkdir $parentsnapshotdirectory/$nextdate 2>&1 | out-null
    }

    LogMessage "Renaming 'Artifact Snapshot.zip' to $snapshotname.zip
and moving to $parentsnapshotdirectory/$nextdate"
    mv $workingdir/Artifact
Snapshot.zip' $workingdir/$snapshotname.zip >> $logfile 2>&1

mv $workingdir/$snapshotname.zip $parentsnapshotdirectory/$nextdate
>> $logfile 2>&1
}

function callSendMail
{
    $logfile=$logfile -replace "\\\"", "/"
    $elements=$logfile.split('/')
    $logfile=$elements[-1]

    if (${emailtoaddress} -match "@") {
        epmautomate.bat login ${username} ${userpassword} ${serviceurl}
        epmautomate.bat uploadFile "$logfile"
        epmautomate.bat sendMail $emailtoaddress "Recreating An Old
EPM Cloud Environment results" Body="The results of recreating an old
EPM Cloud Environment are attached." Attachments=$logfile
        epmautomate.bat deleteFile "$logfile"
        epmautomate.bat logout
    }
}

#----- main body of processing
date
Init
LogAndEchoMessage "Starting upgrade snapshots processing"
$snapshotdirs=@(Get-ChildItem -Directory "$parentsnapshotdirectory" -
name)
LogMessage "snapshot directories: $snapshotdirs"

```

```

$latestreleasedate=$snapshotdirs[-1]
LogMessage "latest release date: $latestreleasedate"
$latestreleasesnapshotdir="$parentsnapshotdirectory/$latestreleasedate"
LogMessage "latest release snapshot dir: $latestreleasesnapshotdir"
$nextdate=$(GetNextDate "$latestreleasedate")
$snapshotfiles=@(Get-ChildItem -File "$latestreleasesnapshotdir")
if ($snapshotfiles.length -eq 0) {
    LogAndEchoMessage "No snapshot files found in
directory $latestreleasesnapshotdir. Exiting script."
    exit
}
foreach ($snapshotfile in $snapshotfiles) {
    LogAndEchoMessage "Processing snapshotfile: $snapshotfile"
    ProcessSnapshot $latestreleasesnapshotdir/$snapshotfile $nextdate
}
LogAndEchoMessage "Upgrade snapshots processing completed"
date
callSendMail

```

Linux/UNIX

Erstellen Sie `upgradeSnapshots.sh` und `input.properties` durch Kopieren der folgenden Skripte.

`input.properties` für Linux/UNIX erstellen



Hinweis:

Wenn Ihr Netzwerk nicht für die Verwendung eines Proxyservers für den Internetzugriff konfiguriert ist, entfernen Sie die Eigenschaften `proxyserverusername`, `proxyserverpassword` und `proxyserverdomain` aus der Datei `input.properties`.

```

username=exampleAdmin
userpassword=examplePassword
serviceurl=exampleURL
proxyserverusername=
proxyserverpassword=
proxyserverdomain=
jdkdir=/home/user1/jdk160_35
epmautomatescript=/home/exampleAdmin/epmautomate/bin/epmautomate.sh
parentsnapshotdirectory=/home/exampleAdmin/some_directory/snapshots
emailtoaddress=exampleAdmin@oracle.com

```

`input.properties` aktualisieren

Tabelle 3-9 Parameter für "`input.properties`"

Parameter	Beschreibung
<code>username</code>	Benutzername eines Serviceadministrators.

Tabelle 3-9 (Fortsetzung) Parameter für "input.properties"

Parameter	Beschreibung
userpassword	Kennwort des Serviceadministrators.
serviceurl	URL der Umgebung, die für diese Aktivität verwendet wird.
proxyserverusername	Der Benutzername zum Authentifizieren einer sicheren Session auf dem Proxyserver, der den Zugriff auf das Internet steuert.
proxyserverpassword	Das Kennwort zur Authentifizierung des Benutzers beim Proxyserver.
proxyserverdomain	Der Name der Domain, die für den Proxyserver definiert ist.
jdkdir	Verzeichnis JAVA_HOME.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (epmautomate.sh).
parentsnapshotdirectory	Absoluter Pfad des Verzeichnisses, das als übergeordnetes Verzeichnis des Verzeichnisses verwendet werden soll, in dem der zu verarbeitende Snapshot gespeichert wird.
emailtoaddress	Optional, die E-Mail-Adresse, an die Ergebnisse der Neuerstellung alter Umgebungen gesendet werden sollen.



Hinweis:

Wenn Ihr Kennwort Sonderzeichen enthält, finden Sie unter [Sonderzeichen verarbeiten](#) weitere Informationen.

upgradeSnapshots.sh erstellen

Verwenden Sie dieses Beispielskript zum Erstellen von `upgradeSnapshots.sh`.

```
#!/bin/sh

. ./input.properties
workingdir=$(pwd)
logdir="${workingdir}/logs"
logfile=epmautomate-upgradesnapshots.log
operationmessage="EPM Automate operation:"
operationfailuremessage="EPM Automate operation failed:"
operationsuccessmessage="EPM Automate operation completed successfully:"
logdebugmessages=true

if [ ! -d ${jdkdir} ]
then
    echo "Could not locate JDK/JRE. Please set value for "jdkdir"
property in input.properties file to a valid JDK/JRE location."
    exit
fi

if [ ! -f ${epmautomatescript} ]
```

```

then
    echo "Could not locate EPM Automate script. Please set value for
    "epmautomatescript" property in the input.properties file."
    exit
fi

export JAVA_HOME=${jdkdir}

debugmessage() {
    # logdebugmessages is defined (or not) in testbase input.properties
    if [ "${logdebugmessages}" = "true" ]
    then
        logmessage "$1"
    fi
}

logmessage()
{
    local message=$1
    local _mydate=$(date)

    echo "[$_mydate] ${message}" >> "$logdir/$logfile"
}

echoandlogmessage()
{
    local message=$1
    local _mydate=$(date)

    echo "[$_mydate] ${message}" | tee -a ${logdir}/${logfile}
}

logoutput()
{
    date=`date`
    op="$1"
    opoutput="$2"
    returncode="$3"

    #If error
    #if grep -q "EPMAT-" <<< "$2"
    if [ $returncode -ne 0 ]
    then
        failmessage="[${date}] ${operationfailuremessage} ${op}"
        logmessage "${failmessage}"
        logmessage "${opoutput}"
        logmessage "return code: ${returncode}"
    else
        successmessage="${operationsuccessmessage} ${op}"
        logmessage "${successmessage}"
        logmessage "${opoutput}"
        logmessage "return code: ${returncode}"
    fi
}

```

```

getLatestReleaseSnapshotDir()
{
    local snapshotdirs=$(find ${parentsnapshotdirectory} -type d |
sort)
    debugmessage "snapshot directories: ${snapshotdirs}"
    local latestreleasesnapshotdir=$(echo ${snapshotdirs}##*$\n | rev
| cut -d' ' -f1 | rev)
    debugmessage "latest release snapshot dir: $
{latestreleasesnapshotdir}"
    echo "${latestreleasesnapshotdir}"
}

getNextDate()
{
    local thisyearmonth=$1
    local thisyear=$(echo ${thisyearmonth} | cut -d'.' -f1)
    local thismonth=$(echo ${thisyearmonth} | cut -d'.' -f2)

    intthismonth=$(bc <<< ${thismonth})
    intthisyear=$(bc <<< ${thisyear})

    if [ ${intthismonth} -eq 12 ]
    then
        local intnextmonth=1
        local intnextyear=$((intthisyear+1))
    else
        local intnextmonth=$((intthismonth+1))
        local intnextyear=${intthisyear}
    fi

    nextmonth=$(printf "%02d\n" ${intnextmonth})
    nextyear=$(printf "%02d\n" ${intnextyear})

    debugmessage "next date: ${nextyear}.${nextmonth}"

    echo "${nextyear}.${nextmonth}"
}

init()
{
    if [ ! -d "$logdir" ]
    then
        mkdir $logdir
    fi

    # removing existing epmautomate debug logs
    if ls /*.log >/dev/null 2>&1
    then
        rm /*.log
    fi

    # remove existing log files
    if [ -f "${logdir}/${logfile}" ]
    then
        rm ${logdir}/${logfile}
    fi
}

```

```

        fi
    }

processCommand()
{
    op="$1"
    date=`date`

    logmessage "$operationmessage $op"
    operationoutput=`eval "$epmautomatescript $op"`
    logoutput "$op" "$operationoutput" "$?"
}

processSnapshot()
{
    local snapshotfile="$1"
    local nextdate="$2"
    local snapshotname=$(echo "${snapshotfile}" | rev | cut -d'/' -f1 | rev
| cut -d'.' -f1)

    processCommand "login ${username} ${userpassword} ${serviceurl} ${
proxyserverusername} ${proxyserverpassword}"
    processCommand "recreate -f"
    processCommand "uploadfile ${snapshotfile}"
    processCommand "importsnapshot \"${snapshotname}\""
    processCommand "runDailyMaintenance skipNext=true -f"
    processCommand "downloadfile \"Artifact Snapshot\""
    processCommand "deletefile \"${snapshotname}\""
    processCommand "logout"

    if [ ! -d ${parentsnapshotdirectory}/${nextdate} ]
    then
        mkdir ${parentsnapshotdirectory}/${nextdate}
    fi
runDailyMaintenance -f
    logmessage "Renaming \"Artifact Snapshot.zip\" to ${snapshotname}.zip
and moving to ${parentsnapshotdirectory}/${nextdate}"
    mv "${workingdir}/Artifact Snapshot.zip" "${workingdir}/${
snapshotname}.zip" >> "$logdir/$logfile" 2>&1
    mv "${workingdir}/${snapshotname}.zip" ${parentsnapshotdirectory}/${
nextdate} >> "$logdir/$logfile" 2>&1
}

callSendMail() {

    if [[ "${emailtoaddress}" == *@"* ]]
    then
        ${epmautomatescript} login ${username} ${userpassword} ${serviceurl}
        ${epmautomatescript} uploadFile "$logdir/$logfile"
        ${epmautomatescript} sendMail $emailtoaddress "Recreating An Old EPM
Cloud Environment results" Body="The results of recreating an old EPM Cloud
Environment are attached" Attachments=$logfile
        ${epmautomatescript} deleteFile "$logfile"
        ${epmautomatescript} logout
    fi
}

```

```

}

#----- main body of processing
date
echoandlogmessage "Starting upgrade snapshots processing"
init
latestreleasesnapshotdir=$(getLatestReleaseSnapshotDir)
latestreleasedate=$(echo "${latestreleasesnapshotdir}" | rev | cut -
d'/' -f1 | rev)
debugmessage "latest release date: ${latestreleasedate}"
nextdate=$(getNextDate ${latestreleasedate})

snapshotfiles=$(find ${latestreleasesnapshotdir} -type f -name \*.zip
| tr "\n" "|")
if [ ${#snapshotfiles} -eq 0 ]
then
    echoandlogmessage "No snapshot files found in directory $
{latestreleasesnapshotdir}"
fi

IFS="|"
for snapshotfile in $snapshotfiles
do
    echoandlogmessage "Processing snapshotfile: ${snapshotfile}"
    processSnapshot ${snapshotfile} ${nextdate}
done
unset IFS
echoandlogmessage "Upgrade snapshots processing completed."
callSendMail

```

Audit und Compliance des Datenbankzugriffs automatisieren

Verwenden Sie die PowerShell- und Bash Shell-Skripte in diesem Abschnitt, um EPM Automate-Befehle zum Erfassen von Audit- und Compliance-Daten im Zusammenhang mit dem manuellen Datenbankzugriff zu nutzen.

Mit diesen Skripten können Sie folgende Aufgaben ausführen:

- Aktivitätsbericht für den aktuellen Tag herunterladen
- Bericht parsen, um zu bestimmen, ob manueller Datenbankzugriff für die Umgebung gemeldet wird
- `./reports/dataAccessAuditReport.txt` relativ zu dem Verzeichnis erstellen, in dem Sie das Skript ausführen. Im Bericht werden der Zeitpunkt des Datenbankzugriffs sowie der SQL-Befehl aufgelistet, der ausgeführt wurde. Dies ist eine kumulative Datei, in der die aktuellsten Informationen oben angezeigt werden. Zu den verfügbaren Informationen gehören:
 - Datum und Uhrzeit der Generierung des Berichts.
 - Datenbankzugriffsdetails, falls verfügbar. Datenbankzugriffe ohne Serviceanforderung und Datenbankzugriff mit Serviceanforderung werden in getrennten Abschnitten aufgelistet. Wenn im Aktivitätsbericht kein manueller Datenbankzugriff gemeldet wird, lautet der Bericht `No SQL statements executed`.

- Senden Sie den Bericht optional an eine angegebene E-Mail-Adresse.

So automatisieren Sie Audit und Compliance des Datenbankzugriffs:

1. Kopieren Sie eines der Skripte aus den folgenden Abschnitten in eine Datei, und speichern Sie es in Ihrem Dateisystem. Nennen Sie die Datei `parseActivityReport.ps1` (für Windows siehe [PowerShell-Skript \(parseActivityReport.ps1\)](#)) oder `parseActivityReport.sh` (für Linux/UNIX siehe [Bash Shell-Skript \(parseActivityReport.sh\)](#)).
2. **Nur Windows:** Erstellen Sie eine Batchdatei namens `parseActivityReport.bat`, indem Sie das folgende Skript in eine Datei kopieren. Speichern Sie die Datei in dem Verzeichnis, in dem `parseActivityReport.ps1` gespeichert ist.

```
@echo off
set paramRequiredMessage=Syntax: parseActivityReport.bat USERNAME
PASSWORD/PASSWORD_FILE URL [REPORT_EMAIL_TO_ADDRESS]

if "%~1" == "" (
    echo User Name is missing.
    echo %paramRequiredMessage%
    exit /b 1
)
if "%~2" == "" (
    echo Password or Password_File is missing.
    echo %paramRequiredMessage%
    exit /b 1
)
if "%~3" == "" (
    echo URL is missing.
    echo %paramRequiredMessage%
    exit /b 1
)

PowerShell.exe -File parseActivityReport.ps1 %*
```

3. Ändern Sie `parseActivityReport.bat` (Windows) bzw. `parseActivityReport.sh` (Linux/UNIX), um die Werte für die Parameter in der folgenden Tabelle festzulegen.

Tabelle 3-10 In Skripte einzuschließende Variablenwerte

Variable	Beschreibung
<code>epmuser</code>	Benutzername eines Serviceadministrators Beispiele: Windows: <code>set epmuser="jDoe"</code> Linux/UNIX: <code>epmuser="jDoe"</code>

Tabelle 3-10 (Fortsetzung) In Skripte einzuschließende Variablenwerte

Variable	Beschreibung
epmpassword	<p>Kennwort des Serviceadministrators oder Speicherort der verschlüsselten Kennwortdatei. Informationen zum Erstellen einer verschlüsselten Kennwortdatei finden Sie im Abschnitt zum Befehl encrypt.</p> <p>Wenn Ihr Kennwort Sonderzeichen enthält, finden Sie unter Sonderzeichen verarbeiten weitere Informationen.</p> <p>Beispiele:</p> <p>Windows: set epmpassword = "Example"</p> <p>Linux/UNIX: epmpassword="Example"</p>
epmurl	<p>Die URL der Oracle Enterprise Performance Management Cloud-Umgebung.</p> <p>Beispiele:</p> <p>Windows: set epmurl="https://example.oraclecloud.com"</p> <p>Linux/UNIX: epmurl="https://example.oraclecloud.com"</p>
report_email_to_addresses	<p>Optional, die E-Mail-Adresse, an die der Bericht gesendet werden soll. Der Bericht wird nur per E-Mail gesendet, wenn dieser Wert angegeben wurde.</p> <p>Beispiel: john.doe@example.com</p>

- Nur für parseActivityReport.sh:** Stellen Sie sicher, dass die folgenden Werte für Ihr System richtig festgelegt sind:
 - `JAVA_HOME`
 - Speicherort von `epmautomatescript.sh`, indem Sie den Wert der Anweisung `epmautomatescript` aktualisieren
- Planen Sie mithilfe eines im Betriebssystem verfügbaren Schedulers einmal täglich die Ausführung von `parseActivityReport.bat` (womit `parseActivityReport.ps1` ausgeführt wird) oder `parseActivityReport.sh`. Informationen hierzu finden Sie unter [Skriptausführung automatisieren](#).

PowerShell-Skript (parseActivityReport.ps1)

```
# Parse Activity Report script

$epmuser=$args[0]
$epmpassword=$args[1]
$epmurl=$args[2]
$reportemailtoaddress=$args[3]

$logdir="./logs"
$logfile="${logdir}/data_access.log"
$reportdir="./reports"
$reportfile="${reportdir}/dataAccessAuditReport.txt"
$matchfile="${reportdir}/matchfile.txt"
$nosrfile="${reportdir}/data_access_nosr.csv"
$srfile="${reportdir}/data_access_sr.csv"
$aprfilelist="${reportdir}/aprfilelist.txt"
$activityreportfilelist="${reportdir}/activityreportfiles.txt"
```

```

$activityreportregex='apr/[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}_[0-9]{2}_[0-9]{2}/[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}_[0-9]{2}_[0-9]{2}.html'

$global:activityreportfile=""

$NO_SQL_EXECUTED_STATEMENT="No SQL statements executed"
$SQL_WITH_SR_EXECUTED_STATEMENT="SQL statements executed with an SR"
$SQL_WITH_NO_SR_EXECUTED_STATEMENT="SQL statements executed without an SR"

function DownloadLatestActivityReport() {
    epmautomate.bat login ${epmuser} ${epmpassword} ${epmurl} >> ${logfile}
    epmautomate.bat listfiles > ${aprfilelist}
    foreach ($line in Get-Content $aprfilelist) {
        if ($line -match $activityreportregex){
            echo "$line" >> $activityreportfilelist
        }
    }
    $global:activityreportfile=Get-Content ${activityreportfilelist} -Tail 1
    $global:activityreportfile=$global:activityreportfile.trim()
    echo " "
    echo "Processing activity report file: $global:activityreportfile" | tee
-a ${logfile}
    epmautomate.bat downloadfile "$global:activityreportfile" >> ${logfile}
    epmautomate.bat logout >> ${logfile}
}

function deleteLine($file, $start, $end) {
    $i = 0
    $start--
    $end--
    (Get-Content $file) | where{
        ($i -lt $start -or $i -gt $end)
        $i++
    } > $file
    #(Get-Content $file)
}

function GenerateCsvs()
{
    $sqlregex='<DIV id="Database">.*?</DIV>'
    $activityreportfilename=Split-Path $global:activityreportfile -leaf

    echo "Creating CSV file: ${matchfile} from data in activityreportfile: $
{activityreportfilename}" >> ${logfile}
    # remove tab and newline characters
    $activityreportexists=Test-Path "$activityreportfilename"
    if ($activityreportexists) {
        (Get-Content "$activityreportfilename") -join ' ' | Set-Content
"$activityreportfilename"
        (Get-Content "$activityreportfilename") -replace "`t", "" | Set-
Content "$activityreportfilename"
    }

    # capture text matching regex
    $string=Get-Content $activityreportfilename

```

```

    $ans=$string -match $sqlregex

    if ($ans -eq "True") {
        $Matches.0 > $matchfile
        # remove HTML tags, etc.
        (Get-Content "$matchfile") -replace "<tr", "`n<tr" | Set-
Content "$matchfile"
        (Get-Content "$matchfile") -replace "<tr[^>]*>", "" | Set-
Content "$matchfile"
        (Get-Content "$matchfile") -replace "<th[^>]*>", "" | Set-
Content "$matchfile"
        (Get-Content "$matchfile") -replace "<td[^>]*>", "|" | Set-
Content "$matchfile"
        (Get-Content "$matchfile") -replace "<br>", "" | Set-Content
"$matchfile"
        (Get-Content "$matchfile") -replace "</td>", "" | Set-Content
"$matchfile"
        (Get-Content "$matchfile") -replace "</tr>", "" | Set-Content
"$matchfile"
        (Get-Content "$matchfile") -replace "\s*</table>\s*</DIV>", ""
| Set-Content "$matchfile"
        deleteLine $matchfile 1 2

        # create SR, NOSR CSV files
        Get-Content $matchfile | ForEach-Object {
            $elements=$_split('|')
            $timeval=$elements[1].Trim()
            $srval=$elements[3].Trim()
            $sqlval=$elements[4].Trim()

            if ($srval -eq "") {
                echo "${timeval}|${sqlval}" >> ${nosrfile}
            } else {
                if ($sqlval -ne "") {
                    echo "${srval}|${timeval}|${sqlval}" >> ${srfile}
                }
            }
        }

    } else { # no SQL statements in activity report
        echo "" >> ${reportfile}
        echo $(date) >> ${reportfile}
        echo "Processing activity report
file: $global:activityreportfile" >> ${reportfile}
        echo "${NO_SQL_EXECUTED_STATEMENT}" | tee -a ${reportfile}
        CleanUp
        EmailReportResults
        exit
    }
}

function ReportResults() {
    echo $(date) >> ${reportfile}
    echo "Processing activity report file: $global:activityreportfile"
    >> ${reportfile}
}

```

```

$srfileexists=Test-Path $srfile
if ($srfileexists) {
    echo "" | tee -a ${reportfile}
    echo "${SQL_WITH_SR_EXECUTED_STATEMENT}" | tee -a ${reportfile}
    echo "SR#           Time           SQL Statement" | tee -a ${reportfile}
    echo "---          ----          -----" | tee -a ${reportfile}

    # Loop through csv file and parse
    Get-Content $srfile | ForEach-Object {
        $elements=$_split(',')
        $srval=$elements[0]
        $timeval=$elements[1]
        $sqlval=$elements[2]
        echo "${srval}    ${timeval}    ${sqlval}" | tee -a ${reportfile}
    }
}

$nosrfileexists=Test-Path $nosrfile
if ($nosrfileexists) {
    echo "" | tee -a ${reportfile}
    echo "${SQL_WITH_NO_SR_EXECUTED_STATEMENT}" | tee -a ${reportfile}
    echo "Time           SQL Statement" | tee -a ${reportfile}
    echo "-----          -----" | tee -a ${reportfile}

    # Loop through csv file and parse
    Get-Content $nosrfile | ForEach-Object {
        $elements=$_split(',')
        $timeval=$elements[0]
        $sqlval=$elements[1]
        echo "${timeval}    ${sqlval}" | tee -a ${reportfile}
    }
}
EmailReportResults
}

function EmailReportResults
{
    $elements=$reportfile.split('/')
    $reportfilename=$elements[2]

    if (${reportemailtoaddress} -match "@") {
        echo "Emailing Activity Report Results" | tee -a ${logfile}
        epmautomate.bat login ${epmuser} ${epmpassword} ${epmurl} >> ${logfile}
        epmautomate.bat uploadFile $reportfile >> ${logfile}
        epmautomate.bat sendMail $reportemailtoaddress "Database Access Audit
Report Results" Body="Database Access Audit Report Results are attached."
Attachments=$reportfilename >> ${logfile}
        epmautomate.bat deleteFile $reportfilename >> ${logfile}
        epmautomate.bat logout >> ${logfile}
    }
}

function Init
{
    $logdirexists=Test-Path $logdir

```

```

if (!($logdirexists)) {
    mkdir $logdir 2>&1 | out-null
}

$reportdirexists=Test-Path $reportdir
if (!($reportdirexists)) {
    mkdir $reportdir 2>&1 | out-null
}

$logfileexists=Test-Path $logfile
if ($logfileexists) {
    rm $logfile 2>&1 | out-null
}

$matchfileexists=Test-Path $matchfile
if ($matchfileexists) {
    rm $matchfile 2>&1 | out-null
}

$nosrfileexists=Test-Path $nosrfile
if ($nosrfileexists) {
    rm $nosrfile 2>&1 | out-null
}

$srfileexists=Test-Path $srfile
if ($srfileexists) {
    rm $srfile 2>&1 | out-null
}

$aprfilelistexists=Test-Path $aprfilelist
if ($aprfilelistexists) {
    rm $aprfilelist 2>&1 | out-null
}

$activityreportfilelistexists=Test-Path $activityreportfilelist
if ($activityreportfilelistexists) {
    rm $activityreportfilelist 2>&1 | out-null
}
}

function CleanUp
{
    $matchfileexists=Test-Path $matchfile
    if ($matchfileexists) {
        rm $matchfile 2>&1 | out-null
    }

    $aprfilelistexists=Test-Path $aprfilelist
    if ($aprfilelistexists) {
        rm $aprfilelist 2>&1 | out-null
    }

    $activityreportfilelistexists=Test-Path $activityreportfilelist
    if ($activityreportfilelistexists) {
        rm $activityreportfilelist 2>&1 | out-null
    }
}

```

```

    }
}

Init
DownloadLatestActivityReport
GenerateCsvs
ReportResults
CleanUp

```

Bash Shell-Skript (parseActivityReport.sh)

```

#!/bin/sh

export JAVA_HOME=/scratch/dteHome/autoWork/jdk1.8.0_191
epmautomatescript=/scratch/dteHome/autoWork/epmautomate/19.11.55/bin/
epmautomate.sh

epmuser=""
epmpwd=""
epmurl=""
reportemailtoaddress=""

logdir=./logs
logfile="${logdir}/data_access.log"
reportdir=./reports
reportfile="${reportdir}/dataAccessAuditReport.txt"
nosrfile="${reportdir}/data_access_nosr.csv"
srfile="${reportdir}/data_access_sr.csv"
matchfile="${reportdir}/match.out"
aprfilelist="${reportdir}/aprfilelist.txt"
activityreportfile=""
activityreportregex='apr/[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}_[0-9]{2}_[0-9]{2}/[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}_[0-9]{2}_[0-9]{2}.html'

NO_SQL_EXECUTED_STATEMENT="No SQL statements executed".
SQL_WITH_SR_EXECUTED_STATEMENT="SQL statements executed with an SR"
SQL_WITH_NO_SR_EXECUTED_STATEMENT="SQL statements executed without an SR"

cd "$(dirname "$0")"

generateCsvs ()
{
    local sqlregex='<DIV id="Database">.*?</DIV>'
    local activityreportfilename=$(echo "${activityreportfile}" | rev | cut -
d/' ' -f1 | rev)

    echo "Creating CSV file: ${matchfile} from data in activityreportfile: $
{activityreportfilename}" >> ${logfile}
    # remove tab and newline characters
    cat "${activityreportfilename}" | tr -d "\t\n\r" > ${matchfile}
    # capture text matching regex
    grep -Po "${sqlregex}" ${matchfile} > ${matchfile}.tmp

    # remove HTML tags, etc.

```

```

    sed -e 's/<tr/\n<tr/g' -e 's/<tr[^>]*>//g' -e 's/<th[^>]*>//g' -e
's/<td[^>]*>//g' -e 's/<br>//g' -e 's|</td>||g' -e 's|</tr>||g' -e 's|
[ ]*</table></DIV>||g' -e 's|[[ ]*|/|g' -e 's/[ ]*|/|g' -e 's/<DIV
id="Database">.*<!-- Print Tables -->\n//g' ${matchfile}.tmp > ${
matchfile}

# create SR, NOSR CSV files
while read line
do
    timeval=$(echo "${line}" | cut -d'|' -f2)
    srval=$(echo "${line}" | cut -d'|' -f4)
    sqlval=$(echo "${line}" | cut -d'|' -f5)

    if [[ "${srval}" == "" ]]
    then
        echo "${timeval}|${sqlval}" >> ${nosrfile}
    else
        if [[ "${sqlval}" != "" ]]
        then
            echo "${srval}|${timeval}|${sqlval}" >> ${srfile}
        fi
    fi
done < ${matchfile}
}

reportResults() {
    echo $(date) >> ${reportfile}
    echo "Processing activity report file: $activityreportfile" >> $
{reportfile}
    if [[ -f ${srfile} ]]
    then
        echo "" | tee -a ${reportfile}
        echo "${SQL_WITH_SR_EXECUTED_STATEMENT}" | tee -a ${reportfile}
        echo "SR#          Time          SQL Statement" | tee -a $
{reportfile}
        echo "---          ----          -----" | tee -a $
{reportfile}
        while read line
        do
            srval=$(echo "${line}" | cut -d'|' -f1)
            timeval=$(echo "${line}" | cut -d'|' -f2)
            sqlval=$(echo "${line}" | cut -d'|' -f3)
            echo "${srval}    ${timeval}    ${sqlval}" | tee -a $
{reportfile}
        done < ${srfile}
    fi

    if [[ -f ${nosrfile} ]]
    then
        echo "" | tee -a ${reportfile}
        echo "${SQL_WITH_NO_SR_EXECUTED_STATEMENT}" | tee -a $
{reportfile}
        echo "Time          SQL Statement" | tee -a ${reportfile}
        echo "----          --- -----" | tee -a ${reportfile}
        while read line

```

```

do
    timeval=$(echo "${line}" | cut -d'|' -f1)
    sqlval=$(echo "${line}" | cut -d'|' -f2)
    echo "${timeval}    ${sqlval}" | tee -a ${reportfile}
done < ${nosrfile}
fi

if [[ ! -f ${srfile} ]] && [[ ! -f ${nosrfile} ]]
then
    echo "" | tee -a ${reportfile}
    echo "${NO_SQL_EXECUTED_STATEMENT}" | tee -a ${reportfile}
fi

emailReportResults
}

downloadLatestActivityReport() {
    ${epmautomatescript} login ${epmuser} ${epmpwd} ${epmurl} >> ${logfile}
    ${epmautomatescript} listfiles > ${aprfilelist}
    activityreportfile=$(cat ${aprfilelist} | grep -P "${
activityreportregex}" | tail -n 1 | sed -e 's/^ //' )
    echo " "
    echo "Processing activity report file: ${activityreportfile}" | tee -a $
{logfile}
    ${epmautomatescript} downloadfile "${activityreportfile}" >> ${logfile}
    ${epmautomatescript} logout >> ${logfile}
}

emailReportResults() {
    reportfilename=$(echo "${reportfile}" | cut -d'/' -f3)

    if [[ "${reportemailtoaddress}" == *@"* ]]
    then
        echo "Emailing Activity Report Results" | tee -a ${logfile}
        ${epmautomatescript} login ${epmuser} ${epmpwd} ${epmurl} >> $
{logfile}
        ${epmautomatescript} uploadFile "$reportfile" >> ${logfile}
        ${epmautomatescript} sendMail $reportemailtoaddress "Database Access
Audit Report Results" Body="Database Access Audit Report Results are
attached." Attachments=$reportfilename >> ${logfile}
        ${epmautomatescript} deleteFile "$reportfilename" >> ${logfile}
        ${epmautomatescript} logout >> ${logfile}
    fi
}

checkParams()
{
    if [ -z "$epmuser" ]
    then
        echo "Username is missing."
        echo "Syntax: parseActivityReport.sh USERNAME PASSWORD URL"
        exit 2
    fi

    if [ -z "$epmpwd" ]

```



```

then
    echo "Password is missing."
    echo "Syntax: parseActivityReport.sh USERNAME PASSWORD URL"
    exit 2
fi

if [ -z "$epmurl" ]
then
    echo "URL is missing."
    echo "Syntax: parseActivityReport.sh USERNAME PASSWORD URL"
    exit 2
fi
}

init()
{
    checkParams

    if [ ! -d "${logdir}" ]
    then
        mkdir ${logdir}
    fi

    if [ ! -d "${reportdir}" ]
    then
        mkdir ${reportdir}
    fi

    if [ ! -f "${epmautomatescript}" ]
    then
        echo "Cannot locate EPMAutomate script: ${epmautomatescript}.
Please check setting and run script again. Exiting." | tee -a $
{logfile}
        exit
    fi

    if [ -f "${srfile}" ]
    then
        rm ${srfile}
    fi

    if [ -f "${nosrfile}" ]
    then
        rm ${nosrfile}
    fi

    if [ -f "${matchfile}" ]
    then
        rm ${matchfile}
    fi

    if [ -f "${aprfilelist}" ]
    then
        rm ${aprfilelist}
    fi
}

```

```
}

cleanup()
{
    if [ -f "${matchfile}" ]
    then
        rm ${matchfile}
    fi

    if [ -f "${matchfile}.tmp" ]
    then
        rm ${matchfile}.tmp
    fi

    if [ -f "${aprfilelist}" ]
    then
        rm ${aprfilelist}
    fi
}

init
downloadLatestActivityReport
generateCsvs
reportResults
cleanup
```

Benutzer und vordefinierte Rollenzuweisungen replizieren

Mit den Skripten in diesem Abschnitt können Sie Benutzer und vordefinierte Rollenzuweisungen aus einer Umgebung in eine andere migrieren.

Informationen zu den Skripten

Sie verwenden zwei separate Skripte: eines zum Replizieren von Benutzern in Identitätsdomains und ein weiteres zum Replizieren vordefinierter Rollenzuweisungen der Benutzer. Die Ausführung dieser Skripte erfolgt in der folgenden Reihenfolge:

- Führen Sie das Skript zum Replizieren von Benutzern (`replicateusers`) aus, und überprüfen Sie, dass alle Benutzer in der Zielidentitätsdomain erstellt werden. Der Benutzer, der dieses Skript ausführt, muss in beiden Umgebungen über die Rolle Identitätsdomainadministrator und Serviceadministrator verfügen.
- Führen Sie das Skript zum Replizieren von Rollenzuweisungen aus (`replicatepredefinedroles`).

Hinweis:

- Wenn die Kennwörter Sonderzeichen enthalten, finden Sie unter [Sonderzeichen verarbeiten](#) weitere Informationen.
- Die Skripte in diesem Abschnitt funktionieren nur für vordefinierte Rollen: Serviceadministrator, Poweruser, Benutzer und Leseberechtigter.

Skripte ausführen

Informationen zum Erstellen der erforderlichen Skripte und Batchdateien finden Sie in diesen Themen:

- [Benutzer aus einer Identitätsdomain in eine andere replizieren](#)
- [Vordefinierte Rollenzuweisungen aus einer Umgebung in eine andere replizieren](#)

Schritte unter Windows

1. Erstellen Sie die Skripte `replicateusers.bat`, `replicateusers.ps1`, `replicatepredefinedroles.bat` und `replicatepredefinedroles.ps1`, und speichern Sie sie in einem lokalen Verzeichnis, in dem Sie Schreib- und Ausführungsberechtigungen haben.
2. Aktualisieren Sie die Batchdateien mit Informationen für die Quell- und Zielumgebung und den Internetproxyserver, falls erforderlich.
3. Führen Sie `replicateusers.bat` aus, wodurch `replicateusers.ps1` ausgeführt wird. Sie müssen das Standardkennwort, das replizierten Benutzern als Befehlszeilenparameter zugeordnet werden soll, wie folgt angeben:

```
replicateusers.bat Pwd_for_users
```

Wenn das Kennwort Sonderzeichen enthält, achten Sie darauf, dass Sie das entsprechende Escapezeichen verwenden. Informationen hierzu finden Sie unter [Sonderzeichen verarbeiten](#).
4. Führen Sie `replicatepredefinedroles.bat` aus, um Rollenzuweisungen zu erstellen, die mit denen identisch sind, die in der Quellumgebung vorhanden sind.

Schritte unter Linux/UNIX

1. Erstellen Sie die Skripte `replicateusers.sh` und `replicatepredefinedroles.sh`, und speichern Sie sie in einem lokalen Verzeichnis, in dem Sie Schreib- und Ausführungsberechtigungen haben.
2. Aktualisieren Sie `replicateusers.sh` und `replicatepredefinedroles.sh` mit Informationen für die Quell- und Zielumgebungen und den Internetproxyserver, sofern erforderlich.
3. Führen Sie `replicateusers.sh` aus. Sie müssen das Standardkennwort, das replizierten Benutzern als Befehlszeilenparameter zugeordnet werden soll, wie folgt angeben:

```
./replicateusers.sh Pwd_for_users
```

Wenn das Kennwort Sonderzeichen enthält, achten Sie darauf, dass Sie das entsprechende Escapezeichen verwenden. Informationen hierzu finden Sie unter [Sonderzeichen verarbeiten](#).
4. Führen Sie das Skript `replicatepredefinedroles.sh` aus, um Rollenzuweisungen zu erstellen, die mit denen identisch sind, die in der Quellumgebung vorhanden sind.

Benutzer aus einer Identitätsdomain in eine andere replizieren

Verwenden Sie die Skripte in diesem Abschnitt, um Benutzer aus einer Identitätsdomain in eine andere Identitätsdomain zu klonen. Der Benutzer, der diese

Skripte ausführt, muss in der Quell- und in der Zielumgebung über die Rollen Identitätsdomainadministrator und Serviceadministrator verfügen.

Windows

Erstellen Sie `replicateusers.bat` und `replicateusers.ps1`, indem Sie die Skripte in diesem Abschnitt kopieren.

1. Erstellen Sie `replicateusers.ps1`, indem Sie das folgende Skript kopieren:

```
# Replicate users script

param(
    [string]$epmusersource,
    [string]$epmpwdsource,
    [string]$epmurlsource,
    [string]$epmidentitydomainsource,
    [string]$epmuserstarget,
    [string]$epmpwdtarget,
    [string]$epmurltarget,
    [string]$epmidentitydomaintarget,
    [string]$proxyserverusername,
    [string]$proxyserverpassword,
    [string]$proxyserverdomain,
    [string]$userpassword,
    [string]$resetpassword,
    [string]$emailtoaddress
)

$roleassignmentreport="roleassignmentreport.csv"
$usersreport="users.csv"

echo "Replicate users script started"

# delete existing reports
$roleassignmentreportexists=Test-Path $roleassignmentreport
if ($roleassignmentreportexists) {
    rm $roleassignmentreport 2>&1 | out-null
}

$usersreportexists=Test-Path $usersreport
if ($usersreportexists) {
    rm $usersreport 2>&1 | out-null
}

# epmautomate login Source App as an IDM Admin
echo "Logging into source application at ${epmurlsource}"
epmautomate login ${epmusersource} ${epmpwdsource} ${epmurlsource} $
{epmidentitydomainsource} ${proxyserverusername} ${proxyserverpassword} $
{proxyserverdomain}
echo "Creating role assignment report: ${roleassignmentreport}"
epmautomate roleAssignmentReport ${roleassignmentreport}
if (${emailtoaddress} -match "@") {
    epmautomate.bat sendMail $emailtoaddress "Role assignment report"
```

```

Body="Role assignment report is attached."
Attachments=${roleassignmentreport}
echo "Downloading role assignment report"
epmautomate downloadfile ${roleassignmentreport}
epmautomate deletefile ${roleassignmentreport}
epmautomate logout

# Create users report
Get-Content ${roleassignmentreport} | ForEach-Object {
    $user=$_.split(',')[0]
    $firstname=$_.split(',')[1]
    $lastname=$_.split(',')[2]
    $email=$_.split(',')[3]

    if ($firstname -eq "First Name") {
        return
    } else {
        echo "${firstname},${lastname},${email},${user}" >> $
{usersreport}
    }
}

Get-Content -Path "${usersreport}" | Sort-Object -Unique > "$
{usersreport}.tmp"
mv -Force "${usersreport}.tmp" "${usersreport}"
$userheader="First Name,Last Name,Email,User Login"
"${userheader}`r`n" + (Get-Content $usersreport -Raw) | Set-
Content $usersreport

# epmautomate login Target App as an IDM Admin
echo "Logging into target application at ${epmurltarget}"
epmautomate login ${epmusertarget} ${epmpwdtarget} ${epmurltarget} $
{epmidentitydomaintarget} ${proxyserverusername} $
{proxyserverpassword} ${proxyserverdomain}
epmautomate deletefile ${usersreport} | Out-Null
echo "Uploading file ${usersreport}"
epmautomate uploadfile ${usersreport}
echo "Adding users"
epmautomate addUsers ${usersreport} userPassword=${userpassword}
resetPassword=${resetpassword}
epmautomate deletefile ${usersreport}
epmautomate logout
rm deletefile*.log | Out-Null
echo "Replicate users script completed"

```

2. Erstellen Sie replicateusers.bat, indem Sie das folgende Skript kopieren:

```

@ECHO OFF
SET thisdir=%~dp0
SET scriptpath=%thisdir%replicateusers.ps1
SET paramRequiredMessage=Syntax: replicateusers.bat "USER_PASSWORD"

REM USER DEFINED VARIABLES
REM -----

```

```

set epusersource="<EPM USER FOR SOURCE ENVIRONMENT>"
set epmpwdsourc="<EPM PASSWORD FOR SOURCE ENVIRONMENT>"
set epmurlsource="<EPM URL FOR SOURCE ENVIRONMENT>"
set epmidentitydomainsourc="<EPM IDENTITY DOMAIN FOR SOURCE ENVIRONMENT>"
set epmusertarget="<EPM USER FOR TARGET ENVIRONMENT>"
set epmpwdtarget="<EPM PASSWORD FOR TARGET ENVIRONMENT>"
set epmurltarget="<EPM URL FOR TARGET ENVIRONMENT>"
set epmidentitydomaintarget="<EPM IDENTITY DOMAIN FOR TARGET ENVIRONMENT>"
set proxyserverusername="<PROXY SERVER USER NAME>"
set proxyserverpassword="<PROXY SERVER PASSWORD>"
set proxyserverdomain="<PROXY SERVER DOMAIN>"
set resetpassword=false
set emailtoaddress="<EMAIL_TO_ADDRESS>"
REM -----

if "%~1" == "" (
    echo USER_PASSWORD is missing. This is used to set the default
password for the replicated users.
    echo %paramRequiredMessage%
    exit /b 1
)

PowerShell -NoProfile -ExecutionPolicy Bypass -Command "& '%scriptpath%' -
epusersource '%epusersource%' -epmpwdsourc '%epmpwdsourc%' -
epmurlsource '%epmurlsource%' -epmidentitydomainsourc
'%epmidentitydomainsourc%' -epmusertarget '%epmusertarget%' -
epmpwdtarget '%epmpwdtarget%' -epmurltarget '%epmurltarget%' -
epmidentitydomaintarget '%epmidentitydomaintarget%' -proxyserverusername
'%proxyserverusername%' -proxyserverpassword '%proxyserverpassword%' -
proxyserverdomain '%proxyserverdomain%' -userpassword '%~1' -
resetpassword '%resetpassword%' -emailtoaddress '%emailtoaddress%'"

```

3. Aktualisieren Sie `replicateusers.bat`. In der folgenden Tabelle finden Sie die Werte, die Sie angeben müssen.

Parameter	Beschreibung
epusersource	Benutzername eines Benutzers mit der Rolle Identitätsdomainadministrator und Serviceadministrator in der Quellumgebung. Beispiele: Windows: <code>set epusersource="jDoe"</code> Linux/UNIX: <code>epusersource="jDoe"</code>
epmpwdsourc	Kennwort des Benutzers oder der absolute Pfad der verschlüsselten Kennwortdatei. Beispiele: Windows: <code>set epmpwdsourc="Example"</code> Linux/UNIX: <code>epmpwdsourc="Example"</code>

Parameter	Beschreibung
epmurlsource	<p>URL der Umgebung, aus der Benutzer kopiert werden sollen.</p> <p>Beispiele:</p> <p>Windows: set epmurlsource="https://example.oraclecloud.com"</p> <p>Linux/UNIX: epmurlsource="https://example.oraclecloud.com"</p>
epidentitydomainsource	<p>Name der Identitätsdomain, die von der Quellumgebung verwendet wird.</p> <p>Beispiele:</p> <p>Windows: set epidentitydomainsource="example_source_dom"</p> <p>Linux/UNIX: epidentitydomainsource="example_source_dom"</p>
epusertarget	<p>Benutzername eines Benutzers mit der Rolle Identitätsdomainadministrator und Serviceadministrator in der Zielumgebung.</p> <p>Beispiele:</p> <p>Windows: set epusertarget="John.Doe"</p> <p>Linux/UNIX: set epusertarget="John.Doe"</p>
eppwdtarget	<p>Kennwort des Benutzers oder der absolute Pfad der verschlüsselten Kennwortdatei.</p> <p>Beispiele:</p> <p>Windows: set eppwdtarget="Example1"</p> <p>Linux/UNIX: eppwdtarget="Example1"</p>
epurltarget	<p>URL der Umgebung, in der Benutzer erstellt werden sollen.</p> <p>Beispiele:</p> <p>Windows: set epurltarget="https://example.oraclecloud.com"</p> <p>Linux/UNIX: epurltarget="https://example.oraclecloud.com"</p>
epidentitydomaintarget	<p>Name der Identitätsdomain, die von der Zielumgebung verwendet wird.</p> <p>Beispiele:</p> <p>Windows: set epidentitydomaintarget="example_source_dom"</p> <p>Linux/UNIX: epidentitydomaintarget="example_target_dom"</p>
proxyserverusername	<p>Der Benutzername zum Authentifizieren einer sicheren Session auf dem Proxyserver, der den Zugriff auf das Internet steuert. Löschen Sie alle Vorkommen dieser Eigenschaft, wenn sie nicht verwendet wird.</p> <p>Beispiele:</p> <p>Windows: set proxyserverusername="Example"</p> <p>Linux/UNIX: proxyserverusername="Example"</p>

Parameter	Beschreibung
proxyserverpassword	Das Kennwort zur Authentifizierung des Benutzers beim Proxyserver. Löschen Sie alle Vorkommen dieser Eigenschaft, wenn sie nicht verwendet wird. Beispiele: Windows: set proxyserverpassword="examplePwd" Linux/UNIX: proxyserverpassword="examplePwd"
proxyserverdomain	Der Name der Domain, die für den Proxyserver definiert ist. Löschen Sie alle Vorkommen dieser Eigenschaft, wenn sie nicht verwendet wird. Beispiele: Windows: set proxyserverdomain="exampleDom" Linux/UNIX: proxyserverdomain="exampleDom"
emailtoaddress	Optional, die E-Mail-Adresse, an die der Bericht zur Rollenzuweisung gesendet werden soll. Der Bericht wird nur per E-Mail gesendet, wenn dieser Wert angegeben wurde. Beispiel: emailtoaddress=john.doe@example.com

Linux/UNIX

1. Erstellen Sie `replicateusers.sh`, indem Sie das folgende Skript kopieren.

```
#!/bin/sh

userpassword="$1"

# USER DEFINED VARIABLES
#-----
javahome=""
epmautomatescript=""
epmusersource=""
epmpwdsource=""
epmurlsource=""
epmidentitydomainsource=""
epmuserstarget=""
epmpwdtarget=""
epmurltarget=""
epmidentitydomaintarget=""
proxyserverusername=""
proxyserverpassword=""
proxyserverdomain=""
resetpassword="false"
emailtoaddress=""
#-----

roleassignmentreport="roleassignmentreport.csv"
usersreport="users.csv"
paramrequiredmessage='Syntax: replicateusers.sh "USER_PASSWORD"'

export JAVA_HOME=${javahome}

if [ "${userpassword}" == "" ]
then
```



```

        echo "USER_PASSWORD is missing. This is used to set the default
password for the replicated users."
        echo "${paramrequiredmessage}"
        exit
    fi

echo "Replicate users script started"

# epmautomate login Source App as an IDM Admin
echo "Logging into source application at ${epmurlsource}"
${epmautomatescript} login ${epmusersource} ${epmpwdsource} $
{epmurlsource} ${epmidentitydomainsource} ${proxyserverusername} $
{proxyserverpassword} ${proxyserverdomain}
echo "Creating role assignment report: ${roleassignmentreport}"
${epmautomatescript} roleAssignmentReport ${roleassignmentreport}
if [[ "${emailtoaddress}" == *@"*" ]]
then
    ${epmautomatescript} sendMail $emailtoaddress "Role assignment
report" Body="Role assignment report is attached."
Attachments=${roleassignmentreport}
fi
echo "Downloading role assignment report"
${epmautomatescript} downloadfile ${roleassignmentreport}
${epmautomatescript} deletefile ${roleassignmentreport}
${epmautomatescript} logout

awk -F, '{print $2","$3","$4","$1}' ${roleassignmentreport} | (read
-r; printf "%s\n" "$REPLY"; sort -u) > ${usersreport}

# epmautomate login Target App as an IDM Admin
echo "Logging into target application at ${epmurltarget}"
${epmautomatescript} login ${epmuserstarget} ${epmpwdtarget} $
{epmurltarget} ${epmidentitydomaintarget} ${proxyserverusername} $
{proxyserverpassword} ${proxyserverdomain}
${epmautomatescript} deletefile ${usersreport} > /dev/null 2>&1
echo "Uploading file ${usersreport}"
${epmautomatescript} uploadfile ${usersreport}
echo "Adding users"
${epmautomatescript} addUsers ${usersreport} userPassword=$
{userpassword} resetPassword=${resetpassword}
${epmautomatescript} deletefile ${usersreport}
${epmautomatescript} logout
rm deletefile*.log > /dev/null 2>&1

echo "Replicate users script completed"

```

2. Aktualisieren Sie `replicateusers.sh`. In der vorstehenden Tabelle finden Sie Informationen zu den Werten, die Sie angeben müssen. Darüber hinaus müssen Sie die Werte für diese Eigenschaften angeben:

- `javahome`: der absolute Pfad zu dem Verzeichnis, in dem Java installiert ist.
- `epmautomatescript`: Speicherort von `epmautomatescript.sh`; z.B. `epmautomatescript="/home/user1/epmautomate/bin/epmautomate.sh"`

Vordefinierte Rollenzuweisungen aus einer Umgebung in eine andere replizieren

Verwenden Sie die Skripte in diesem Abschnitt, um vordefinierte Rollenzuweisungen aus einer Umgebung in eine andere zu klonen. Der Benutzer, der diese Skripte ausführt, muss in beiden Umgebungen über die Rolle Serviceadministrator verfügen.

Hinweis:

Wenn Sie die PDF-Version dieses Dokuments verwenden: Um zu verhindern, dass Zeilenumbrüche und Fußzeileninformationen diese Skripte unbrauchbar machen, kopieren Sie sie aus der [HTML-Version dieses Themas](#).

Windows

1. Erstellen Sie `replicatepredefineroles.ps1`, indem Sie das folgende Skript kopieren.

```
# Replicate predefined roles script

param(
    [string]$epmusersource,
    [string]$epmpwdsource,
    [string]$epmurlsource,
    [string]$epmidentitydomainsource,
    [string]$epmusertarget,
    [string]$epmpwdtarget,
    [string]$epmurltarget,
    [string]$epmidentitydomaintarget,
    [string]$proxyserverusername,
    [string]$proxyserverpassword,
    [string]$proxyserverdomain,
    [string]$emailtoaddress
)

$roleassignmentreport="roleassignmentreport.csv"

function replicateroles
{
    # epmautomate login Source App as an IDM Admin
    echo "Logging into source application at ${epmurlsource}"
    epmautomate login ${epmusersource} ${epmpwdsource} ${epmurlsource} $
    {epmidentitydomainsource} ${proxyserverusername} ${proxyserverpassword} $
    {proxyserverdomain}
    echo "Creating role assignment report: ${roleassignmentreport}"
    epmautomate roleAssignmentReport ${roleassignmentreport}
    if (${emailtoaddress} -match "@") {
        epmautomate.bat sendMail $emailtoaddress "Role assignment report"
        Body="Role assignment report is attached."
        Attachments=$roleassignmentreport
    }
    echo "Downloading role assignment report"
    epmautomate downloadfile ${roleassignmentreport}
    epmautomate deletefile ${roleassignmentreport}
}
```

```

epmautomate logout

echo "Creating files to use with epmautomate assignRoles"

Get-Content ${roleassignmentreport} | ForEach-Object {
    $user=$_.split(',')[0]
    $rolename=$_.split(',')[4]

    if ($rolename -like '*User' -And $rolename -notlike '*Power
User') {
        $rolenamearray=$rolename.split(" ")
        $arraysize=$rolenamearray.count
        $rolename="User"
        if ($arraysize.count -le 2) {
            echo "${user}" | Out-File -Append -Encoding "UTF8"
"role-`${rolename}.csv"
        }
    }
    elseif ($rolename -like '*Viewer') {
        $rolenamearray=$rolename.split(" ")
        $arraysize=$rolenamearray.count
        $rolename="Viewer"
        if ($arraysize -le 2) {
            echo "${user}" | Out-File -Append -Encoding "UTF8"
"role-`${rolename}.csv"
        }
    }
    elseif ($rolename -like '*Power User') {
        $rolenamearray=$rolename.split(" ")
        $arraysize=$rolenamearray.count
        $rolename="Power User"
        if ($arraysize -le 3) {
            echo "${user}" | Out-File -Append -Encoding "UTF8"
"role-`${rolename}.csv"
        }
    }
    elseif ($rolename -like '*Service Administrator') {
        $rolenamearray=$rolename.split(" ")
        $arraysize=$rolenamearray.count
        $rolename="Service Administrator"
        if ($arraysize -le 3) {
            echo "${user}" | Out-File -Append -Encoding "UTF8"
"role-`${rolename}.csv"
        }
    }
    elseif ($rolename -like 'Planner') {
        echo "${user}" | Out-File -Append -Encoding "UTF8"
"role-User.csv"
    }
}

# Add header and format
$rolefiles = Get-ChildItem "role-*.csv"
foreach ($rolefile in $rolefiles) {
    $rolefilecontent = Get-Content "$rolefile"
}

```

```

$headerline='User Login'
Set-Content $rolefile -value $headerline,$rolefilecontent
$txt = [io.file]::ReadAllText("$rolefile") -replace "`r`n","`n"
[io.file]::WriteAllText("$rolefile", $txt)
}

# epmautomate login Target App as an IDM Admin
echo "Logging into target application at ${epmurltarget}"
epmautomate login ${epmusertarget} ${epmpwdtarget} ${epmurltarget} $
{epmidentitydomaintarget} ${proxyserverusername} ${proxyserverpassword} $
{proxyserverdomain}

$rolefiles = Get-ChildItem "role-*.csv"
foreach ($rolefile in $rolefiles) {
    $rolenamecsv=$rolefile.BaseName.split('-')[1]
    $rolename=$rolenamecsv.split('.')[0]
    epmautomate deletefile "${rolefile}" | Out-Null
    echo "Uploading file ${rolefile}"
    epmautomate uploadfile "${rolefile}"
    echo "Assigning ${rolename} roles"
    epmautomate assignRole "role-${rolename}.csv" "${rolename}"
    epmautomate deletefile "role-${rolename}.csv"
}
epmautomate logout
rm deletefile*.log | Out-Null
}

function init
{
    # delete ${role}.csv files
    $rolefiles = Get-ChildItem "role-*.csv"
    foreach ($rolefile in $rolefiles) {
        $rolefileexists=Test-Path $rolefile
        if ($rolefileexists) {
            rm "${rolefile}"
        }
    }
}

echo "Replicate predefined roles script started"
init
replicateroles
echo "Replicate predefined roles script completed"

```

2. Erstellen Sie replicaterepredefineroles.bat, indem Sie das folgende Skript kopieren.

```

@ECHO OFF
SET thisdir=%~dp0
SET scriptpath=%thisdir%replicaterepredefineroles.ps1

REM USER DEFINED VARIABLES
REM -----
set epusersource="<EPM USER FOR SOURCE ENVIRONMENT>"
set epmpwdsource="<EPM PASSWORD FOR SOURCE ENVIRONMENT>"
set epurlsource="<EPM URL FOR SOURCE ENVIRONMENT>"

```

```
set epmidentitydomainsource="<EPM IDENTITY DOMAIN FOR SOURCE ENVIRONMENT>"
set epmuserstarget="<EPM USER FOR TARGET ENVIRONMENT>"
set epmpwdtarget="<EPM PASSWORD FOR TARGET ENVIRONMENT>"
set epmurltarget="<EPM URL FOR TARGET ENVIRONMENT>"
set epmidentitydomaintarget="<EPM IDENTITY DOMAIN FOR TARGET ENVIRONMENT>"
set proxyserverusername="<PROXY SERVER USER NAME>"
set proxyserverpassword="<PROXY SERVER PASSWORD>"
set proxyserverdomain="<PROXY SERVER DOMAIN>"
set emailtoaddress="<EMAIL_TO_ADDRESS>"
REM -----
```

```
PowerShell -NoProfile -ExecutionPolicy Bypass -Command "&
'%scriptpath%' -epmusersource '%epmusersource%' -epmpwdsourc
e '%epmpwdsourc%' -epmurlsource '%epmurlsource%' -
epmidentitydomainsource '%epmidentitydomainsource%' -epmuserstarget
'%epmuserstarget%' -epmpwdtarget '%epmpwdtarget%' -epmurltarget
'%epmurltarget%' -epmidentitydomaintarget
'%epmidentitydomaintarget%' -proxyserverusername
'%proxyserverusername%' -proxyserverpassword
'%proxyserverpassword%' -proxyserverdomain '%proxyserverdomain%' -
emailtoaddress '%emailtoaddress%'"
```

3. Aktualisieren Sie `replicatepredefineroles.bat` je nach Bedarf. In der folgenden Tabelle finden Sie Informationen zu den Werten, die Sie für die Eigenschaften in dieser Datei festlegen müssen.

replicatepredefineroles.bat aktualisieren

Parameter	Beschreibung
epmusersource	Benutzername eines Benutzers mit der Rolle Identitätsdomainadministrator und Serviceadministrator in der Quellumgebung. Beispiele: Windows: set epmusersource="jDoe" Linux/UNIX: epmusersource="jDoe"
epmpwdsourc	Kennwort des Benutzers oder der absolute Pfad der verschlüsselten Kennwortdatei. Beispiele: Windows: set epmpwdsourc="Example" Linux/UNIX: epmpwdsourc="Example"
epmurlsource	URL der Umgebung, aus der Benutzer kopiert werden sollen. Beispiele: Windows: set epmurlsource="https://example.oraclecloud.com" Linux/UNIX: epmurlsource="https://example.oraclecloud.com"

Parameter	Beschreibung
epmidentitydomainsource	<p>Name der Identitätsdomain, die von der Quellumgebung verwendet wird.</p> <p>Beispiele:</p> <p>Windows: set epmidentitydomainsource="example_source_dom"</p> <p>Linux/UNIX: epmidentitydomainsource="example_source_dom"</p>
epmusertarget	<p>Benutzername eines Benutzers mit der Rolle Identitätsdomainadministrator und Serviceadministrator in der Zielumgebung.</p> <p>Beispiele:</p> <p>Windows: set epmusertarget="John.Doe"</p> <p>Linux/UNIX: set epmusertarget="John.Doe"</p>
epmpwdtarget	<p>Kennwort des Benutzers oder der absolute Pfad der verschlüsselten Kennwortdatei.</p> <p>Beispiele:</p> <p>Windows: set epmpwdtarget="Example1"</p> <p>Linux/UNIX: epmpwdtarget="Example1"</p>
epmurltarget	<p>URL der Umgebung, in der Benutzer erstellt werden sollen.</p> <p>Beispiele:</p> <p>Windows: set epmurltarget="https://example.oraclecloud.com"</p> <p>Linux/UNIX: epmurltarget="https://example.oraclecloud.com"</p>
epmidentitydomaintarget	<p>Name der Identitätsdomain, die von der Zielumgebung verwendet wird.</p> <p>Beispiele:</p> <p>Windows: set epmidentitydomaintarget="example_target_dom"</p> <p>Linux/UNIX: epmidentitydomaintarget="example_target_dom"</p>
proxyserverusername	<p>Der Benutzername zum Authentifizieren einer sicheren Session auf dem Proxyserver, der den Zugriff auf das Internet steuert. Löschen Sie alle Vorkommen dieser Eigenschaft, wenn sie nicht verwendet wird.</p> <p>Beispiele:</p> <p>Windows: set proxyserverusername="Example"</p> <p>Linux/UNIX: proxyserverusername="Example"</p>
proxyserverpassword	<p>Das Kennwort zur Authentifizierung des Benutzers beim Proxyserver. Löschen Sie alle Vorkommen dieser Eigenschaft, wenn sie nicht verwendet wird.</p> <p>Beispiele:</p> <p>Windows: set proxyserverpassword="examplePwd"</p> <p>Linux/UNIX: proxyserverpassword="examplePwd"</p>

Parameter	Beschreibung
proxyserverdomain	Der Name der Domain, die für den Proxyserver definiert ist. Löschen Sie alle Vorkommen dieser Eigenschaft, wenn sie nicht verwendet wird. Beispiele: Windows: set proxyserverdomain="exampleDom" Linux/UNIX: proxyserverdomain="exampleDom"
emailtoaddress	Optional, die E-Mail-Adresse, an die der Bericht zur Rollenzuweisung gesendet werden soll. Der Bericht wird nur per E-Mail gesendet, wenn dieser Wert angegeben wurde. Beispiel: emailtoaddress=john.doe@example.com

Linux/UNIX

1. Erstellen Sie replicatepredefineroles.sh, indem Sie das folgende Skript kopieren.

```
#!/bin/sh

# USER DEFINED VARIABLES
#-----
javahome=""
epmautomatescript=""
epmusersource=""
epmpwdsource=""
epmurlsource=""
epmidentitydomainsource=""
epmusertarget=""
epmpwdtarget=""
epmurltarget=""
epmidentitydomaintarget=""
proxyserverusername=""
proxyserverpassword=""
proxyserverdomain=""
emailtoaddress=""
#-----

roleassignmentreport="roleassignmentreport.csv"

export JAVA_HOME=${javahome}

replicateroles()
{
    # epmautomate login Source App as an DM Admin
    echo "Logging into source application at ${epmurlsource}"
    ${epmautomatescript} login ${epmusersource} ${epmpwdsource} $
    {epmurlsource} ${epmidentitydomainsource} ${proxyserverusername} $
    {proxyserverpassword} ${proxyserverdomain}
    echo "Creating role assignment report: ${roleassignmentreport}"
    ${epmautomatescript} roleAssignmentReport $
}
```

```

{roleassignmentreport}
  if [[ "${emailtoaddress}" == *@"* ]]
  then
    ${epmautomatescript} sendMail $emailtoaddress "Role assignment
report" Body="Role assignment report is attached."
Attachments=${roleassignmentreport}
  fi
  echo "Downloading role assignment report"
  ${epmautomatescript} downloadfile ${roleassignmentreport}
  ${epmautomatescript} deletefile ${roleassignmentreport}
  ${epmautomatescript} logout

  echo "Creating files to use with epmautomate assignRoles"
  while read line
  do
    user=$(echo "${line}" | cut -d',' -f1)
    rolename=$(echo "${line}" | cut -d',' -f5)

    if [[ "$rolename" == *"User" ]] && [[ "$rolename" != *"Power
User" ]]
    then
      count=$(echo "${rolename}" | wc -w);
      rolename="User"
      if [[ $count -le 2 ]]
      then
        echo "${user}" >> "role-${rolename}.csv"
      fi
    elif [[ "$rolename" == *"Viewer" ]]
    then
      count=$(echo "${rolename}" | wc -w);
      rolename="Viewer"
      if [[ $count -le 2 ]]
      then
        echo "${user}" >> "role-${rolename}.csv"
      fi
    elif [[ "$rolename" == *"Power User" ]]
    then
      count=$(echo "${rolename}" | wc -w);
      rolename="Power User"
      if [[ $count -le 3 ]]
      then
        echo "${user}" >> "role-${rolename}.csv"
      fi
    elif [[ "$rolename" == *"Service Administrator" ]]
    then
      count=$(echo "${rolename}" | wc -w);
      rolename="Service Administrator"
      if [[ $count -le 3 ]]
      then
        echo "${user}" >> "role-${rolename}.csv"
      fi
    elif [[ "$rolename" == "Planner" ]]
    then
      echo "${user}" >> "role-User.csv"
    fi
  fi

```



```

done < ${roleassignmentreport}

# write header line
for f in role-*.csv
do
    sed -i 'liUser Login' "$f"
done

# epmautomate login Target App as an IDM Admin
echo "Logging into target application at ${epmurltarget}"
${epmautomatescript} login ${epmuserstarget} ${epmpwdtarget} $
{epmurltarget} ${epmidentitydomaintarget} ${proxyserverusername} $
{proxyserverpassword} ${proxyserverdomain}

for rolefile in role-*.csv
do
    rolenamecsv=$(echo "$rolefile" | cut -d'-' -f2)
    rolename=$(echo "$rolenamecsv" | cut -d'.' -f1)
    ${epmautomatescript} deletefile "${rolefile}" > /dev/null
2>&1
    echo "Uploading file ${rolefile}"
    ${epmautomatescript} uploadfile "${rolefile}"
    echo "Assigning roles"
    ${epmautomatescript} assignrole "${rolefile}" "${rolename}"
    ${epmautomatescript} deletefile "${rolefile}"
done

${epmautomatescript} logout
rm deletefile*.log > /dev/null 2>&1
}

init()
{
    # delete role-${role}.csv files
    for f in role-*.csv
    do
        rm "$f" > /dev/null 2>&1
    done
}

echo "Replicate predefined roles script started"
init
replicateroles
echo "Replicate predefined roles script completed"

```

2. Aktualisieren Sie `replicatepredefineroles.sh`. In der vorstehenden Tabelle finden Sie Informationen zu den Werten, die Sie angeben müssen. Darüber hinaus müssen Sie die Werte für diese Eigenschaften angeben:

- `javahome`: der absolute Pfad zu dem Verzeichnis, in dem Java installiert ist.
- `epmautomatescript`: Speicherort von `epmautomatescript.sh`; z.B. `epmautomatescript="/home/user1/epmautomate/bin/epmautomate.sh"`

Vierteljährliches Upgradeintervall für EPM Cloud erstellen

Verwenden Sie diese Skripte, um eine Selfservice-Lösung zum Überspringen von Updates zu erstellen, sodass Oracle Enterprise Performance Management Cloud-Umgebungen quartalsweise mit einem zweiwöchigen Testzyklus upgedatet werden. In diesem Fall werden Produktionsumgebungen zwei Wochen später als Testumgebungen upgedatet.

Mit diesem Skript können Sie auch jede andere monatliche Aktualisierung bei Bedarf überspringen. Standardmäßig führt EPM Cloud ein monatliches Update für Ihre Umgebungen durch. Mit dem Befehl `skipUpdate` überspringen Sie das Durchführen monatlicher Updates für eine Umgebung. Alternativ können Sie aktuelle Anfragen zum Überspringen von Updates anzeigen. Sie können das manuelle Ausführen der `skipUpdate`-Befehle automatisieren, indem Sie die Skripte in diesem Abschnitt verwenden. Diese Skripte automatisieren den Prozess zum Überspringen von Updates, sodass die Updates quartalsweise oder jeden zweiten Monat durchgeführt werden.

Note:

1. Sie können Updates nicht für mehr als zwei aufeinanderfolgende Monate überspringen. Beispiel: Das Skript gibt einen Fehler aus, wenn Sie versuchen, das Update für eine EPM Cloud-Umgebung nur im Februar, Juni und November durchzuführen.
2. Alle Updates, die in der Zwischenzeit aufgetreten sind, werden beim nächsten Update auf Ihre Umgebung angewendet. Beispiel: Angenommen, Sie verwenden dieses Skript, um quartalsweise Updates nur im Februar, Mai, August und November einzuplanen. In diesem Fall werden beispielsweise beim Update im Mai alle anwendbaren monatlichen EPM Cloud-Updates und Patches für Ihre Umgebung ausgeführt, die nach dem Update im Februar veröffentlicht wurden. Der Wartungsprozess kann mehr Zeit als üblich in Anspruch nehmen, wenn das Update durchgeführt wird.
3. Mit diesem Skript wird das Update-Intervall nur für ein Quartal festgelegt. Führen Sie dieses Skript auf monatlicher Basis aus, um sicherzustellen, dass das Update-Intervall für das gesamte Jahr konfiguriert wird.

- [Skript und Anweisungen für Windows](#)
- [Skript und Anweisungen für UNIX/Linux](#)
- [Groovy-Skript](#)

Skripte ausführen

1. So führen Sie die Skripte für Windows und Linux/UNIX aus:
 - a. Erstellen Sie die Datei `input.properties`, und aktualisieren Sie sie mit Informationen für Ihre Umgebung. Speichern Sie die Datei in einem lokalen Verzeichnis. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.

- b. Erstellen Sie `skip_update.ps1` (Windows PowerShell) oder das Bash-Skript `skip_update.sh` (Linux/UNIX), und speichern Sie es in dem Verzeichnis, in dem sich `input.properties` befindet.
- c. Starten Sie das Skript.
 - Linux/UNIX: Führen Sie `./skip_update.sh` aus.
 - Windows PowerShell: Führen Sie `skip_update.ps1` aus.
2. Um das Groovy-Skript auszuführen, verwenden Sie das Groovy-Fenster in einem EPM Cloud-Geschäftsprozess, oder automatisieren Sie die Skriptausführung mit [runBusinessRule](#). Informationen zur Ausführung des Groovy-Skripts mit EPM Automate finden Sie unter [Befehle ohne Installation von EPM Automate ausführen](#).

Skript und Anweisungen für Windows

Erstellen Sie `input.properties` und `skip_update.ps1`, indem Sie die Skripte in diesem Abschnitt kopieren.

1. Erstellen Sie `input.properties`, indem Sie das folgende Skript kopieren:

```
username=exampleAdmin
password=examplePassword.epw
url=exampleURL
updatemonths=02,05,08,11
```

2. Aktualisieren Sie `input.properties`, indem Sie Parameterwerte festlegen.

Table 3-11 Parameter für "input.properties"

Parameter	Beschreibung
<code>username</code>	Benutzername eines Serviceadministrators.
<code>password</code>	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
<code>url</code>	URL der Umgebung, in der Sie das nicht monatliche Update-Intervall festlegen möchten.
<code>updatemonths</code>	Eine durch Komma getrennte Liste der Monate, in denen Oracle Enterprise Performance Management Cloud-Updates auf die vom Parameter <code>url</code> angegebene Umgebung angewendet werden sollen. Beispiel: <code>updatemonths=02,05,08,11</code> . Monate müssen mit zwei Ziffern angegeben werden: 01 für Januar bis 12 für Dezember. Achten Sie darauf, für die Monate Januar bis September eine Null voranzustellen. Das Skript versucht, den Befehl skipUpdate für die Monate auszuführen, die nicht im Parameterwert <code>updatemonths</code> enthalten sind. Beispiel: Wenn Sie <code>updatemonths=02,05,08,11</code> festlegen, versucht das Skript, Kennzeichen zum Überspringen von Updates für Januar, März, April, Juni, Juli, September, Oktober und Dezember festzulegen, damit Updates nur im Februar, Mai, August und November durchgeführt werden.

3. Erstellen Sie skip_updates.ps1, indem Sie das folgende Skript kopieren:

```
# Skip Update PowerShell script

$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -
raw)
$username="$($inputproperties.username)"
$password="$($inputproperties.password)"
$url="$($inputproperties.url)"
$updatemonths="$($inputproperties.updatemonths)"

$monthsarr = ("01","02","03","04","05","06","07","08","09","10","11","12")
$global:monthsarrfromcurrent = @()
$global:yearsarrfromcurrent = @()
$updatemonthsarr = $updatemonths.Split(",")
$currentyear=Get-Date -Format yy
$currentmonth=Get-Date -Format MM
$nextyear=[int]$currentyear+1

function populateFromCurrentArrays() {
    $startposition = 0

    for ($i = 0; $i -le ($monthsarr.length - 1); $i++) {
        if (${currentmonth} -eq $monthsarr[$i]) {
            $startposition=$i
            break
        }
    }

    for ($i = 0; $i -le ($monthsarr.length - 1); $i++) {
        if (${i} -ge ${startposition}) {
            $global:monthsarrfromcurrent += $monthsarr[$i]
            $global:yearsarrfromcurrent += $currentyear
        }
    }

    for ($i = 0; $i -le ($monthsarr.length - 1); $i++) {
        if (${i} -lt ${startposition}) {
            $global:monthsarrfromcurrent += $monthsarr[$i]
            $global:yearsarrfromcurrent += $nextyear
        }
    }
}

function skipUpdateAdd($yearnumber, $monthnumber) {
    echo "Running: epmautomate.bat skipUpdate add version=${yearnumber}.${
monthnumber} comment=`"adding skipUpdate`""
    epmautomate skipUpdate add version=${yearnumber}.${monthnumber}
comment="adding skipUpdate"
}

function processSkipUpdates() {
    $addcount = 0

    echo "Running: epmautomate.bat login ${username} ${password} ${url}"
}
```

```

epmautomate login ${username} ${password} ${url}
echo "Running: epmautomate.bat skipUpdate remove"
epmautomate skipUpdate remove

for ($i = 0; $i -le ($global:monthsarrfromcurrent.length -
1); $i++) {
    $match = 1

    if (${addcount} -eq 2) {
        echo "Two skip update add calls have been made. No more
will be attempted."
        break
    }

    for ($j = 0; $j -le ($updatemonthsarr.length - 1); $j++) {
        if ($global:monthsarrfromcurrent[$i] -
eq $updatemonthsarr[$j]) {
            $match = 0
            break
        }
    }

    if (${match} -eq 1) {

skipUpdateAdd $global:yearsarrfromcurrent[$i] $global:monthsarrfromc
urrent[$i]
        $addcount += 1
    }
}

echo "Running: epmautomate.bat skipUpdate list"
epmautomate skipUpdate list
echo "Running: epmautomate.bat logout"
epmautomate logout
}

function compareUpdateMonths($thismonth, $nextmonth) {
    $nextmonthorig=${nextmonth}

    if (${nextmonth} -lt ${thismonth}) {
        $nextmonth+=12
    }

    $monthdiff = $nextmonth - $thismonth

    if (${monthdiff} -gt 3) {
        echo "There are more than 2 months skipped from month $
{thismonth} to month ${nextmonthorig}. Please correct updatemonths
in input.properties so that there are not more than two months
skipped between each update month. Exiting."
        exit 1
    }
}

function validateUpdateMonths() {

```

```

for ($i = 0; $i -le ($updateMonthsarr.length - 1); $i++) {
    $nextint = $i + 1
    $thisupdateMonth = $updateMonthsarr[$i]
    $thisupdateMonthint=[int]$thisupdateMonth
    $nextupdateMonth=$updateMonthsarr[$nextint]
    $nextupdateMonthint=[int]$nextupdateMonth

    if ($nextupdateMonth -eq "") {
        $nextupdateMonth=$updateMonthsarr[0]
        $nextupdateMonthint=[int]$nextupdateMonth
    }

    compareUpdateMonths $thisupdateMonthint $nextupdateMonthint
}

}

validateUpdateMonths
populateFromCurrentArrays
processSkipUpdates

```

Skript und Anweisungen für UNIX/Linux

Erstellen Sie `input.properties` und `skip_update.sh`, indem Sie die Skripte in diesem Abschnitt kopieren.

1. Erstellen Sie `input.properties`, indem Sie das folgende Skript kopieren:

```

javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
password=examplePassword.epw
url=exampleURL
updateMonths=02,05,08,11

```

2. Aktualisieren Sie `input.properties`, indem Sie Parameterwerte festlegen.

Table 3-12 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis JAVA_HOME.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (<code>epmautomate.sh</code>).
username	Benutzername eines Serviceadministrators.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
url	URL der Umgebung, in der Sie das nicht monatliche Update-Intervall festlegen möchten.

Table 3-12 (Cont.) Parameter für "input.properties"

Parameter	Beschreibung
updatemonths	Eine durch Komma getrennte Liste der Monate, in denen Oracle Enterprise Performance Management Cloud-Updates auf die vom Parameter <code>url</code> angegebene Umgebung angewendet werden sollen. Beispiel: <code>updatemonths=02,05,08,11</code> . Monate müssen mit zwei Ziffern angegeben werden. Stellen Sie für die Monate Januar bis September eine Null voran. Das Skript versucht, den Befehl <code>skipUpdate</code> für die Monate auszuführen, die nicht im Parameterwert <code>updatemonths</code> enthalten sind. Beispiel: Wenn Sie <code>updatemonths=02,05,08,11</code> festlegen, versucht das Skript, Kennzeichen zum Überspringen von Updates für Januar, März, April, Juni, Juli, September, Oktober und Dezember festzulegen, damit Updates nur im Februar, Mai, August und November durchgeführt werden.

3. Erstellen Sie `skip_updates.sh`, indem Sie das folgende Skript kopieren:

```
#!/bin/sh

./input.properties
export JAVA_HOME=${javahome}

declare -a monthsarr=(01 02 03 04 05 06 07 08 09 10 11 12)
declare -a monthsarrfromcurrent
declare -a yearsarrfromcurrent
updatemonthsarr=( $(echo "${updatemonths}" | sed 's/,/ /g') )
currentyear=$(date +%y)
nextyear=$((currentyear+1))
currentmonth=$(date +%m)

populateFromCurrentArrays() {
    for i in ${!monthsarr[@]}
    do
        if [[ "${currentmonth}" == "${monthsarr[$i]}" ]]
        then
            startposition=$i
            break
        fi
    done

    for i in ${!monthsarr[@]}
    do
        if [[ $i -ge $startposition ]]
        then
            monthsarrfromcurrent=("${monthsarrfromcurrent[@]}" "${monthsarr[$i]}")
            yearsarrfromcurrent=("${yearsarrfromcurrent[@]}" "${currentyear}")
        fi
    done
}
```

```

    for i in ${!monthsarr[@]}
    do
        if [[ ${i} -lt ${startposition} ]]
        then
            monthsarrfromcurrent=("${monthsarrfromcurrent[@]}" "$
{monthsarr[$i]}")
            yearsarrfromcurrent=("${yearsarrfromcurrent[@]}" "$
{nextyear}")
            fi
        done
    }

skipUpdateAdd() {
    local yearnumber="$1"
    local monthnumber="$2"

    echo "Running: ${epmautomatescript} skipUpdate add version=$
{yearnumber}.${monthnumber} comment=\"adding skipUpdate\""
    ${epmautomatescript} skipUpdate add version=${yearnumber}.${
monthnumber} comment="adding skipUpdate"
}

processSkipUpdates() {
    local addcount=0

    echo "Running: ${epmautomatescript} login ${username} ${password} $
{url}"
    ${epmautomatescript} login ${username} ${password} ${url}
    echo "Running: ${epmautomatescript} skipUpdate remove"
    ${epmautomatescript} skipUpdate remove

    for i in ${!monthsarrfromcurrent[@]}
    do
        local match=1

        if [[ ${addcount} -eq 2 ]]
        then
            echo "Two skip update add calls have been made. No more will
be attempted."
            break
        fi

        for j in ${!updatemonthsarr[@]}
        do
            if [[ "${monthsarrfromcurrent[$i]}" == "$
{updatemonthsarr[$j]}" ]]
            then
                match=0
                break
            fi
        done

        if [[ ${match} -eq 1 ]]
        then
            skipUpdateAdd ${yearsarrfromcurrent[$i]} "$

```



```

{monthsarrfromcurrent[$i]}"
        addcount=$((addcount+1))
    fi
done

echo "Running: ${epmautomatescript} skipUpdate list"
${epmautomatescript} skipUpdate list
echo "Running: ${epmautomatescript} logout"
${epmautomatescript} logout
}

compareUpdateMonths() {
    local thismonth=$1
    local nextmonth=$2
    local nextmonthorig=${nextmonth}

    if [[ ${nextmonth} -lt ${thismonth} ]]
    then
        nextmonth=$((nextmonth+12))
    fi

    monthdiff=$((nextmonth-thismonth))

    if [[ ${monthdiff} -gt 3 ]]
    then
        echo "There are more than 2 months skipped from month $
{thismonth} to month ${nextmonthorig}. Please correct updatemonths
in input.properties so that there are not more than two months
skipped between each update month. Exiting."
        exit 1
    fi
}

validateUpdateMonths() {
    for i in ${!updatemonthsarr[@]}
    do
        nextint=$((i+1))
        thisupdatemonth="${updatemonthsarr[$i]}"
        thisupdatemonthint=${thisupdatemonth#0}
        nextupdatemonth="${updatemonthsarr[$nextint]}"
        nextupdatemonthint=${nextupdatemonth#0}

        if [[ ${nextupdatemonth} == "" ]]
        then
            nextupdatemonth="${updatemonthsarr[0]}"
            nextupdatemonthint=${nextupdatemonth#0}
        fi

        compareUpdateMonths ${thisupdatemonthint} $
{nextupdatemonthint}
    done
}

validateUpdateMonths
populateFromCurrentArrays

```

processSkipUpdates

Groovy-Skript

Wenn Kennwörter Sonderzeichen enthalten, finden Sie weitere Informationen unter [Sonderzeichen verarbeiten](#). Stellen Sie zudem sicher, dass die folgenden Parameterwerte so ersetzt werden, dass Sie zu Ihren Umgebungen passen:

Table 3-13 Zu ändernde Parameter

Parameter	Beschreibung
user	Benutzername eines Serviceadministrators.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
url	URL der Umgebung, in der Sie das nicht monatliche Update-Intervall festlegen möchten.
updatemonths	Eine durch Komma getrennte Liste der Monate, in denen Oracle Enterprise Performance Management Cloud-Updates auf die vom Parameter <code>url</code> angegebene Umgebung angewendet werden sollen. Beispiel: <code>updatemonths=02,05,08,11</code> . Monate müssen mit zwei Ziffern angegeben werden: 01 für Januar bis 12 für Dezember. Achten Sie darauf, für die Monate Januar bis September eine Null voranzustellen. Das Skript versucht, den Befehl <code>skipUpdate</code> für die Monate auszuführen, die nicht im Parameterwert <code>updatemonths</code> enthalten sind. Beispiel: Wenn Sie <code>updatemonths=02,05,08,11</code> festlegen, versucht das Skript, Kennzeichen zum Überspringen von Updates für Januar, März, April, Juni, Juli, September, Oktober und Dezember festzulegen, damit Updates nur im Februar, Mai, August und November durchgeführt werden.

```
import java.text.SimpleDateFormat

String user = 'service_administrator'
String password = 'examplePWD'
String url = 'example_EPM_URL'
String updatemonths = '02,05,08,11'

def currentdate = new Date()
def yf = new SimpleDateFormat("yy")
def mf = new SimpleDateFormat("MM")
String[] monthsarr = ["01", "02", "03", "04", "05", "06", "07", "08", "09",
"10", "11", "12"]
List<String> monthsarrfromcurrent = new ArrayList<>()
List<String> yearsarrfromcurrent = new ArrayList<>()
String currentyear = yf.format(currentdate)
String nextyear = (currentyear.toInteger() + 1).toString()
String currentmonth = mf.format(currentdate)

String[] updateMonthsStringArr = updatemonths.split(',');
```

```

def updatemonthsarr = new int[updateMonthsStringArr.length];
for(int i = 0; i < updateMonthsStringArr.length; i++)
{
    updatemonthsarr[i] = Integer.parseInt(updateMonthsStringArr[i]);
}

def LogMessage(String message) {
    def date = new Date()
    def sdf = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss")
    println('[' + sdf.format(date) + '][GROOVY] ' + message);
}

def LogOperationStatus(EpmAutomateStatus opstatus) {
    def returncode = opstatus.getStatus()
    if (returncode != 0){
        LogMessage(opstatus.getOutput())
    }
    LogMessage('return code: ' + returncode)
}

int CompareUpdateMonths(int thismonth, int nextmonth) {
    int nextmonthorig = nextmonth

    if (nextmonth < thismonth) {
        nextmonth = nextmonth + 12
    }

    int monthdiff = nextmonth - thismonth

    if (monthdiff > 3) {
        LogMessage('There are more than 2 months skipped from month '
+ thismonth + ' to month ' + nextmonthorig + '. Please correct
updateMonths so that there are not more than two months skipped
between each update month. Exiting.')
        return 1
    }

    return 0
}

int ValidateUpdateMonths(int[] updatemonthsarr) {
    for(int i = 0; i < updatemonthsarr.length; i++)
    {
        int nextint = i + 1
        String nextupdatemonth = ""
        int nextupdatemonthint = 0
        String thisupdatemonth = updatemonthsarr[i]
        int thisupdatemonthint = thisupdatemonth.toInteger()

        if (nextint < updatemonthsarr.length) {
            nextupdatemonth = updatemonthsarr[nextint]
        } else {
            nextupdatemonth = updatemonthsarr[0]
        }
    }
}

```

```

        nextupdatemonthhint = nextupdatemonth.toInteger()

        int returncode = CompareUpdateMonths(thisupdatemonthhint,
nextupdatemonthhint)
        if (returncode > 0) {
            return 1
        }
    }
    return 0
}

def SkipUpdateAdd(EpmAutomate automate, String yearnumber, String
monthnumber) {
    String yeardotmonth = yearnumber + '.' + monthnumber
    LogMessage('Running: epmautomate skipUpdate add version=' + yeardotmonth
+ ' comment=\"adding skipUpdate\"')
    EpmAutomateStatus status =
automate.execute('skipupdate', 'add', 'version=' +
yeardotmonth, 'comment=\"adding skipUpdate\"')
    LogOperationStatus(status)
}

LogMessage('Starting skip update processing')
EpmAutomate automate = getEpmAutomate()

// validate update months
int returncode = ValidateUpdateMonths(updatemonthsarr)
if (returncode != 0) {
    return 1
}

// populate arrays
int startposition = 0
for(int i = 0; i < monthsarr.length; i++)
{
    if (currentmonth == monthsarr[i]) {
        startposition = i
        break
    }
}

for(int i = 0; i < monthsarr.length; i++)
{
    if (i >= startposition) {
        monthsarrfromcurrent.add(monthsarr[i])
        yearsarrfromcurrent.add(currentyear)
    }
}

for(int i = 0; i < monthsarr.length; i++)
{
    if (i <= startposition) {
        monthsarrfromcurrent.add(monthsarr[i])
        yearsarrfromcurrent.add(nextyear)
    }
}

```

```

}

// process skip updates
LogMessage("Operation: encrypt " + password + " oracleKey
password.epw")
EpmAutomateStatus status =
automate.execute('encrypt',password,"oracleKey","password.epw")
LogOperationStatus(status)

LogMessage("Operation: login " + user + " password.epw " + url)
status = automate.execute('login',user,"password.epw",url)
LogOperationStatus(status)

LogMessage('Running: epmautomate skipUpdate remove')
status = automate.execute('skipupdate','remove')
LogOperationStatus(status)

int addcount = 0

for (int i = 0; i < monthsarrfromcurrent.size(); i++) {
    int match = 1

    if (addcount == 2){
        LogMessage('Two skip update add calls have been made. No more
will be attempted.')
        break
    }

    for(int j = 0; j < updatemonthsarr.length; j++) {

        if (Integer.parseInt(monthsarrfromcurrent.get(i)) ==
updatemonthsarr[j]) {
            match = 0
            break
        }
    }

    if (match == 1) {
        SkipUpdateAdd(automate, yearsarrfromcurrent.get(i),
monthsarrfromcurrent.get(i))
        addcount+=1
    }
}

LogMessage('Running: epmautomate skipUpdate list')
status = automate.execute('skipupdate','list')
LogOperationStatus(status)

LogMessage('Running: epmautomate logout')
status = automate.execute('logout')
LogOperationStatus(status)

LogMessage('Skip update processing completed')

```

Vierteljährliches Upgradeintervall für EPM Cloud mit sechswöchigen Testzyklen erstellen

Verwenden Sie das Skript in diesem Abschnitt, um eine Selfservice-Lösung zum Überspringen von Updates zu erstellen, sodass Oracle Enterprise Performance Management Cloud-Umgebungen quartalsweise mit einem sechswöchigen Testzyklus upgedatet werden. In diesem Fall werden Produktionsumgebungen sechs Wochen später als die Testumgebungen upgedatet.

Standardmäßig führt EPM Cloud ein monatliches Update für Ihre Umgebungen durch. Mit dem Befehl `skipUpdate` überspringen Sie das Durchführen monatlicher Updates für eine Umgebung. Alternativ können Sie aktuelle Anfragen zum Überspringen von Updates anzeigen. Sie können das manuelle Ausführen der `skipUpdate`-Befehle automatisieren, indem Sie die Skripte in diesem Abschnitt verwenden. Diese Skripte automatisieren den Prozess zum Überspringen von Updates, sodass die Updates quartalsweise mit einem sechswöchigen Testzyklus durchgeführt werden.

Note:

1. Sie können Updates nicht für mehr als drei aufeinanderfolgende Monate überspringen. Dieses Skript gibt einen Fehler aus, wenn Sie versuchen, das Update für eine EPM Cloud-Umgebung nur im Februar und im Oktober durchzuführen.
2. Alle Updates, die in der Zwischenzeit aufgetreten sind, werden beim nächsten Update auf Ihre Umgebung angewendet. Beispiel: Angenommen, Sie verwenden dieses Skript, um Updates vierteljährlich nur für Februar, Mai, August und November einzuplanen. In diesem Fall werden beispielsweise beim Update im Mai alle anwendbaren monatlichen EPM Cloud-Updates und Patches für Ihre Umgebung ausgeführt, die nach dem Update im Februar veröffentlicht wurden. Der Wartungsprozess kann mehr Zeit als üblich in Anspruch nehmen, wenn das Update durchgeführt wird.
3. Mit diesem Skript wird das Update-Intervall nur für ein Quartal festgelegt. Beispielszenario: Der Updatezyklus der Testumgebung wird als erster Freitag im Februar (Update 24.02), Mai (Update 24.05), August (Update 24.08) und November (Update 24.11) eingerichtet. Die Produktionsumgebung wird am dritten Freitag im März (Update 24.02) mit der Version aktualisiert, die verwendet wurde, um die Testumgebung am ersten Freitag im Februar zu aktualisieren (Update 24.02). Ähnliche Updates der Produktionsumgebung erfolgen in der dritten Woche im Juni (Update 24.05), September (Update 24.08) und Dezember (Update 24.11). In diesem Szenario werden die Produktionsumgebungen nicht auf das neueste verfügbare Update aktualisiert, sondern auf das aktuelle Update der Testumgebung.

Windows-Beispielskript

Erstellen Sie die Datei `skip_update.ps1`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis. Informationen zum Ausführen dieses Skriptes finden Sie unter [Skripte ausführen](#).

```
# Skip Update PowerShell script

$inputproperties = ConvertFrom-StringData(Get-Content ./
input.properties -raw)
$username="$($inputproperties.username)"
$password="$($inputproperties.password)"
$url="$($inputproperties.url)"
$updateversions="$($inputproperties.updateversions)"
$podtype="$($inputproperties.podtype)"
$proxyserverusername="$($inputproperties.proxyserverusername)"
$proxyserverpassword="$($inputproperties.proxyserverpassword)"
$proxyserverdomain="$($inputproperties.proxyserverdomain)"

echo "Starting skip_update.ps1 script."

$monthsarr =
("01","02","03","04","05","06","07","08","09","10","11","12")
$global:monthsarrfromcurrent = @()
$global:yearsarrfromcurrent = @()
$updateversionsarr = $updateversions.Split(",")
$currentyear=Get-Date -Format yy
$currentmonth=Get-Date -Format MM
$nextyear=[int]$currentyear+1

function populateFromCurrentArrays() {
    $startposition = 0

    for ($i = 0; $i -le ($monthsarr.length - 1); $i++) {
        if (${currentmonth} -eq $monthsarr[$i]) {
            if (${podtype} -eq "prod") {
                if (${updateversionsarr} -contains ${currentmonth}) {
                    $startposition=$i-2
                } else {
                    $startposition=$i-1
                }
            } else {
                if (${updateversionsarr} -contains ${currentmonth}) {
                    $startposition=$i
                } else {
                    $startposition=$i-1
                }
            }
        }
        break
    }

    if (${startposition} -lt 0) {
        $startposition=$startposition+12
    }
}
```

```

    }

    for ($i = 0; $i -le ($monthsarr.length - 1); $i++) {
    if (${i} -ge ${startposition}) {
        $global:monthsarrfromcurrent += $monthsarr[$i]
        $global:yearsarrfromcurrent += $currentyear
    }
    }

    for ($i = 0; $i -le ($monthsarr.length - 1); $i++) {
    if (${i} -lt ${startposition}) {
        $global:monthsarrfromcurrent += $monthsarr[$i]
        $global:yearsarrfromcurrent += $nextyear
    }
    }
}

function skipUpdateAdd($yearnumber, $monthnumber) {
    echo "Running: epmautomate.bat skipUpdate add version=${yearnumber}.${
monthnumber} comment=`"adding skipUpdate`""
    epmautomate skipUpdate add version=${yearnumber}.${monthnumber}
comment="adding skipUpdate"
}

function processSkipUpdates() {
    $addcount = 0
    $countlimit = 0

    if (${podtype} -eq "prod") {
        $countlimit = 3
    } else {
        $countlimit = 2
    }

    if ((${proxyserverusername} -eq "") -And (${proxyserverpassword} -eq "")
-And (${proxyserverdomain} -eq "")) {
        echo "Running: epmautomate.bat login ${username} ${password} ${url}"
        epmautomate login ${username} ${password} ${url}
    } else {
        echo "Running: epmautomate.bat login ${username} ${password} ${url}
ProxyServerUserName=${proxyserverusername} ProxyServerPassword=${
proxyserverpassword} ProxyServerDomain=${proxyserverdomain}"
        epmautomate login ${username} ${password} ${url}
ProxyServerUserName=${proxyserverusername} ProxyServerPassword=${
proxyserverpassword} ProxyServerDomain=${proxyserverdomain}
    }

    echo "Running: epmautomate.bat skipUpdate remove"
    epmautomate skipUpdate remove

    for ($i = 0; $i -le ($global:monthsarrfromcurrent.length - 1); $i++) {
        $match = 1

        if (${addcount} -eq ${countlimit}) {
            echo "Update calls are completed. No more will be attempted."

```



```

        break
    }

    for ($j = 0; $j -le ($updateversionsarr.length - 1); $j++) {
        if (($currentmonth -eq $updateversionsarr[$j]) -And ($
{addcount} -gt 0)) {
            $match = 1
            break
        }

        if (($global:monthsarrfromcurrent[$i] -
eq $updateversionsarr[$j]) -And (${addcount} -eq 0)){
            $match = 0
            break
        }
    }

    if (${match} -eq 1) {

skipUpdateAdd $global:yearsarrfromcurrent[$i] $global:monthsarrfromcurr
ent[$i]
        $addcount += 1
    }
}

echo "Running: epmautomate.bat skipUpdate list"
epmautomate skipUpdate list
echo "Running: epmautomate.bat logout"
epmautomate logout
}

function compareUpdateMonths($thismonth, $nextmonth) {
    $nextmonthorig=${nextmonth}

    if (${nextmonth} -lt ${thismonth}) {
        $nextmonth+=12
    }

    $monthdiff = $nextmonth - $thismonth

    if (${monthdiff} -gt 4) {
        echo "There are more than 3 versions skipped from version $
{thismonth} to version ${nextmonthorig}. Please correct updateversions
in input.properties so that there are not more than three versions
skipped between each update version. Exiting."
        exit 1
    }
}

function validateUpdateVersions() {
    for ($i = 0; $i -le ($updateversionsarr.length - 1); $i++) {
        $nextint = $i + 1
        $thisupdatemonth = $updateversionsarr[$i]
        $thisupdatemonthint=[int]$thisupdatemonth
        $nextupdatemonth=$updateversionsarr[$nextint]
    }
}

```

```

    $nextupdatemonthhint=[int]$nextupdatemonth

    if (${nextupdatemonth} -eq "") {
        $nextupdatemonth=$updateversionsarr[0]
        $nextupdatemonthhint=[int]$nextupdatemonth
    }

    compareUpdateMonths $thisupdatemonthhint $nextupdatemonthhint
}

}

validateUpdateVersions
populateFromCurrentArrays
processSkipUpdates

```

Linux-/UNIX-Beispielskript

Erstellen Sie die Datei `skip_update.sh`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis. Informationen zum Ausführen dieses Skriptes finden Sie unter [Skripte ausführen](#).

```

#!/bin/sh

. ./input.properties

echo "Starting skip_update.sh script."

export JAVA_HOME=${javahome}

declare -a monthsarr=(01 02 03 04 05 06 07 08 09 10 11 12)
declare -a monthsarrfromcurrent
declare -a yearsarrfromcurrent
updateversionsarr=( $(echo "${updateversions}" | sed 's/,/ /g') )
currentyear=$(date +%y)
nextyear=$((currentyear+1))
currentmonth=$(date +%m)

populateFromCurrentArrays() {
    local startposition=0

    for i in ${!monthsarr[@]}
    do
        if [[ "${currentmonth}" == "${monthsarr[$i]}" ]]
        then
            if [[ "${podtype}" == "prod" ]]
            then
                if [[ ${updateversionsarr[@]} =~ ${currentmonth} ]]
                then
                    startposition=$((i-2))
                else
                    startposition=$((i-1))
                fi
            break
            else
                if [[ ${updateversionsarr[@]} =~ ${currentmonth} ]]

```

```

        then
            startposition=$i
        else
            startposition=$((i-1))
        fi
        break
    fi
done

if [[ ${startposition} -lt 0 ]]
then
    startposition=$((startposition+12))
fi

for i in ${!monthsarr[@]}
do
    if [[ ${i} -ge ${startposition} ]]
    then
        monthsarrfromcurrent=("${monthsarrfromcurrent[@]}" "$
monthsarr[$i]")
        yearsarrfromcurrent=("${yearsarrfromcurrent[@]}" "$
currentyear")
    fi
done

for i in ${!monthsarr[@]}
do
    if [[ ${i} -lt ${startposition} ]]
    then
        monthsarrfromcurrent=("${monthsarrfromcurrent[@]}" "$
monthsarr[$i]")
        yearsarrfromcurrent=("${yearsarrfromcurrent[@]}" "$
nextyear")
    fi
done
}

skipUpdateAdd() {
    local yearnumber="$1"
    local monthnumber="$2"

    echo "Running: ${epmautomatescript} skipUpdate add version=$
{yearnumber}.${monthnumber} comment=\"adding skipUpdate\"
    ${epmautomatescript} skipUpdate add version=${yearnumber}.${
monthnumber} comment=\"adding skipUpdate"
}

processSkipUpdates() {
    local addcount=0
    local countlimit=0

    if [[ "${podtype}" == "prod" ]]
    then
        countlimit=3
    fi
}

```

```

else
    countlimit=2
fi

if [[ "${proxyserverusername}" == "" ]] && [[ "${proxyserverpassword}"
== "" ]] && [[ "${proxyserverdomain}" == "" ]]
then
    echo "Running: ${epmautomatescript} login ${username} ${password} ${
url}"
    ${epmautomatescript} login ${username} ${password} ${url}
else
    echo "Running: ${epmautomatescript} login ${username} ${password} ${
url} ProxyServerUserName=${proxyserverusername} ProxyServerPassword=${
proxyserverpassword} ProxyServerDomain=${proxyserverdomain}"
    ${epmautomatescript} login ${username} ${password} ${url}
ProxyServerUserName=${proxyserverusername} ProxyServerPassword=${
proxyserverpassword} ProxyServerDomain=${proxyserverdomain}
fi
echo "Running: ${epmautomatescript} skipUpdate remove"
${epmautomatescript} skipUpdate remove

for i in ${!monthsarrfromcurrent[@]}
do
    local match=1

    if [[ ${addcount} -eq ${countlimit} ]]
    then
        echo "Update add calls are completed. No more will be attempted."
        break
    fi

    for j in ${!updateversionsarr[@]}
    do
        if [[ "${currentmonth}" == "${updateversionsarr[$j]}" ]] && [[ $
addcount} -gt 0 ]]
        then
            match=1
            break
        fi

        if [[ "${monthsarrfromcurrent[$i]}" == "${
updateversionsarr[$j]}" ]] && [[ ${addcount} -eq 0 ]]
        then
            match=0
            break
        fi
    done

    if [[ ${match} -eq 1 ]]
    then
        skipUpdateAdd ${yearsarrfromcurrent[$i]} "${
monthsarrfromcurrent[$i]}"
        addcount=$((addcount+1))
    fi
done

```

```

    echo "Running: ${epmautomatescript} skipUpdate list"
    ${epmautomatescript} skipUpdate list
    echo "Running: ${epmautomatescript} logout"
    ${epmautomatescript} logout
}

compareUpdateMonths() {
    local thismonth=$1
    local nextmonth=$2
    local nextmonthorig=${nextmonth}

    if [[ ${nextmonth} -lt ${thismonth} ]]
    then
        nextmonth=$((nextmonth+12))
    fi

    monthdiff=$((nextmonth-thismonth))

    if [[ ${monthdiff} -gt 4 ]]
    then
        echo "There are more than 3 versions skipped from version $
        {thismonth} to version ${nextmonthorig}. Please correct updateversions
        in input.properties so that there are not more than three versions
        skipped between each update version. Exiting."
        exit 1
    fi
}

validateUpdateVersions() {
    for i in ${!updateversionsarr[@]}
    do
        nextint=$((i+1))
        thisupdatemonth="${updateversionsarr[$i]}"
        thisupdatemonthhint=${thisupdatemonth#0}
        nextupdatemonth="${updateversionsarr[$nextint]}"
        nextupdatemonthhint=${nextupdatemonth#0}

        if [[ ${nextupdatemonth} == "" ]]
        then
            nextupdatemonth="${updateversionsarr[0]}"
            nextupdatemonthhint=${nextupdatemonth#0}
        fi

        compareUpdateMonths ${thisupdatemonthhint} ${nextupdatemonthhint}
    done
}

validateUpdateVersions
populateFromCurrentArrays
processSkipUpdates

```

Serverseitiges Groovy-Skript

Erstellen Sie das Groovy-Skript `skip_update.groovy`, indem Sie das folgende Skript kopieren und anschließend aktualisieren. Informationen zum Ausführen dieses Skriptes finden Sie unter [Skripte ausführen](#).

Aktualisieren Sie in diesem Groovy-Skript die folgenden Variablen:

- `username` - Benutzername eines Serviceadministrators für die Umgebung, für die Sie das nicht monatliche Update-Intervall festlegen möchten.
- `password` - Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
- `url` - URL der Umgebung, für die Sie das nicht monatliche Update-Intervall festlegen möchten.
- `updateversions` - Eine durch Komma getrennte Liste der EPM Cloud-Updates, die auf die durch den Parameter `url` identifizierte Umgebung angewendet werden sollen.
Beispiel: `updateversions=02,05,08,11`.
Versionen müssen mit zwei Ziffern angegeben werden. Stellen Sie für die Updates 01 bis 09 eine Null voran. Das Skript versucht, den Befehl [skipUpdate](#) für die Updates auszuführen, die nicht im Parameterwert `updateversions` enthalten sind. Beispiel: Wenn Sie `updateversions=02,05,08,11` festlegen, versucht das Skript, Kennzeichen zum Überspringen von Updates für 01 (Januar), 03 (März), 04 (April), 06 (Juni), 07 (Juli), 09 (September), 10 (Oktober) und 12 (Dezember) festzulegen. In diesem Fall werden die EPM Cloud-Updates 02 (Februar), 05 (Mai), 08 (August) und 11 (November) auf die Umgebung angewendet.
- `podtype` - EPM Cloud-Umgebungstyp. Gültige Werte sind `test` und `prod`.
- `proxyserverusername` - Der Benutzername zum Authentifizieren einer sicheren Session auf dem Proxyserver, der den Zugriff auf das Internet steuert.
- `proxyserverpassword` - Das Kennwort zum Authentifizieren des Benutzers beim Proxyserver.
- `proxyserverdomain` - Der Name der Domain, die für den Proxyserver definiert ist.



Note:

Wenn Sie keinen Proxyserver verwenden, geben Sie keine Werte für die Parameter `proxyserverusername`, `proxyserverpassword` und `proxyserverdomain` an.

```
import java.text.SimpleDateFormat

String username = 'service_administrator'
String password = 'examplePWD'
String url = 'example_EPM_URL'
String updateversions = '01,04,07,10'
String podtype = 'test'
String proxyserverusername = ''
String proxyserverpassword = ''
String proxyserverdomain = ''

def currentdate = new Date()
```

```

def yf = new SimpleDateFormat("yy")
def mf = new SimpleDateFormat("MM")
String[] monthsarr = ["01", "02", "03", "04", "05", "06", "07", "08",
"09", "10", "11", "12"]
List<String> monthsarrfromcurrent = new ArrayList<>()
List<String> yearsarrfromcurrent = new ArrayList<>()
String currentyear = yf.format(currentdate)
String nextyear = (currentyear.toInteger() + 1).toString()
String currentmonth = mf.format(currentdate)

String[] updateVersionsStringArr = updateversions.split(',');
def updateversionsarr = new int[updateVersionsStringArr.length];
for(int i = 0; i < updateVersionsStringArr.length; i++)
{
    updateversionsarr[i] =
Integer.parseInt(updateVersionsStringArr[i]);
}

def LogMessage(String message) {
    def date = new Date()
    def sdf = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss")
    println('[' + sdf.format(date) + '][GROOVY] ' + message);
}

def LogOperationStatus(EpmAutomateStatus opstatus) {
    def returncode = opstatus.getStatus()
    LogMessage(opstatus.getOutput())
    LogMessage('return code: ' + returncode)
}

int CompareUpdateMonths(int thismonth, int nextmonth) {
    int nextmonthorig = nextmonth

    if (nextmonth < thismonth) {
        nextmonth = nextmonth + 12
    }

    int monthdiff = nextmonth - thismonth

    if (monthdiff > 4) {
        LogMessage('There are more than 3 versions skipped from
version ' + thismonth + ' to version ' + nextmonthorig + '. Please
correct updateversions so that there are not more than three versions
skipped between each update version. Exiting.')
        return 1
    }

    return 0
}

int ValidateUpdateMonths(int[] updateversionsarr) {
    for(int i = 0; i < updateversionsarr.length; i++)
    {
        int nextint = i + 1
        String nextupdatemonth = ""
    }
}

```

```

        int nextupdatemonthint = 0
        String thisupdatemonth = updateversionsarr[i]
        int thisupdatemonthint = thisupdatemonth.toInteger()

        if (nextint < updateversionsarr.length) {
            nextupdatemonth = updateversionsarr[nextint]
        } else {
            nextupdatemonth = updateversionsarr[0]
        }

        nextupdatemonthint = nextupdatemonth.toInteger()

        int returncode = CompareUpdateMonths(thisupdatemonthint,
nextupdatemonthint)
        if (returncode > 0) {
            return 1
        }
    }
    return 0
}

def SkipUpdateAdd(EpmAutomate automate, String yearnumber, String
monthnumber) {
    String yeardotmonth = yearnumber + '.' + monthnumber
    LogMessage('Running: epmautomate skipUpdate add version=' + yeardotmonth
+ ' comment=\"adding skipUpdate\"')
    EpmAutomateStatus status =
automate.execute('skipupdate', 'add', 'version=' +
yeardotmonth, 'comment=\"adding skipUpdate\"')
    LogOperationStatus(status)
}

LogMessage('Starting skip update processing')
EpmAutomate automate = getEpmAutomate()

// validate update months
int returncode = ValidateUpdateMonths(updateversionsarr)
if (returncode != 0) {
    return 1
}

// populate arrays
int startposition = 0
for(int i = 0; i < monthsarr.length; i++)
{
    if (currentmonth == monthsarr[i]) {
        if (podtype.equals("prod")) {
            if (updateVersionsStringArr.contains(currentmonth)) {
                startposition = (i-2)
            } else {
                startposition = (i-1)
            }
        }
    } else {
        if (updateVersionsStringArr.contains(currentmonth)) {
            startposition = i
        }
    }
}

```



```

        } else {
            startposition = (i-1)
        }
    }

    break
}
}

if (startposition < 0) {
    startposition = startposition + 12
}

for(int i = 0; i < monthsarr.length; i++)
{
    if (i >= startposition) {
        monthsarrfromcurrent.add(monthsarr[i])
        yearsarrfromcurrent.add(currentyear)
    }
}

for(int i = 0; i < monthsarr.length; i++)
{
    if (i <= startposition) {
        monthsarrfromcurrent.add(monthsarr[i])
        yearsarrfromcurrent.add(nextyear)
    }
}

// process skip updates
LogMessage("Operation: encrypt " + password + " oracleKey
password.epw")
EpmAutomateStatus status =
automate.execute('encrypt',password,"oracleKey","password.epw")
LogOperationStatus(status)

if ((proxyserverusername != null && proxyserverusername != '') &&
(proxyserverpassword != null && proxyserverpassword != '') &&
(proxyserverdomain != null && proxyserverdomain != '')) {
    LogMessage("Operation: login " + username + " password.epw " + url
+ " ProxyServerUserName=" + proxyserverusername + "
ProxyServerPassword=" + proxyserverpassword + " ProxyServerDomain=" +
proxyserverdomain)
    status =
automate.execute('login',username,"password.epw",url,"ProxyServerUserNa
me=" + proxyserverusername,"ProxyServerPassword=" +
proxyserverpassword,"ProxyServerDomain=" + proxyserverdomain)
    LogOperationStatus(status)
} else {
    LogMessage("Operation: login " + username + " password.epw " + url)
    status = automate.execute('login',username,"password.epw",url)
    LogOperationStatus(status)
}
LogMessage('Running: epmautomate skipUpdate remove')
status = automate.execute('skipupdate','remove')

```

```

LogOperationStatus(status)

int addcount = 0
int countlimit = 0

if (podtype.equals("prod")) {
    countlimit = 3
} else {
    countlimit = 2
}

for (int i = 0; i < monthsarrfromcurrent.size(); i++) {
    int match = 1

    if (addcount == countlimit){
        LogMessage('Update add calls are completed. No more will be
attempted.')
        break
    }

    for(int j = 0; j < updateversionsarr.length; j++) {

        if ((Integer.parseInt(currentmonth) == updateversionsarr[j]) &&
(addcount > 0)) {
            match = 1
            break
        }

        if ((Integer.parseInt(monthsarrfromcurrent.get(i)) ==
updateversionsarr[j]) && (addcount == 0)) {
            match = 0
            break
        }
    }

    if (match == 1) {
        SkipUpdateAdd(automate, yearsarrfromcurrent.get(i),
monthsarrfromcurrent.get(i))
        addcount+=1
    }
}

LogMessage('Running: epmautomate skipUpdate list')
status = automate.execute('skipupdate','list')
LogOperationStatus(status)
println(status.getItemsList())

LogMessage('Running: epmautomate logout')
status = automate.execute('logout')
LogOperationStatus(status)

LogMessage('Skip update processing completed')

```

Datei "input.properties" zum Ausführen der skip_update-Skripte für Windows und Linux/UNIX erstellen

Um die Datei skip_update.ps1 oder skip_update.sh auszuführen, erstellen Sie die Datei input.properties, und aktualisieren Sie sie mit den Informationen für Ihre Umgebung. Speichern Sie die Datei in einem lokalen Verzeichnis. Der Inhalt dieser Datei unterscheidet sich je nach Betriebssystem.

Windows

```
username=exampleAdmin
password=examplePassword.epw
url=exampleURL
updateversions=01,04,07,10
podtype=test
```

Linux/UNIX

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
password=examplePassword.epw
url=exampleURL
updatemonths=02,05,08,11
```

Table 3-14 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis JAVA_HOME. Nur für Linux/UNIX.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (epmautomate.sh). Nur für Linux/UNIX.
username	Benutzername eines Serviceadministrators.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
url	URL der Umgebung, in der Sie das nicht monatliche Update-Intervall festlegen möchten.
updateversions	updateversions - Eine durch Komma getrennte Liste der EPM Cloud-Updates, die auf die durch den Parameter url identifizierte Umgebung angewendet werden sollen. Beispiel: updateversions=02,05,08,11. Versionen müssen mit zwei Ziffern angegeben werden. Stellen Sie für die Updates 01 bis 09 eine Null voran. Das Skript versucht, den Befehl skipUpdate für die Updates auszuführen, die nicht im Parameterwert updateversions enthalten sind. Beispiel: Wenn Sie updateversions=02,05,08,11 festlegen, versucht das Skript, Kennzeichen zum Überspringen von Updates für 01 (Januar), 03 (März), 04 (April), 06 (Juni), 07 (Juli), 09 (September), 10 (Oktober) und 12 (Dezember) festzulegen. In diesem Fall werden die EPM Cloud-Updates 02 (Februar), 05 (Mai), 08 (August) und 11 (November) auf die Umgebung angewendet.

Table 3-14 (Cont.) Parameter für "input.properties"

Parameter	Beschreibung
podtype	EPM Cloud-Umgebungstyp. Gültige Werte sind <code>test</code> und <code>prod</code> .

Skripte ausführen**1. Nur für Windows und Linux/UNIX**

- Erstellen Sie die Datei `skip_update.ps1` oder `skip_update.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
- Erstellen Sie die Datei `input.properties`, und speichern Sie sie im selben Verzeichnis, in dem sich das Skript `skip_update` befindet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Informationen hierzu finden Sie unter [Datei "input.properties" zum Ausführen der skip_update-Skripte für Windows und Linux/UNIX erstellen](#).
Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.
- Starten Sie das Skript.
 - **Windows PowerShell:** Führen Sie die Datei `skip_update.ps1` aus.
 - **Linux/UNIX:** Führen Sie die Datei `./skip_update.sh` aus.

2. Serverseitiges Groovy-Skript:

- Erstellen Sie das Groovy-Skript `skip_update.groovy`, und aktualisieren Sie es nach Bedarf.
- Verwenden Sie das Groovy-Fenster in einem EPM Cloud-Geschäftsprozess, oder automatisieren Sie die Skriptausführung mit [runBusinessRule](#). Informationen zur Ausführung des Groovy-Skripts mit EPM Automate finden Sie unter [Befehle ohne Installation von EPM Automate ausführen](#).

Beispielszenarios für Planning, Consolidation, Tax Reporting und Enterprise Profitability and Cost Management

Mit den in diesem Abschnitt verfügbaren Skripten können Sie Aufgaben in Planning- (einschließlich Planning Modules), Financial Consolidation and Close-, Tax Reporting- und Enterprise Profitability and Cost Management-Umgebungen automatisieren.

Siehe auch:

- [Export einer großen Anzahl von Zellen aus einem Aggregate Storage-Cube automatisieren](#)
Verwenden Sie das PowerShell- oder Bash-Skript in diesem Abschnitt, um eine große Anzahl an Zellen aus einem Aggregate Storage-Cube (ASO) zu exportieren.
- [Metadaten in Anwendungen importieren](#)
Verwenden Sie diese Skripte, um Anwendungsmetadaten manuell aus einer Datei zu importieren.

- [Daten importieren, Berechnungsskripte ausführen und Daten aus einer Block Storage-Datenbank in eine Aggregate Storage-Datenbank kopieren](#)
Verwenden Sie diese Skripte, um Daten aus einer Datei zu importieren, den Cube zu aktualisieren, eine Geschäftsregel zum Berechnen des Cubes auszuführen und Daten anschließend in einen ASO-Cube zu pushen.
- [Metadaten und Daten exportieren und herunterladen](#)
Verwenden Sie diese Skripte, um Anwendungsmetadaten und Daten zu exportieren und die Exportdateien anschließend in ein lokales Verzeichnis herunterzuladen.
- [Anwendungsdaten exportieren und herunterladen](#)
Verwenden Sie diese Skripte, um Anwendungsdaten zu exportieren und anschließend in ein lokales Verzeichnis herunterzuladen.
- [Das Archivieren von Auditdatensätzen der Anwendung automatisieren](#)
Verwenden Sie die Windows- und Linux-Skripte in diesem Abschnitt, um den Prozess des Exportierens und Archivierens von Auditdatensätzen der Anwendung auf einen/einem lokalen Computer zu automatisieren.
- [Eine Datendatei in eine Umgebung hochladen und eine Dataload-Regel ausführen](#)
Verwenden Sie diese Skripte, um eine Datei in eine Umgebung hochzuladen, und führen Sie dann eine Datenregel aus, um Daten aus der Datei in eine Anwendung zu importieren.
- [Integration von täglichen Daten automatisieren](#)
In diesem Szenario wird die Verwendung eines Beispielskripts zur regelmäßigen Automatisierung der Datenintegration untersucht.

Export einer großen Anzahl von Zellen aus einem Aggregate Storage-Cube automatisieren

Verwenden Sie das PowerShell- oder Bash-Skript in diesem Abschnitt, um eine große Anzahl an Zellen aus einem Aggregate Storage-Cube (ASO) zu exportieren.

Aufgrund der Grenzwerte, die durch die Oracle Essbase-Konfigurationseinstellung `QUERYRESULTLIMIT` vorgegeben sind, ist es nicht möglich, eine große Datenmenge über die Benutzeroberfläche zu exportieren. Das in diesem Abschnitt verfügbare PowerShell-Skript teilt den Export in eine festgelegte Anzahl von Jobs auf, führt die einzelnen Jobs aus, lädt die exportierten Daten herunter und verkettet die Exportdateien zu einer Exportdatei. So wird sichergestellt, dass nur ein Header vorhanden ist.

Hinweis:

Diese Skripte führen einen vorhandenen Job vom Typ "Daten exportieren" aus. Ausführliche Anweisungen zum Erstellen von Jobs finden Sie unter *Jobs verwalten* in der Dokumentation *Planning verwalten*.

PowerShell-Skript

```
$user = '<USERNAME>'
$pass = '<PASSWORD>'
$serverURL = '<URL>'
```

```

$applicationName = '<APPLICATIONNAME>'
$cubeName = '<CUBENAME>'
$splitDimension = '<DIMENSION_TO_SPLIT_THE_EXPORT>'
$topLevelMemberForExport = '<TOP_MEMBER_FOR_EXPORT>'
$exportJobName = '<EXPORT_JOB_NAME>'
$exportFilePrefix = '<PREFIX_FOR_EXPORT_FILE>'
$columnMembers = '<MEMBERS_ON_COLUMNS>'
$povMembers = '<POV_MEMBERS>'
$numberOfExportFiles = <NUMBER_OF_FILES_TO_SPLIT_THE_EXPORT>

$memberArray = @()
$exportFileArray = @()

function getLevel0 ($parent) {
    $parent.children.ForEach({
        if ( $_.children.count -eq 0 ) {
            $script:memberArray += $_.name
        }
        getLevel0($_)
    })
}

function findMember ($tree, $memberName) {
    $subtree = ""
    if ($tree.name -eq $memberName){
        return $tree
    } else {
        $tree.children.ForEach({
            #Write-Host $_.name
            if ($subtree -eq ""){ $subtree = findMember $_ $memberName}
        })
        return $subtree
    }
}

#putting together base64 encoded authentication header based un user and
password
$encodedCredentials =
[Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(($($user) +
":" + $($pass))))
$headers = @{ Authorization = "Basic $encodedCredentials" }

#test login
$testRequest = $serverURL + '/HyperionPlanning/rest/v3/applications'

try {
    $response = Invoke-RestMethod -Uri $testRequest -Method Get -
Headers $headers -UseBasicParsing
}
catch {
    Write-Host $_
    return
}

#retrieve dimension hierarchy from application

```

```

Write-Host "Retrieving member list for split dimension
" $splitDimension
$request = $serverURL + '/HyperionPlanning/rest/v3/internal/
applications/' + $applicationName + '/plantypes/' + $cubeName + '/'
dimensions/' + $splitDimension
try {
    $response = Invoke-RestMethod -Uri $request -Method Get -
Headers $headers -UseBasicParsing
}
catch {
    Write-Host $_
    return
}
Write-Host $splitDimension " member list retrieved"

#search for the top of the export hierarchy
Write-Host "Searching for member " $topLevelMemberForExport " in
hierarchy"
$member = findMember $response $topLevelMemberForExport
if ( $member.name -ne $topLevelMemberForExport ) {
    Write-Host $topLevelMemberForExport " not found in hierarchy,
exiting ..."
    return 128
}
Write-Host "Found member " $topLevelMemberForExport " in hierarchy"

#retrieve level 0 members in export hierarchy
Write-Host "Retrieving Level 0 members for hierarchy"
getLevel0($member)
if ( $memberArray.Length -eq 0 ) {
    Write-Host "no level 0 members found in hierarchy, exiting ..."
    return 128
}
Write-Host $memberArray.Length " Level 0 members for export hierarchy
retrieved"

$request = $serverURL + '/HyperionPlanning/rest/v3/applications/'
+ $applicationName + '/jobs'

#splitting member list into the number of export files
$numberOfEntitiesPerFile =
[math]::truncate($memberArray.Length / $numberOfExportFiles)
for ($i = 1; $i -le $numberOfExportFiles; $i++) {
    $memberList = ""
    $firstMember = ($i - 1) * $numberOfEntitiesPerFile
    if ($i -lt $numberOfExportFiles) {
        $lastMember = $i * $numberOfEntitiesPerFile
    } else {
        $lastMember = $i * $numberOfEntitiesPerFile
    }
    + $memberArray.Length % $numberOfExportFiles
    for ($j = $firstMember; $j -lt $lastMember; $j++) {
        $memberList += $memberArray[$j]
        if ($j -lt $lastMember - 1) {$memberList += ","} #avoid adding

```

```

a comma (,) after the last member of each set
}

$jobDetails='
{
"jobType":"EXPORT_DATA","jobName":"' + $exportJobName + '",
"parameters":{
    "exportFileName":"Export-' + $i + '.zip",
    "rowMembers":"' + $memberList + '",
    "columnMembers":"' + $columnMembers + '",
    "povMembers":"' + $povMembers + '"
}
}'

#start export job
try{
    $response = Invoke-RestMethod -Uri $request -Method Post -
Headers $headers -Body $jobDetails -ContentType "application/json"
    catch {
        Write-Host $_
        return
    }

    Write-Host "Started export job " $i " out of " $numberOfExportFiles

    #checking job status, continue once jos is completed
    $statusRequest = $serverURL + '/HyperionPlanning/rest/v3/applications/'
+ $applicationName + '/jobs/' + $response.jobId
    $statusResponse = Invoke-RestMethod -Uri $statusRequest -Method Get -
Headers $headers -UseBasicParsing

    while ( $statusResponse.descriptiveStatus -eq "Processing" ) {
        Write-Host $statusResponse.descriptiveStatus
        Start-Sleep -s 10
        $statusResponse = Invoke-RestMethod -Uri $statusRequest -Method Get -
Headers $headers -UseBasicParsing
    }
    Write-Host $statusResponse.descriptiveStatus

    Write-Host "Downloading export file ..."
    $downloadRequest = $serverURL + '/interop/rest/11.1.2.3.600/
applicationsnapshots/Export-' + $i + '.zip/contents'
    $statusResponse = Invoke-RestMethod -Uri $downloadRequest -Method Get -
Headers $headers -OutFile "$exportFilePrefix-$i.zip"

    Write-Host "Expanding archive ..."
    Expand-Archive -Force -LiteralPath "$exportFilePrefix-$i.zip" -
DestinationPath "$exportFilePrefix-$i"
    Remove-Item "$exportFilePrefix-$i.zip"

    Get-ChildItem -Path "$exportFilePrefix-$i" -File -Name | ForEach-Object
{ $exportFileArray += "$exportFilePrefix-$i\" + $_ }
}

Write-Host "creating outputfile ..."

```



```
#write header to outputfile
Get-Content $exportFileArray[0] | Select-Object -First 1 | Out-File
"$exportFilePrefix.csv"

#write content to outputfile skipping header
ForEach ($exportFile in $exportFileArray) {
    Get-Content $exportFile | Select-Object -Skip 1 | Out-File -Append
"$exportFilePrefix.csv"
}

Compress-Archive -LiteralPath "$exportFilePrefix.csv" -
DestinationPath "$exportFilePrefix.zip"

Write-Host "cleaning up ..."
Remove-Item "$exportFilePrefix-*" -Recurse
Remove-Item "$exportFilePrefix.csv"
```

Bash-Skript

```
#!/bin/bash

user='<USERNAME>'
pass='<PASSWORD>'
serverURL='<URL>'
applicationName='<APPLICATIONNAME>'
cubeName='<CUBENAME>'
splitDimension='<DIMENSION_TO_SPLIT_THE_EXPORT>'
topLevelMemberForExport='<TOP_MEMBER_FOR_EXPORT>'
exportJobName='<EXPORT_JOB_NAME>'
exportFilePrefix='<PREFIX_FOR_EXPORT_FILE>'
columnMembers='<MEMBERS_ON_COLUMNS>'
povMembers='<POV_MEMBERS>'
numberOfExportFiles=<NUMBER_OF_FILES_TO_SPLIT_THE_EXPORT>

getRowMembers() {
    local memberList="$1"
    local firstMember=$2
    local lastMember=$3
    local nameCount=0
    local rowMember=""
    local rowMembers=""

    while IFS= read -r line
    do
        if [[ "${line}" == *"name"* ]]
        then
            if [[ ${nameCount} -ge ${firstMember} ]] && [[ ${
{nameCount} -lt ${lastMember} ]]
            then
                rowMember=$(echo "${line}" | cut -d':' -f2- | sed s/'
[","]//g')
                rowMembers="${rowMembers}${rowMember},"
            fi
            ((nameCount+=1))
        fi
    done
}
```

```

        fi
        done <<< "${memberList}"
        rowMembers=$(echo "${rowMembers}" | rev | cut -d',' -f2- | rev)
        echo "${rowMembers}"
    }

getLevel0()
{
    local memberList="$1"
    local names=$(echo "${memberList}" | jq 'recurse (try .children[])
| .name' | sed -e 's/"//g')
    local elements=""

    formerIFS=$IFS
    IFS=$'\n'
    namesarr=(($names))
    IFS=$formerIFS

    for i in ${!namesarr[@]}
    do
        testelement=$(echo "${memberList}" | jq --arg currentName "$
${namesarr[i]}" 'recurse (try .children[]) | select(.name==$currentName)')
        if [[ "${testelement}" != *"children"* ]]
        then
            elements="${elements}${testelement}"
        fi
    done

    echo "${elements}"
}

#test login
header="Content-Type: application/x-www-form-urlencoded"
applicationsRequest="${serverURL}/HyperionPlanning/rest/v3/applications"
response=$(curl -X "GET" -s -w "%{http_code}" -u "${user}:${pass}" -H "${
header}" "${applicationsRequest}")
http_response_code=$(echo "${response}" | rev | cut -d'}' -f1 | rev)

if [ ${http_response_code} -ne 200 ]
then
    echo "${response}"
    exit
fi

#retrieve dimension hierarchy from application
echo "Retrieving member list for split dimension ${splitDimension}"
splitDimensionRequest="${serverURL}/HyperionPlanning/rest/v3/internal/
applications/${applicationName}/plantypes/${cubeName}/dimensions/${
splitDimension}"
response=$(curl -X GET -s -w "%{http_code}" -u "${user}:${pass}" -o
"response-memberlist.txt" -D "respHeader-memberlist.txt" -H "${header}" "${
splitDimensionRequest}")
http_response_code=$(echo "${response}" | rev | cut -d'}' -f1 | rev)

if [ ${http_response_code} -ne 200 ]

```

```

then
    echo "${response}"
    exit
fi

echo "${splitDimension} member list retrieved"

#search for the top of the export hierarchy
echo "Searching for member ${topLevelMemberForExport} in hierarchy"
memberList=$(cat response-memberlist.txt | jq --arg topLevelMember "${topLevelMemberForExport}" 'recurse(try .children[]) | select (.name == $topLevelMember)')
if [[ "${memberList}" == "" ]]
then
    echo "${topLevelMemberForExport} not found in hierarchy, exiting ..."
    exit 128
fi

echo "Found member ${topLevelMemberForExport} in hierarchy"

#retrieve level 0 members in export hierarchy
echo "Retrieving Level 0 members for hierarchy"
totalCount=$(echo "${memberList}" | grep "name" | wc -l)
grepChildrenCount=$(echo "${memberList}" | grep "children" | wc -l)
levelZeroCount=$((totalCount-grepChildrenCount))

if [[ "${levelZeroCount}" -eq 0 ]]
then
    echo "no level 0 members found in hierarchy, exiting ..."
    exit 128
fi

echo "${levelZeroCount} Level 0 members for export hierarchy retrieved"

#splitting member list into the number of export files
numberOfEntitiesPerFile=$((levelZeroCount/numberOfExportFiles))
jobsRequest="${serverURL}/HyperionPlanning/rest/v3/applications/{applicationName}/jobs"
header="Content-Type: application/json"

for ((i = 1 ; i <= ${numberOfExportFiles}; i++))
do
    firstMember=$((($i-1)*numberOfEntitiesPerFile)
    if [[ $i -lt $numberOfExportFiles ]]
    then
        lastMember=$((i*numberOfEntitiesPerFile))
    else
        lastMember=$((i*numberOfEntitiesPerFile+levelZeroCount%numberOfExportFiles))
    fi

    elements=$(getLevel0 "${memberList}")
    rowMembers=$(getRowMembers "${elements}" ${firstMember} ${lastMember})

```

```

    response=$(curl -X POST -s -w "%{http_code}" -u "${user}:${pass}" -o
"response-job.txt" -D "respHeader-job.txt" -H "${header}" "${jobsRequest}" -
d '{"jobType":"EXPORT_DATA","jobName":"'${exportJobName}'',"parameters":
{"exportFileName":"Export-'${i}'.zip","rowMembers":"","$
{rowMembers}","", "columnMembers":"'${columnMembers}'","", "povMembers":"'${
povMembers}'"'})

    echo "Started export job " $i " out of " $numberOfExportFiles
    jobId=$(cat response-job.txt | grep -o '"jobId":[^\, }]*' | cut -d':' -f2)
    descriptiveStatus=$(cat response-job.txt | grep -o '"descriptiveStatus":
[^, }]*' | cut -d':' -f2 | sed -e 's/"//g')
    jobIdRequest="${serverURL}/HyperionPlanning/rest/v3/applications/$
{applicationName}/jobs/${jobId}"
    response=$(curl -X GET -s -w "%{http_code}" -u "${user}:${pass}" -o
"response-jobstatus.txt" -D "respHeader-jobstatus.txt" -H "${header}" "${
jobIdRequest}")

    jobId=$(cat response-jobstatus.txt | grep -o '"jobId":[^\, }]*' | cut -
d':' -f2)
    descriptiveStatus=$(cat response-jobstatus.txt | grep -o
'"descriptiveStatus":[^\, }]*' | cut -d':' -f2 | sed -e 's/"//g')

    while [[ "${descriptiveStatus}" == "Processing" ]]
    do
        echo "${descriptiveStatus}"
        sleep 10
        response=$(curl -X GET -s -w "%{http_code}" -u "${user}:${pass}" -o
"response-jobstatus.txt" -D "respHeader-jobstatus.txt" -H "${header}" "${
jobIdRequest}")
        descriptiveStatus=$(cat response-jobstatus.txt | grep -o
'"descriptiveStatus":[^\, }]*' | cut -d':' -f2 | sed -e 's/"//g')
    done

    echo "${descriptiveStatus}"

    echo "Downloading export file ..."
    contentsRequest="${serverURL}/interop/rest/11.1.2.3.600/
applicationsnapshots/Export-${i}.zip/contents"
    curl -X GET -s -w "%{http_code}" -u "${user}:${pass}" -D "respHeader-
download.txt" "${contentsRequest}" > "${exportFilePrefix}-${i}.zip"

    echo "Expanding archive ..."
    unzip "${exportFilePrefix}-${i}.zip" -d "${exportFilePrefix}-${i}"
    rm "${exportFilePrefix}-${i}.zip"

    echo "Writing to outputfile ..."
    if [[ -d "${exportFilePrefix}-${i}" ]]
    then
        find "${exportFilePrefix}-${i}" -name \*.csv | xargs cat | tail -n
+2 >> "${exportFilePrefix}.csv"
    fi
done
zip "${exportFilePrefix}.zip" "${exportFilePrefix}.csv"

```

```
echo "cleaning up ..."  
find . -name "${exportFilePrefix}-*" | xargs rm -r  
rm "${exportFilePrefix}.csv"
```

So exportieren Sie eine große Anzahl von Zellen aus einem Aggregate Storage-Cube (ASO):

1. Kopieren Sie das PowerShell- oder Bash-Skript, und speichern Sie es in Ihrem Dateisystem, z.B. unter `ASOCellExport.ps1` oder `ASOCellExport.sh`.
2. Ändern Sie die Skriptdatei, und legen Sie die Parameterwerte fest. Details finden Sie in der nachstehenden Tabelle.

Tabelle 3-15 In die PowerShell- und Bash-Skripte einzuschließende Variablenwerte

Variable	Beschreibung
user	Domain und Benutzername für einen Serviceadministrator im Format <i>DOMAIN.USER</i> . Beispiele: Windows: <code>\$user = 'exampleDomain.jDoe'</code> Linux/UNIX: <code>user = 'exampleDomain.jDoe'</code>
pass	Kennwort des Serviceadministrators oder Speicherort der verschlüsselten Kennwortdatei. Informationen zum Erstellen einer verschlüsselten Kennwortdatei finden Sie im Abschnitt zum Befehl encrypt . Beispiele: Windows: <code>\$pass = 'Example'</code> Linux/UNIX: <code>pass = 'Example'</code>
serverURL	Die URL der Oracle Enterprise Performance Management Cloud-Umgebung. Beispiele: Windows: <code>\$serverURL = 'https://example.oraclecloud.com'</code> Linux/UNIX: <code>serverURL = 'https://example.oraclecloud.com'</code>
applicationName	Name einer Planning-, Financial Consolidation and Close-, Tax Reporting- oder Enterprise Profitability and Cost Management-Anwendung. Beispiele: Windows: <code>\$applicationName = 'Vision'</code> Linux/UNIX: <code>applicationName = 'Vision'</code>
cubeName	Name eines Cubes in der Anwendung. Beispiele: Windows: <code>\$cubeName = 'VisASO'</code> Linux/UNIX: <code>cubeName = 'VisASO'</code>

Tabelle 3-15 (Fortsetzung) In die PowerShell- und Bash-Skripte einzuschließende Variablenwerte

Variable	Beschreibung
splitDimension	<p>Name einer Dimension, deren Elemente dazu verwendet werden, den Export in Gruppen aufzuteilen.</p> <p>Beispiele:</p> <p>Windows: \$splitDimension = 'Account'</p> <p>Linux/UNIX: splitDimension = 'Account'</p>
topLevelMemberForExport	<p>Name eines Elements der Dimensionsunterhierarchie, unter der eine Liste von Elementen der Ebene 0 erstellt wird.</p> <p>Beispiele:</p> <p>Windows: \$topLevelMemberForExport = 'Total Cash Flow'</p> <p>Linux/UNIX: topLevelMemberForExport = 'Total Cash Flow'</p>
exportJobName	<p>Name eines vorhandenen Jobs vom Typ "Daten exportieren". Die in diesem Job angegebenen Einstellungen werden durch die Parameter überschrieben, die Sie im Skript festlegen.</p> <p>Beispiele:</p> <p>Windows: \$exportJobName = 'ASO Cell Export'</p> <p>Linux/UNIX: exportJobName = 'ASO Cell Export'</p>
exportFilePrefix	<p>Ein Dateinamenspräfix zur eindeutigen Identifizierung der Dateien, die vom Exportjob generiert werden.</p> <p>Beispiele:</p> <p>Windows: \$exportFilePrefix = 'cashflow'</p> <p>Linux/UNIX: exportFilePrefix = 'cashflow'</p>
columnMembers	<p>Die Elementspalten, die in den Export eingeschlossen werden sollen.</p> <p>Beispiele:</p> <p>Windows: \$columnMembers = 'Period'</p> <p>Linux/UNIX: columnMembers = 'Period'</p>
povMembers	<p>Point of Views, die in den Export eingeschlossen werden sollen. POV-Elemente müssen alle anderen Dimensionen enthalten und können Funktionen aufweisen, wie im Folgenden gezeigt:</p> <pre> ILvl0Descendants (YearTotal), ILvl0Descendants (Year), ILvl0Descendants (Scenario), ILvl0Descendants (Version), ILvl0Descendants (P_TP), ILvl0Descendants (AltYear) </pre> <p>Beispiele:</p> <p>Windows: \$povMembers = 'YTD'</p> <p>Linux/UNIX: povMembers = 'YTD'</p>

Tabelle 3-15 (Fortsetzung) In die PowerShell- und Bash-Skripte einzuschließende Variablenwerte

Variable	Beschreibung
numberOfExportFiles	Anzahl von Jobs, die für diesen Exportvorgang ausgeführt werden sollen. Erhöhen Sie diese Zahl, wenn die Einschränkungen aufgrund von Abfragegrenzwerten weiterhin dazu führen, dass der Export nicht erfolgreich ist. Beispiele: Windows: \$numberOfExportFiles = 3 Linux/UNIX: numberOfExportFiles = 3

- Planen Sie die Ausführung des Skripts zu einem geeigneten Zeitpunkt mit dem Windows Scheduler oder einem Cron-Job. Detaillierte Schritte finden Sie unter [Skriptausführung automatisieren](#)

Metadaten in Anwendungen importieren

Verwenden Sie diese Skripte, um Anwendungsmetadaten manuell aus einer Datei zu importieren.

Diese Skripte führen die folgenden Aktivitäten aus:

- Anmeldung bei einer Umgebung
- Hochladen einer Metadatendatei
- Import von Metadaten aus der hochgeladenen Datei in eine Anwendung mit einem Job
- Aktualisierung des Cubes
- Abmeldung

Windows-Beispielskript

Erstellen Sie die Datei `importMetadata.ps1`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
$inputproperties = ConvertFrom-StringData (Get-Content ./
input.properties -raw)
$username="$($inputproperties.username) "
$passwordfile="$($inputproperties.passwordfile) "
$serviceURL="$($inputproperties.serviceURL) "
$file1="$($inputproperties.file1) "
$jobName="$($inputproperties.jobName) "

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile ${file1}
epmautomate importmetadata ${jobName} ${file1}
epmautomate refreshcube
epmautomate logout
```

Linux-/UNIX-Beispielskript

Erstellen Sie die Datei `importMetadata.sh`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "${serviceURL}"
${epmautomatescript} uploadfile "${file1}"
${epmautomatescript} importmetadata "${jobName}" "${file1}"
${epmautomatescript} refreshcube
${epmautomatescript} logout
```

Datei "input.properties" erstellen

Erstellen Sie die Datei `input.properties`, indem Sie eines der folgenden Skripte kopieren, und aktualisieren Sie sie mit Informationen für Ihre Umgebung. Speichern Sie die Datei im selben Verzeichnis, in dem die Datei `importMetadata.ps1` oder `importMetadata.sh` gespeichert ist.

Windows

```
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
file1=FILE_NAME.zip
jobName=JOB_NAME
```

Linux/UNIX

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
file1=FILE_NAME.zip
jobName=JOB_NAME
```

Tabelle 3-16 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis <code>JAVA_HOME</code> . Nur für Linux/UNIX.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (<code>epmautomate.sh</code>). Nur für Linux/UNIX.
username	Benutzername eines Serviceadministrators, der auch über die Rolle Identitätsdomainadministrator verfügt.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.

Tabelle 3-16 (Fortsetzung) Parameter für "input.properties"

Parameter	Beschreibung
serviceURL	URL der Umgebung, aus der Sie den Snapshot generieren möchten.
File1	Name der ZIP-Datei, die die zu importierenden Metadaten enthält.
JobName	Job, der zum Importieren der Metadaten verwendet werden soll.

Skripte ausführen

1. Erstellen Sie die Datei `importMetadata.ps1` oder `importMetadata.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
2. Erstellen Sie die Datei `input.properties`, und speichern Sie sie im selben Verzeichnis, in dem sich das Skript `importMetadata` befindet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Informationen hierzu finden Sie unter [Datei "input.properties" erstellen](#).
Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.
3. Starten Sie das Skript.
 - **Windows PowerShell:** Führen Sie die Datei `importMetadata.ps1` aus.
 - **Linux/UNIX:** Führen Sie die Datei `./importMetadata.sh` aus.

Daten importieren, Berechnungsskripte ausführen und Daten aus einer Block Storage-Datenbank in eine Aggregate Storage-Datenbank kopieren

Verwenden Sie diese Skripte, um Daten aus einer Datei zu importieren, den Cube zu aktualisieren, eine Geschäftsregel zum Berechnen des Cubes auszuführen und Daten anschließend in einen ASO-Cube zu pushen.

Diese Skripte führen die folgenden Aktionen aus:

- Anmeldung bei einer Umgebung
- Hochladen einer `data.csv`-Datei
- Import von Daten aus der `data.csv`-Datei in die Anwendung mit dem Job `loadingq1data`
- Aktualisierung des Cubes
- Ausführung von Geschäftsregeln zur Umwandlung von Daten
- Daten-Push in eine Aggregate Storage-Datenbank mit einem Job
- Abmeldung

Windows-Beispielskript

Erstellen Sie die Datei `importDataPlus.ps1`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -
raw)
$username="$($inputproperties.username) "
$passwordfile="$($inputproperties.passwordfile) "
$serviceURL="$($inputproperties.serviceURL) "
$importDataJobName="$($inputproperties.importDataJobName) "
$businessRuleName="$($inputproperties.businessRuleName) "
$planTypeMapName="$($inputproperties.planTypeMapName) "
$param1Key="$($inputproperties.param1Key) "
$param1Value="$($inputproperties.param1Value) "
$param2Key="$($inputproperties.param2Key) "
$param2Value="$($inputproperties.param2Value) "
$clearData="$($inputproperties.clearData) "

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile ${file1}
epmautomate importdata ${importDataJobName} ${file1}
epmautomate refreshcube
epmautomate runbusinessrule ${businessRuleName} ${param1Key}=${param1Value} ${
param2Key}=${param2Value}
epmautomate runplantypemap ${planTypeMapName} clearData=${clearData}
epmautomate logout
```

Linux-/UNIX-Beispielskript

Erstellen Sie die Datei `importDataPlus.ps1`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "${serviceURL}"
${epmautomatescript} uploadfile "${file1}"
${epmautomatescript} importdata "${importDataJobName}" "${file1}"
${epmautomatescript} refreshcube
${epmautomatescript} runbusinessrule "${businessRuleName}" "${param1Key}=${
param1Value}" "${param2Key}=${param2Value}"
${epmautomatescript} runplantypemap "${planTypeMapName}" clearData=${
clearData}
${epmautomatescript} logout
```

Datei "input.properties" erstellen

Windows

```
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
```

```
File1=FILE_NAME.csv
importDataJobName=FILE_NAME
businessRuleName=RULE_NAME
planTypeMapName=PLAN_TYPE_MAP_NAME
param1Key=RUN-TIME PARAMETER_1
param1Value=RUN-TIME PARAMETER_1_VALUE
param2Key=RUN-TIME PARAMETER_2
param2Value=RUN-TIME PARAMETER_2_VALUE
clearData=true
```

Linux/UNIX

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
File1=FILE_NAME.csv
importDataJobName=FILE_NAME
businessRuleName=RULE_NAME
planTypeMapName=PLAN_TYPE_MAP_NAME
param1Key=RUN-TIME PARAMETER_1
param1Value=RUN-TIME PARAMETER_1_VALUE
param2Key=RUN-TIME PARAMETER_2
param2Value=RUN-TIME PARAMETER_2_VALUE
clearData=true
```

Tabelle 3-17 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis JAVA_HOME. Nur für Linux/UNIX.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (epmautomate.sh). Nur für Linux/UNIX.
username	Benutzername eines Serviceadministrators, der auch über die Rolle Identitätsdomainadministrator verfügt.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
serviceURL	URL der Umgebung, aus der Sie den Snapshot generieren möchten.
File1	Importdatei, aus der die Daten in die Anwendung geladen werden sollen.
iimportDataJobName	Name des Jobs, der für den Datenimport verwendet werden soll.
businessRuleName	Die Geschäftsregel, die für die importierten Daten ausgeführt werden soll.
planTypeMapName	Die Plantypzuordnung, die zum Kopieren der Daten aus einer BSO-Datenbank in eine ASO-Datenbank oder aus einer BSO-Datenbank in eine andere BSO-Datenbank verwendet werden soll.
param1Key	Runtime Prompt 1 zum Ausführen der Geschäftsregel.
param1Value	Wert von Runtime Prompt 1.

Tabelle 3-17 (Fortsetzung) Parameter für "input.properties"

Parameter	Beschreibung
param2Key	Runtime Prompt 2 zum Ausführen der Geschäftsregel.
param2Value	Wert von Runtime Prompt 2.
clearData	Angabe, ob die Daten in der empfangenden Datenbank gelöscht werden sollen. Geben Sie <code>false</code> an, um die Daten beizubehalten.

Skripte ausführen

1. Erstellen Sie die Datei `importDataPlus.ps1` oder `importDataPlus.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
2. Erstellen Sie die Datei `input.properties`, und speichern Sie sie im selben Verzeichnis, in dem sich das Skript `importDataPlus` befindet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Informationen hierzu finden Sie unter [Datei "input.properties" erstellen](#).
Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.
3. Starten Sie das Skript.
 - **Windows PowerShell:** Führen Sie die Datei `importDataPlus.ps1` aus.
 - **Linux/UNIX:** Führen Sie die Datei `./importDataPlus.sh` aus.

Metadaten und Daten exportieren und herunterladen

Verwenden Sie diese Skripte, um Anwendungsmetadaten und Daten zu exportieren und die Exportdateien anschließend in ein lokales Verzeichnis herunterzuladen.

Diese Skripte führen die folgenden Aktivitäten aus:

- Anmeldung bei einer Umgebung
- Export der Metadaten in eine ZIP-Datei mit einem angegebenen Job.
- Export der Anwendungsdaten in eine ZIP-Datei mit einem angegebenen Job.
- Auflisten des Inhalts der Inbox/Outbox.
- Download der exportierten Datendateien auf den lokalen Computer.
- Melden Sie sich ab.

Windows-Beispielskript

Erstellen Sie die Datei `exportDownloadMetadataAndData.ps1`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -
raw)
$username="$($inputproperties.username) "
$passwordfile="$($inputproperties.passwordfile) "
$serviceURL="$($inputproperties.serviceURL) "
```

```

$exportFile1="$($inputproperties.exportFile1) "
$exportFile2="$($inputproperties.exportFile2) "
$exportMetaDataJobName="$($inputproperties.exportMetaDataJobName) "
$exportDataJobName="$($inputproperties.exportDataJobName) "

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate exportmetadata ${exportMetaDataJobName} ${exportFile1}
epmautomate exportdata ${exportDataJobName} ${exportFile2}
epmautomate listfiles
epmautomate downloadfile ${exportFile1}
epmautomate downloadfile f${exportFile2}
epmautomate logout

```

Linux-/UNIX-Beispielskript

Erstellen Sie die Datei `exportDownloadMetadataAndData.sh`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```

#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "${serviceURL}"
${epmautomatescript} exportmetadata "${exportMetaDataJobName}" "${exportFile1}"
${epmautomatescript} exportdata "${exportDataJobName}" "${exportFile2}"
${epmautomatescript} listfiles
${epmautomatescript} downloadfile "${exportFile1}"
${epmautomatescript} downloadfile "${exportFile2}"
${epmautomatescript} logout

```

Eigenschaftendatei erstellen

Erstellen Sie die Datei `input.properties`, indem Sie eines der folgenden Skripte kopieren, und aktualisieren Sie sie mit Informationen für Ihre Umgebung. Speichern Sie die Datei im selben Verzeichnis, in dem die Datei `exportDownloadMetadataAndData.ps1` oder `exportDownloadMetadataAndData.sh` gespeichert ist.

Windows

```

username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
exportFile1=FILE_NAME1.zip
exportFile2=FILE_NAME2.zip
exportMetaDataJobName=METADATA_EXPORT_JOB_NAME
exportDataJobName=DATA_EXPORT_JOB_NAME

```

Linux/UNIX

```

javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin

```

```
passwordfile=examplePassword.epw
serviceURL=exampleURL
exportFile1=FILE_NAME1.zip
exportFile2=FILE_NAME2.zip
exportMetaDataJobName=METADATA_EXPORT_JOB_NAME
exportDataJobName=DATA_EXPORT_JOB_NAME
```

Tabelle 3-18 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis JAVA_HOME. Nur für Linux/UNIX.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (epmautomate.sh). Nur für Linux/UNIX.
username	Benutzername eines Serviceadministrators, der auch über die Rolle Identitätsdomainadministrator verfügt.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
serviceURL	URL der Umgebung, aus der Sie den Snapshot generieren möchten.
exportFile1	Name der Datei, in die die Metadaten exportiert werden sollen.
exportFile2	Name der Datei, in die die Daten exportiert werden sollen.
exportDataJobName1	Job, der zum Exportieren der Metadaten verwendet werden soll.
exportDataJobName2	Job, der zum Exportieren der Daten verwendet werden soll.

Skripte ausführen

1. Erstellen Sie die Datei `exportDownloadMetadataAndData.ps1` oder `exportDownloadMetadataAndData.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
2. Erstellen Sie die Datei `input.properties`, und speichern Sie sie im selben Verzeichnis, in dem sich das Skript `exportDownloadMetadataAndData` befindet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Informationen hierzu finden Sie unter [Eigenschaftendatei erstellen](#).
Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.
3. Starten Sie das Skript.
 - **Windows PowerShell:** Führen Sie die Datei `exportDownloadMetadataAndData.ps1` aus.
 - **Linux/UNIX:** Führen Sie die Datei `./exportDownloadMetadataAndData.sh` aus.

Anwendungsdaten exportieren und herunterladen

Verwenden Sie diese Skripte, um Anwendungsdaten zu exportieren und anschließend in ein lokales Verzeichnis herunterzuladen.

Diese Skripte führen die folgenden Vorgänge aus:

- Anmeldung bei einer Umgebung
- Sicherung zweier Datasets mit den von Ihnen angegebenen Jobs
- Download der exportierten Datendateien
- Abmeldung

Windows-Beispielskript

Erstellen Sie die Datei `exportDownloadData.ps1`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./
input.properties -raw)
$username="$($inputproperties.username) "
$passwordfile="$($inputproperties.passwordfile) "
$serviceURL="$($inputproperties.serviceURL) "
$exportFile1="$($inputproperties.exportFile1) "
$exportFile2="$($inputproperties.exportFile2) "
$exportDataJobName1="$($inputproperties.exportDataJobName1) "
$exportDataJobName2="$($inputproperties.exportDataJobName2) "

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate exportdata ${exportDataJobName1} ${exportFile1}
epmautomate exportdata ${exportDataJobName2} ${exportFile2}
epmautomate listfiles
epmautomate downloadfile ${exportFile1}
epmautomate downloadfile ${exportFile2}
epmautomate logout
```

Linux-/UNIX-Beispielskript

Erstellen Sie die Datei `exportDownloadData.sh`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "$
{serviceURL}"
${epmautomatescript} exportdata "${exportDataJobName1}" "$
{exportFile1}"
${epmautomatescript} exportdata "${exportDataJobName2}" "$
{exportFile2}"
${epmautomatescript} listfiles
${epmautomatescript} downloadfile "${exportFile1}"
${epmautomatescript} downloadfile "${exportFile2}"
${epmautomatescript} logout
```

Datei "input.properties" erstellen

Erstellen Sie die Datei `input.properties`, indem Sie eines der folgenden Skripte kopieren, und aktualisieren Sie sie mit Informationen für Ihre Umgebung. Speichern Sie die Datei im selben Verzeichnis, in dem die Datei `exportDownloadData.ps1` oder `exportDownloadData.sh` gespeichert ist.

Windows

```
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
exportFile1=FILE_NAME.zip
exportFile2=FILE_NAME.zip
exportDataJobName1=JOB_NAME
exportDataJobName2=FILE_NAME
```

Linux/UNIX

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
exportFile1=FILE_NAME.zip
exportFile2=FILE_NAME.zip
exportDataJobName1=FILE_NAME
exportDataJobName2=FILE_NAME
```

Tabelle 3-19 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis JAVA_HOME. Nur für Linux/UNIX.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (epmautomate.sh). Nur für Linux/UNIX.
username	Benutzername eines Serviceadministrators, der auch über die Rolle Identitätsdomainadministrator verfügt.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
serviceURL	URL der Umgebung, aus der Sie den Snapshot generieren möchten.
exportFile1 und exportFile2	Name der Datei, in die die Daten exportiert werden sollen.
exportDataJobName1 und exportDataJobName2	Job, der zum Exportieren der Daten verwendet werden soll.

Skripte ausführen

1. Erstellen Sie die Datei `exportDownloadData.ps1` oder `exportDownloadData.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
2. Erstellen Sie die Datei `input.properties`, und speichern Sie sie im selben Verzeichnis, in dem sich das Skript `exportDownloadData` befindet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Informationen hierzu finden Sie in [Tabelle 1](#). Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.
3. Starten Sie das Skript.

- **Windows PowerShell:** Führen Sie die Datei `exportDownloadData.ps1` aus.
- **Linux/UNIX:** Führen Sie die Datei `./exportDownloadData.sh` aus.

Das Archivieren von Auditdatensätzen der Anwendung automatisieren

Verwenden Sie die Windows- und Linux-Skripte in diesem Abschnitt, um den Prozess des Exportierens und Archivierens von Auditdatensätzen der Anwendung auf einen/ einem lokalen Computer zu automatisieren.

Anwendungsauditdaten werden nur für 365 Tage beibehalten. Passen Sie diese Skripte an, und führen Sie sie einmal alle 180 Tage oder wie für Ihre Datenaufbewahrungs-Policys erforderlich aus, um den Verlust von historischen Auditdaten, die älter als 365 Tage sind, zu verhindern.



Note:

Diese Skripte sind darauf zugeschnitten, Daten im lokalen Speicher zu archivieren. Sie können sie ändern, sodass die exportierten Auditdatendateien im Netzwerkspeicher oder in der Speicher-Cloud (z.B. Oracle Object Storage) archiviert werden.

Table 3-20 Parameter und deren Werte

Parameter	Wert
url	Die URL der Umgebung. Beispiele: <ul style="list-style-type: none"> • Windows: <code>set url=https://example-epmidm.epm.usphoenix-1.ocs.oraclecloud.com/epmcloud</code> • Linux: <code>url=https://example-epmidm.epm.usphoenix-1.ocs.oraclecloud.com/epmcloud</code>
user	Der Benutzername eines Serviceadministrators für die Anmeldung bei der Umgebung, um Auditdaten herunterzuladen. Beispiele: <ul style="list-style-type: none"> • Windows: <code>set user=ExampleAdmin</code> • Linux: <code>user=ExampleAdmin</code>
password	Kennwort des Serviceadministrators (nicht empfohlen) oder Name und Speicherort der verschlüsselten Kennwortdatei. Informationen zum Erstellen einer verschlüsselten Kennwortdatei finden Sie im Abschnitt zum Befehl encrypt . Beispiele: <ul style="list-style-type: none"> • Windows: <code>set password="C:\mySecuredir\password.epw"</code> • Linux: <code>password="/home/user1/mySecuredir/password.epw"</code>

Table 3-20 (Cont.) Parameter und deren Werte

Parameter	Wert
AuditFileName	Name der Auditdatendatei. Um diese Datei eindeutig zu benennen, hängt das Skript den Zeitstempel des Auditdatenexports an den Dateinamen an. Beispiel: <ul style="list-style-type: none"> • Windows: set AuditFileName=AuditData • Linux: AuditFileName=AuditData
NumberOfBackups	Anzahl an Backupdateien, die im Speicher beibehalten werden sollen. Der Standardwert lautet 10. Danach wird das jeweils älteste Backup nach Bedarf ersetzt. Beispiel: <ul style="list-style-type: none"> • Windows: set NumberOfBackups=20 • Linux: NumberOfBackups=20
Nur für Linux-Skripte	
epmautomatescript	Der Speicherort, an dem EPM Automate installiert ist. Beispiel: /home/user1/epmautomate/bin/epmautomate.sh
javahome	Der Speicherort <code>JAVA_HOME</code> . Beispiel: /home/user1/jdk1.8.0_191/

Windows-Skript

Erstellen Sie eine Batch-Datei (z.B. `AuditExport.bat`), die ein ähnliches Skript wie das folgende enthält, um das Exportieren und Herunterladen von Auditdaten auf einen lokalen Computer zu automatisieren.

```
@echo off
rem Sample script to download and maintain 10 audit data backups
rem Update the following parameters

SET url=https://example.oraclecloud.com
SET user=ServiceAdmin
SET password=Example.epw
SET AuditFileName="AuditBackup"
SET NumberOfBackups=10

rem EPM Automate commands
call epmautomate login %user% %password% %url%
IF %ERRORLEVEL% NEQ 0 goto :ERROR
    call epmautomate exportAppAudit %AuditFileName% nDays=180
IF %ERRORLEVEL% NEQ 0 goto :ERROR
    call epmautomate downloadfile %AuditFileName%.zip
IF %ERRORLEVEL% NEQ 0 goto :ERROR
    call epmautomate logout
IF %ERRORLEVEL% NEQ 0 goto :ERROR

rem Rename downloaded audit data backup, keep the last 10 backups
Set Timestamp=%date:~4,2%_%date:~7,2%_%date:~10,2%%
Set Second=%time:~0,2%%time:~3,2%
ren %AuditFileName%.zip %AuditFileName%_%Timestamp%_%Second%.zip
```

```

SET Count=0
FOR %%A IN (%AuditFileName%*.*) DO SET /A Count += 1
IF %Count% gtr %NumberOfBackups% FOR %%A IN (%AuditFileName%*.*) DO
del "%%A" && GOTO EOF
:EOF

echo Scheduled Task Completed successfully
exit /b %errorlevel%
:ERROR
echo Failed with error #%errorlevel%.
exit /b %errorlevel%

```

Linux-Skript

Erstellen Sie ein Shell-Skript (z.B. AuditExport.sh), das ähnlich wie das folgende ist, um das Exportieren und Herunterladen von Auditdaten auf einen lokalen Computer zu automatisieren.

```

#!/bin/sh
# Sample script to export, download and maintain 10 audit data backups
# Update the following seven parameters
url=https://example.oraclecloud.com
user=serviceAdmin
password=/home/user1/epmautomate/bin/example.epw
auditfilename="AuditBackup"
numberofbackups=10
epmautomatescript=/home/user1/epmautomate/bin/epmautomate.sh
javahome=/home/user1/jdk1.8.0_191/

export JAVA_HOME=${javahome}

printResult()
{
    op="$1"
    opoutput="$2"
    returncode="$3"

    if [ "${returncode}" -ne 0 ]
    then
        echo "Command failed. Error code: ${returncode}. ${opoutput}"
    else
        echo "${opoutput}"
    fi
}

processCommand()
{
    op="$1"
    date=`date`

    echo "Running ${epmautomatescript} ${op}"
    operationoutput=`eval "$epmautomatescript $op"`
    printResult "$op" "$operationoutput" "$?"
}

```

```

op="login ${user} ${password} ${url}"
processCommand "${op}"
op="exportAppAudit \"${auditfilename}\" -nDays=180"
processCommand "${op}"
op="downloadfile \"${auditfilename}.zip\""
processCommand "${op}"
op="logout"
processCommand "${op}"
# Rename the downloaded audit data backup, keep the last 10 backups
timestamp=`date +%m_%d_%Y_%I%M`
mv "${auditfilename}.zip" "${auditfilename}_${timestamp}.zip"

((numberofbackups+=1))
ls -tp ${auditfilename}*.zip | grep -v '/$' | tail -n +${numberofbackups} |
xargs -d '\n' -r rm --

```

Eine Datendatei in eine Umgebung hochladen und eine Dataload-Regel ausführen

Verwenden Sie diese Skripte , um eine Datei in eine Umgebung hochzuladen, und führen Sie dann eine Datenregel aus, um Daten aus der Datei in eine Anwendung zu importieren.

Voraussetzungen

- Die folgenden Definitionen in Data Management:
 - Eine Dataload-Regeldefinition namens `VisionActual`. Es wird davon ausgegangen, dass die Datenregel keinen Dateipfad für die Eingabedatei angibt.
 - Periodendefinitionen `Mar-15` bis `Jun-15`
- Eine ordnungsgemäß formatierte Datendatei (`GLActual.dat`), die Daten enthält.

Um Daten zu importieren und Dataload-Regeln auszuführen, müssen Sie Befehle ausführen, die die folgenden Schritte abschließen:

- Anmeldung bei der Umgebung
- Laden Sie eine Datei `GLActual.dat`, die Daten für die Perioden `Mar-15` bis `Jun-15` enthält, in den Data Management-Ordner `inbox/Vision` hoch.
- Importieren Sie Daten aus `GLActual.dat` in Data Management mit der Dataload-Regel `VisionActual`, Startperiode `Mar-15`, Endperiode `Jun-15` und Importmodus `REPLACE`.
- Exportieren Sie Daten mit der Option `STORE_DATA`, um die Daten in der Staging-Tabelle der Komponente Datenmanagement mit vorhandenen Anwendungsdaten zusammenzuführen.
- Melden Sie sich ab.

Windows-Beispielskript

Erstellen Sie die Datei `runDataLoadRule.ps1`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```

$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -
raw)
$username="${inputproperties.username}"

```

```

$passwordfile="$($inputproperties.passwordfile)"
$serviceURL="$($inputproperties.serviceURL)"
$dataFile="$($inputproperties.dataFile)"
$dataRuleName="$($inputproperties.dataRuleName)"
$startPeriod="$($inputproperties.startPeriod)"
$endPeriod="$($inputproperties.endPeriod)"
$importMode="$($inputproperties.importMode)"
$exportMode="$($inputproperties.exportMode)"

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile ${datafile} ${dataFileUploadLocation}
epmautomate rundatarule ${dataRuleName} ${startPeriod} ${endPeriod} $
{importMode} ${exportMode} ${dataFileUploadLocation}/${dataFile}
epmautomate logout
    
```

Linux-/UNIX-Beispielskript

Erstellen Sie die Datei `runDataLoadRule.sh`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```

#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "${
serviceURL}"
${epmautomatescript} uploadfile "${datafile}" "${
dataFileUploadLocation}"
${epmautomatescript} rundatarule "${dataRuleName}" "${startPeriod}" "${
endPeriod}" "${importMode}" "${exportMode}" "${
dataFileUploadLocation}/${dataFile}"
${epmautomatescript} logout
    
```

Datei "input.properties" erstellen

Erstellen Sie die Datei `input.properties`, indem Sie eines der folgenden Skripte kopieren, und aktualisieren Sie sie mit Informationen für Ihre Umgebung. Speichern Sie die Datei im selben Verzeichnis, in dem die Datei `runDataLoadRule.ps1` oder `runDataLoadRule.sh` gespeichert ist.

Windows

```

username=serviceAdmin
passwordfile=./password.epw
serviceURL=https://example.oraclecloud.com
dataFile=GLActual.dat
dataFileUploadLocation=UPLOAD_LOCATION
dataRuleName=RULE_NAME
startPeriod=START_PERIOD
endPeriod=END_PERIOD
importMode=IMPORT_MODE
exportMode=EXPORT_MODE
    
```

Linux/UNIX

```

javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURLdataFile=GLActual.dat
dataFileUploadLocation=UPLOAD_LOCATION
dataRuleName=RULE_NAME
startPeriod=START_PERIOD
endPeriod=END_PERIOD
importMode=IMPORT_MODE
exportMode=EXPORT_MODE

```

Tabelle 3-21 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis JAVA_HOME. Nur für Linux/UNIX.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (epmautomate.sh). Nur für Linux/UNIX.
username	Benutzername eines Serviceadministrators, der auch über die Rolle Identitätsdomainadministrator verfügt.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
serviceURL	URL der Umgebung, aus der Sie den Snapshot generieren möchten.
dataFile	Die Datei, die die Daten enthält, die mit der Datenregel importiert werden sollen.
dataFileUploadLocation	Speicherort, an den die Datendatei hochgeladen werden soll.
dataRuleName	Name einer in der Komponente Datenintegration definierten Dataload-Regel.
startPeriod	Erste Periode, für die Daten geladen werden sollen. Dieser Periodenname muss in der Periodenzuordnung der Komponente Datenintegration definiert sein.
endPeriod	Bei Dataloads für mehrere Perioden ist dies die letzte Periode, für die Daten geladen werden sollen. Bei Dataloads für eine einzelne Periode ist dies dieselbe Periode wie die Startperiode. Dieser Periodenname muss in der Periodenzuordnung der Komponente Datenintegration definiert sein.
importMode	Modus des Datenimports in die Komponente Datenintegration. Verwenden Sie die Werte APPEND, REPLACE oder RECALCULATE. Verwenden Sie den Wert NONE, um den Datenimport in Staging-Tabellen zu überspringen.

Tabelle 3-21 (Fortsetzung) Parameter für "input.properties"

Parameter	Beschreibung
exportMode	<p>Modus des Datenexports in die Anwendung. Verwenden Sie die Komponente Datenintegration. Verwenden Sie die Werte STORE_DATA, ADD_DATA, SUBTRACT_DATA oder REPLACE_DATA. Verwenden Sie den Wert NONE, um den Datenexport aus der Komponente Datenintegration in die Anwendung zu überspringen.</p> <div style="border: 1px solid #0070c0; padding: 10px; margin-top: 20px;"> <p> Hinweis:</p> <p>Financial Consolidation and Close unterstützt nur die Modi MERGE und NONE.</p> </div>

Skripte ausführen

1. Erstellen Sie die Datei `runDataLoadRule.ps1` oder `runDataLoadRule.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
2. Erstellen Sie die Datei `input.properties`, und speichern Sie sie im selben Verzeichnis, in dem sich das Skript `runDataLoadRule` befindet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Informationen hierzu finden Sie unter [Datei "input.properties" erstellen](#). Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.
3. Starten Sie das Skript.
 - **Windows PowerShell:** Führen Sie die Datei `runDataLoadRule.ps1` aus.
 - **Linux/UNIX:** Führen Sie die Datei `./runDataLoadRule.sh` aus.

Integration von täglichen Daten automatisieren

In diesem Szenario wird die Verwendung eines Beispielskripts zur regelmäßigen Automatisierung der Datenintegration untersucht.

Um Integrationsaktivitäten zu automatisieren, erstellen Sie eine Batch- (.bat) oder Shell-Datei (.sh) mit einem ähnlichen Skript wie das folgende. Das folgende Beispielskript für Windows automatisiert die Integration von täglichen Anwendungsdaten, indem die folgenden Aktivitäten abgeschlossen werden:

- Melden Sie sich bei einer Umgebung an.
- Löschen Sie `DailyPlanData`, falls vorhanden.
- Laden Sie `DailyPlanData` in den Service hoch.
- Führen Sie die Geschäftsregel `Clear Plan Targets` (Planziele löschen) für den Plantyp `Plan1` aus.

- Importieren Sie Daten mit dem Jobnamen `LoadDailyPlan`.
- Führen Sie die Geschäftsregel `Balance Sheet - Plan` (Bilanz - Plan) aus.
- Führen Sie die Geschäftsregel `Allocate Plan Targets` (Planziele umlegen) aus.
- Löschen Sie `DailyTarget.zip`, falls vorhanden.
- Exportieren Sie Daten mit dem Jobnamen `ExportDailyTarget` in die Datei `DailyTarget.zip`.
- Laden Sie `DailyTarget.zip` auf Ihren Server herunter, und hängen Sie den Zeitstempel an.
- Melden Sie sich von der Umgebung ab.

 **Hinweis:**

Wenn Sie dieses Skript an Ihre Zwecke anpassen möchten, müssen Sie die Werte für die Parameter `SET url` und `SET user` ändern. Zusätzlich können Sie die Werte für die Parameter `dataimportfilename`, `dataexportfilename`, `importdatajobname`, `exportdatajobname`, `br_clear`, `br_calculatebalancesheet` und `br_allocatetarget` an Ihre Bedürfnisse anpassen.

Informationen zum Planen von Skripten mit der Windows-Aufgabenplanung finden Sie unter [Skriptausführung automatisieren](#).

```
@echo off

rem Sample Script to demonstrate daily data integration with
rem EPM Cloud application.
rem This script uploads Plan data, clears target numbers,
rem runs a business rule to calculate balance sheet data, and
rem recalculates target numbers on the Vision demo application

rem Please update these parameters
SET url=https://example.oraclecloud.com
SET user=serviceAdmin
SET dataimportfilename=DailyPlanData.csv
SET dataexportfilename=DailyTarget
SET importdatajobname=LoadDailyPlan
SET exportdatajobname=ExportDailyTarget
SET br_clear=Clear Plan Targets
SET br_calculatebalancesheet=Balance Sheet - Plan
SET br_allocatetarget=Allocate Plan Targets

SET password=%1

rem Executing EPM Automate commands

CD /D %~dp0
call epmautomate login %user% %password% %url%
IF %ERRORLEVEL% NEQ 0 goto :ERROR
```



```

for /f %i in ('call epmautomate listfiles') do if %
%i==%dataimportfilename% (call epmautomate deletefile %i)
IF %ERRORLEVEL% NEQ 0 goto :ERROR

call epmautomate uploadfile %dataimportfilename%
IF %ERRORLEVEL% NEQ 0 goto :ERROR

call epmautomate runbusinessrule "%br_clear%"
IF %ERRORLEVEL% NEQ 0 goto :ERROR

call epmautomate importdata "%importdatajobname%"
IF %ERRORLEVEL% NEQ 0 goto :ERROR

call epmautomate runbusinessrule "%br_calculatebalancesheet%"
IF %ERRORLEVEL% NEQ 0 goto :ERROR

call epmautomate runbusinessrule "%br_allocatetarget%"
"TargetVersion=Baseline"
IF %ERRORLEVEL% NEQ 0 goto :ERROR

for /f %i in ('call epmautomate listfiles') do if %
%i=="%dataexportfilename%.zip" (call epmautomate deletefile %i)
IF %ERRORLEVEL% NEQ 0 goto :ERROR

call epmautomate exportdata %exportdatajobname%
"%dataexportfilename%.zip"
IF %ERRORLEVEL% NEQ 0 goto :ERROR

call epmautomate downloadfile "%dataexportfilename%.zip"
IF %ERRORLEVEL% NEQ 0 goto :ERROR

rem Section to rename the file

Set Timestamp=%date:~4,2%_%date:~7,2%_%date:~10,4%_%time:~1,1%
%time:~3,2%%
ren "%dataexportfilename%.zip" "%dataexportfilename%_Timestamp%.zip"

call epmautomate logout
IF %ERRORLEVEL% NEQ 0 goto :ERROR

:EOF
echo Scheduled Task Completed successfully
exit /b %errorlevel%

:ERROR
echo Failed with error #%errorlevel%.
exit /b %errorlevel%

```

Beispielszenarios für Account Reconciliation

Siehe auch:

- [Vorformatierte Salden in eine Periode laden](#)
Verwenden Sie diese Skripte zum Importieren von zugeordneten Daten aus einer hochgeladenen Datei in eine Account Reconciliation-Umgebung.
- [Backup-Snapshots hochladen und importieren](#)
Verwenden Sie diese Skripte, um einen Backup-Snapshot hochzuladen und in eine Account Reconciliation-Umgebung zu importieren.
- [Alte abgeglichene Transaktionen archivieren und archivierte Transaktionen löschen](#)
Verwenden Sie die Skripte in diesem Abschnitt, um abgeglichene Transaktionen einschließlich Bestätigungs- und Anpassungsdetails, die mindestens dem angegebenen Alter entsprechen, zu archivieren. Löschen Sie die archivierten Transaktionen anschließend aus Account Reconciliation. Die archivierten abgeglichenen Transaktionen werden in einer ZIP-Datei erfasst.

Vorformatierte Salden in eine Periode laden

Verwenden Sie diese Skripte zum Importieren von zugeordneten Daten aus einer hochgeladenen Datei in eine Account Reconciliation-Umgebung.

Windows-Beispielskript

Erstellen Sie eine Datei namens `runPreformattedBalances.ps1`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -
raw)
$username="$($inputproperties.username)"
$passwordfile="$($inputproperties.passwordfile)"
$serviceURL="$($inputproperties.serviceURL)"
$dataFile="$($inputproperties.dataFile)"
$period="$($inputproperties.period)"
$balanceType="$($inputproperties.balanceType)"
$currencyBucket="$($inputproperties.currencyBucket)"

$elements=$dataFile.split('/')
$dataFileName=$elements[-1]

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile ${dataFile}
epmautomate importpremappedbalances ${period} ${dataFileName} $
{balanceType} ${currencyBucket}
epmautomate deletefile ${dataFileName}
epmautomate logout
```

Linux-/UNIX-Beispielskript

Erstellen Sie eine Datei namens `runPreformattedBalances.sh`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
#!/bin/bash

. ./input.properties

export JAVA_HOME=${javahome}

dataFileName=$(echo "${dataFile}" | rev | cut -d'/' -f1 | rev)

${epmautomatescript} login "${username}" "${passwordfile}" "${serviceURL}"
${epmautomatescript} uploadfile "${dataFile}"
${epmautomatescript} importpremappedbalances "${period}" "${dataFileName}" "${balanceType}" "${currencyBucket}"
${epmautomatescript} deletefile "${dataFileName}"
${epmautomatescript} logout
```

Beispiel für die Datei "input.properties"

Um die `runPreformattedBalances`-Skripte auszuführen, erstellen Sie die Datei `input.properties`, und aktualisieren Sie sie mit den Informationen für Ihre Umgebung. Speichern Sie die Datei im selben Verzeichnis, in dem die Datei `runPreformattedBalances.sh` oder `runPreformattedBalances.ps1` gespeichert ist.

Windows

```
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
dataFile=DATA_FILE_NAME.csv
period=PERIOD_NAME
balanceType=BALANCE_TYPE
currencyBucket=CURRENCY_BUCKET
```

Linux/UNIX

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
dataFile=DATA_FILE_NAME.csv
period=PERIOD_NAME
balanceType=BALANCE_TYPE
currencyBucket=CURRENCY_BUCKET
```

Tabelle 3-22 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis JAVA_HOME. Nur für Linux/UNIX.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (epmautomate.sh). Nur für Linux/UNIX.
username	Benutzername eines Serviceadministrators.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
serviceURL	URL der Umgebung, die die Anwendung hostet, in die Sie die vorformatierten Salden laden möchten.
dataFile	Eine CSV-Datei, die die vorformatierten Salden enthält (in der Regel aus einem Hauptbuch erstellt), die Sie in die Anwendung laden möchten. Diese Datei muss bereits mit dem Befehl uploadFile in die Umgebung hochgeladen worden sein.
period	Die Abstimmungsperiode, in die die vorformatierten Salden hochgeladen werden sollen.
balanceType	Der Typ der im Parameter dataFile enthaltenen vorformatierten Salden.
currencyBucket	Die Währungsgruppe für die vorformatierten Salden.

Skripte ausführen

1. Erstellen Sie die Datei `runPreformattedBalances.ps1` oder `runPreformattedBalances.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
2. **Nur für Windows und Linux/UNIX**
 - Erstellen Sie die Datei `input.properties`, und speichern Sie sie im selben Verzeichnis, in dem sich das Skript `runPreformattedBalances` befindet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Informationen hierzu finden Sie in [Tabelle 1](#).
Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.
 - Starten Sie das Skript.
 - **Windows PowerShell:** Führen Sie die Datei `runPreformattedBalances.ps1` aus.
 - **Linux/UNIX:** Führen Sie die Datei `./runPreformattedBalances.sh` aus.

Backup-Snapshots hochladen und importieren

Verwenden Sie diese Skripte, um einen Backup-Snapshot hochzuladen und in eine Account Reconciliation-Umgebung zu importieren.

Windows-Beispielskript

Erstellen Sie eine Datei namens `importBackupSnapshot.ps1`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
$inputproperties = ConvertFrom-StringData (Get-Content ./
input.properties -raw)
$username="$($inputproperties.username) "
$passwordfile="$($inputproperties.passwordfile) "
$serviceURL="$($inputproperties.serviceURL) "
$snapshotName="$($inputproperties.snapshotName) "
$userPassword="$($inputproperties.userPassword) "

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile ${snapshotName}.zip
epmautomate importsnapshot ${snapshotName} "userPassword=$
{userPassword} "
epmautomate deletefile ${snapshotName}.zip
epmautomate logout
```

Linux-/UNIX-Beispielskript

Erstellen Sie eine Datei namens `importBackupSnapshot.sh`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
#!/bin/bash

. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "$
{serviceURL}"
${epmautomatescript} uploadfile "${snapshotName}.zip"
${epmautomatescript} importsnapshot "${snapshotName}" "userPassword=$
{userPassword}"
${epmautomatescript} deletefile "${snapshotName}.zip"
${epmautomatescript} logout
```

Beispiel für die Datei "input.properties"

Um die `importBackupSnapshot`-Skripte auszuführen, erstellen Sie die Datei `input.properties`, und aktualisieren Sie sie mit den Informationen für Ihre Umgebung. Speichern Sie die Datei im selben Verzeichnis, in dem die Datei `importBackupSnapshot.sh` oder `importBackupSnapshot.ps1` gespeichert ist.

Windows

```
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
```

```
snapshotName=SNAPSHOT_NAME
userPassword=IDM_NEW_USER_PWD
```

Linux/UNIX

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
snapshotName=SNAPSHOT_NAME
userPassword=IDM_NEW_USER_PWD
```

Tabelle 3-23 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis JAVA_HOME. Nur für Linux/UNIX.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (epmautomate.sh). Nur für Linux/UNIX.
username	Benutzername eines Serviceadministrators.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
serviceURL	URL der Umgebung, in die Sie den Snapshot importieren möchten.
snapshotName	Der Name des Snapshots, aus dem Artefakte und Daten importiert werden sollen. Dieser Snapshot muss bereits mit dem Befehl uploadFile in die Umgebung hochgeladen worden sein.
userPassword	Das Standardkennwort, das allen neuen Benutzern zugewiesen werden muss, die infolge eines Snapshot-Imports in der Identitätsdomain erstellt werden.

Skripte ausführen

1. Erstellen Sie die Datei `importBackupSnapshot.ps1` oder `importBackupSnapshot.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
2. Erstellen Sie die Datei `input.properties`, und speichern Sie sie im selben Verzeichnis, in dem sich das Skript `runPreformattedBalances` befindet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Informationen hierzu finden Sie in [Beispiel für die Datei "input.properties"](#).
Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.
3. Starten Sie das Skript.
 - **Windows PowerShell:** Führen Sie die Datei `importBackupSnapshot.ps1` aus.
 - **Linux/UNIX:** Führen Sie die Datei `./importBackupSnapshot.sh` aus.

Alte abgeglichene Transaktionen archivieren und archivierte Transaktionen löschen

Verwenden Sie die Skripte in diesem Abschnitt, um abgeglichene Transaktionen einschließlich Bestätigungs- und Anpassungsdetails, die mindestens dem angegebenen Alter entsprechen, zu archivieren. Löschen Sie die archivierten Transaktionen anschließend aus Account Reconciliation. Die archivierten abgeglichenen Transaktionen werden in einer ZIP-Datei erfasst.

Funktionsweise des Skripts

1. Mit den Informationen in der Datei `input.properties` wird die Anmeldung in der Umgebung durchgeführt.
2. Der folgende `archiveTmTransactions`-Befehl wird ausgeführt, um ein Archiv zu erstellen. Die daraus resultierende ZIP-Datei erhält standardmäßig den Namen `Archive_Transactions_INTERCO_JOB_ID.zip`, und die daraus resultierende Logdatei erhält standardmäßig den Namen `Archive_Transactions_INTERCO_JOB_ID.log`.

```
epmautomate archiveTmTransactions INTERCO 365 filterOperator=contains  
filterValue=14001
```

Sie können die Befehlsparameter durch Änderung der Datei `input.properties` ändern.

3. Die Logdatei und die ZIP-Datei mit den archivierten Transaktionen werden auf den lokalen Computer heruntergeladen. Wenn keine übereinstimmenden Transaktionen gefunden werden, zeigt das Skript eine Fehlermeldung an.
4. Die ZIP-Datei mit den archivierten Transaktionen wird in den Oracle Object Store kopiert.
5. Der Befehl `purgeArchivedTmTransactions` (mit der Job-ID des `archiveTmTransactions`-Jobs) wird ausgeführt, um die archivierten abgeglichenen Transaktionen aus der Anwendung zu löschen.

Skripte ausführen

1. Erstellen Sie die Datei `input.properties`, und aktualisieren Sie sie mit Informationen für Ihre Umgebung. Speichern Sie die Datei in einem lokalen Verzeichnis. Dieses Verzeichnis wird in dieser Erörterung als `parentsnapshotdirectory` bezeichnet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem.
Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um Skripte ausführen zu können.
2. Erstellen Sie das Skript `transaction_match.ps1` (Windows PowerShell) oder `transaction_match.sh` (Linux/UNIX), und speichern Sie es im Verzeichnis `parentsnapshotdirectory`, in dem sich die Datei `input.properties` befindet.
3. Starten Sie das Skript.
 - Linux/UNIX: Führen Sie die Datei `./transaction_match.sh` aus.
 - Windows PowerShell: Führen Sie die Datei `transaction_match.ps1` aus.

Skript "input.properties" erstellen

Erstellen Sie das Skript `input.properties`, indem Sie das folgende Skript kopieren und aktualisieren.

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
epmusername=exampleAdmin1
epmpassword=examplePassword1.epw
epmurl=exampleURL1
objectstorageusername=exampleAdmin2
objectstoragepassword=examplePassword2
objectstorageurl=exampleURL2
matchtype=INTERCO
age=365
filteroperator=contains
filtervalues=FilterValue=14001
proxyserverusername=myProxyserver
proxyserverpassword=myProxyserver_pwd
proxyserverdomain=myProxyDomain
```

Hinweis:

Nur für Windows: Entfernen Sie die folgenden Eigenschaften aus der Datei `input.properties`:

- `javahome=JAVA_HOME`
- `epmautomatescript=EPM_AUTOMATE_LOCATION`

Wenn `authentication at proxy server` für Ihre Windows-Netzwerkumgebung nicht aktiviert ist, entfernen Sie die folgenden Eigenschaften aus der Datei `input.properties`.

- `proxyserverusername`
- `proxyserverpassword`
- `proxyserverdomain`

Tabelle 3-24 Parameter für "input.properties"

Parameter	Beschreibung
<code>javahome</code>	Das Verzeichnis, in dem das von EPM Automate verwendete JDK installiert ist. Löschen Sie diesen Eintrag aus der Windows-Version der Datei <code>input.properties</code> . Beispiel: <code>javahome=./home/JDK/bin</code>
<code>epmautomatescript</code>	Das Verzeichnis, in dem EPM Automate installiert ist. Löschen Sie diesen Eintrag aus der Windows-Version der Datei <code>input.properties</code> . Beispiel: <code>epmautomatescript=./home/Utils/EPMAutomate/bin</code>

Tabelle 3-24 (Fortsetzung) Parameter für "input.properties"

Parameter	Beschreibung
epmusername	Der Benutzername eines Serviceadministrators oder eines Powerusers, Benutzers oder Leseberechtigten, der autorisiert ist, abgegliche Transaktionen zu archivieren. Beispiel: epusername=ServiceAdmin
epmuserpassword	Die Datei mit dem verschlüsselten Kennwort für den Benutzer epusername. Beispiel: epmpassword=myPwd.epw
epmurl	Die URL der Umgebung, in der abgegliche Transaktionen archiviert werden sollen. Beispiel: epmurl=https://test-cloudpln.pbcs.us1.oraclecloud.com
objectstorageusername	Die ID eines Benutzers, der in Oracle Object Storage Cloud über die erforderlichen Schreibzugriffsberechtigungen verfügt. Bei im föderierten Identitätsprovider erstellten Benutzern geben Sie den vollqualifizierten Namen des Benutzers an (Beispiel: exampleIdP/jdoe oder exampleIdP/john.doe@example.com, wobei exampleIdP der Name des föderierten Identitätsproviders ist). Geben Sie bei anderen Benutzern die Benutzer-ID an. Beispiel: epusername=myIdP/jdoe
objectstoragepassword	Das dem Benutzer objectstorageusername zugeordnete Swift-Kennwort oder -Authentifizierungstoken. Dieses Kennwort stimmt nicht mit dem Kennwort überein, das der Benutzer für die Anmeldung bei der Object Storage-Konsole verwendet. Das Authentifizierungstoken ist ein von Oracle generiertes Token, mit dem Sie Drittanbieter-APIs authentifizieren können, beispielsweise für die Authentifizierung mit einem Swift-Client. Anweisungen zum Erstellen dieses Token finden Sie unter So erstellen Sie ein Authentifizierungstoken : in der Dokumentation <i>Oracle Cloud Infrastructure-Dokumentation</i> . Beispiel: objectstoragepassword=jDoe_PWD
objectstorageurl	Die URL der Oracle Object Storage Cloud-Gruppe mit einem optional angehängten Objektnamen. Beispiel: objectstorageurl=https://swiftobjectstorage.region_identifier.oraclecloud.com/v1/namespace/MT_Archives/2023_archives
matchtype	Die ID (TextID) des Abgleichstyps, von dem die abgeglichenen Transaktionen archiviert werden sollen. Beispiel: matchtype=cashrecon
age	Die Anzahl von Tagen, seit die Transaktion abgeglichen wurde. Abgegliche Transaktionen, die diesem Wert entsprechen oder älter sind, werden archiviert. Beispiel: age=180

Tabelle 3-24 (Fortsetzung) Parameter für "input.properties"

Parameter	Beschreibung
filteroperator	Die Filterbedingung zur Identifizierung der Konten, die abgeglichene Transaktionen zur Archivierung enthalten. Der Wert muss einer der folgenden sein: equals, not_equals, starts_with, ends_with, contains oder not_contains. Dieser Wert wird mit filterValue kombiniert, um die Konten zu ermitteln, aus denen abgeglichene Transaktionen archiviert werden sollen. Beispiel: filteroperator=not_equals
filtervalues	Ein oder mehrere Filterwerte zur Identifizierung der zu archivierenden Transaktionen. Wenn equals oder not_equals als filterOperator angegeben ist, können Sie eine durch Leerzeichen getrennte Liste verwenden, um mehrere Werte anzugeben. Wenn mehrere Werte angegeben werden, werden Transaktionen aus Konten zum Archivieren ausgewählt, die zu einer beliebigen Kombination aus Filteroperator und Filterwert passen. Beispiel: filterValue=101-120 filterValue=140-202
proxyserverusername	Der Benutzername zum Authentifizieren einer sicheren Session auf dem Proxyserver, der den Zugriff auf das Internet steuert. Beispiel: proxyserverusername=myProxyserver
proxyserverpassword	Das Kennwort zur Authentifizierung des Benutzers beim Proxyserver. Beispiel: proxyserverpassword=myProxyserver_pwd
proxyserverdomain	Der Name der Domain, die für den Proxyserver definiert ist. Beispiel: proxyserverdomain=myProxyDomain

Windows-Skript

Erstellen Sie die Datei transaction_match.ps1, indem Sie das folgende Skript kopieren. Speichern Sie sie im selben Ordner, in dem die Datei input.properties gespeichert ist.

```
# Archive and Purge Transaction Matching PowerShell script

$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -
raw)
$epmusername="$($inputproperties.epmusername) "
$epmpassword="$($inputproperties.epmpassword) "
$epmurl="$($inputproperties.epmurl) "
$objectstorageusername="$($inputproperties.objectstorageusername) "
$objectstoragepassword="$($inputproperties.objectstoragepassword) "
$objectstorageurl="$($inputproperties.objectstorageurl) "
$matchtype="$($inputproperties.matchtype) "
$age="$($inputproperties.age) "
$filteroperator="$($inputproperties.filteroperator) "
$filtervalues="$($inputproperties.filtervalues) "
$proxyserverusername="$($inputproperties.proxyserverusername) "
$proxyserverpassword="$($inputproperties.proxyserverpassword) "
$proxyserverdomain="$($inputproperties.proxyserverdomain) "
```

```

echo "Running processes to archive and purge transaction matching
transactions ..."
if (${proxyserverusername}) {
    echo "Running epmautomate login ${epmusername} ${epmpassword} $
{epmurl} ${proxyserverusername} ${proxyserverpassword} $
{proxyserverdomain}"
    epmautomate login ${epmusername} ${epmpassword} ${epmurl} $
{proxyserverusername} ${proxyserverpassword} ${proxyserverdomain}
} else {
    echo "Running epmautomate login ${epmusername} ${epmpassword} $
{epmurl}"
    epmautomate login ${epmusername} ${epmpassword} ${epmurl}
}
echo "Running epmautomate archiveTmTransactions \"${matchtype}\" $
{age} filterOperator=${filteroperator} ${filtervalues}"
epmautomate archiveTmTransactions "${matchtype}" "${age}"
"filterOperator=${filteroperator}" "${filtervalues}" > ./
outfile.txt.tmp

$jobIdLine=Select-String -Path "outfile.txt.tmp" -Pattern "Job
ID"; $jobIdLine=($jobIdLine -split ":")[-2]; $jobid=($jobIdLine -split
" ")[1];
$logFile="Archive_Transactions_${matchtype}_${jobid}.log"
$transactionsfile="Archive_Transactions_${matchtype}_${jobid}.zip"
epmautomate listfiles > ./files.txt.tmp
$transactionslogfound=Select-String -Path "./files.txt.tmp" -Pattern "$
{logfile}"
$transactionsfilefound=Select-String -Path "./files.txt.tmp" -Pattern
"${transactionsfile}"
rm ./outfile.txt.tmp
rm ./files.txt.tmp

if (${transactionslogfound}) {
    echo "Running epmautomate downloadfile \"${logfile}\""
    epmautomate downloadfile "${logfile}"
    if (${transactionsfilefound}) {
        echo "Running epmautomate downloadfile ${transactionsfile}"
        epmautomate downloadfile "${transactionsfile}"
        if ($?) {
            echo "Running epmautomate copyToObjectStorage $
{transactionsfile} ${objectstorageusername} ${objectstoragepassword} $
{objectstorageurl}"
            epmautomate copyToObjectStorage "${transactionsfile}" "$
{objectstorageusername}" "${objectstoragepassword}" "$
{objectstorageurl}"
            if ($?) {
                echo "Running epmautomate purgeArchivedTMTransactions
JOBID=${jobid}"
                epmautomate purgeArchivedTMTransactions "JobID=$
{jobid}"
            }
        }
    } else {
        echo "EPMAutomate copyToObjectStorage failed. Purging
of archived matched transactions will not be attempted."
    }
}

```

```

    }
    else {
        echo "Download of transactions file ${transactionsfile} failed.
Copy to object storage, and purging of archived matched transactions will
not be attempted."
    }
}
else {
    echo "No matched transactions found. Archive file download, copy to
object storage, and purging of archived matched transactions will not be
attempted."
}
}
else {
    echo "Matchtype ID \"${matchtype}\" not found. Archive file download,
copy to object storage, and purging of archived matched transactions will
not be attempted."
}

echo "Running epmautomate logout"
epmautomate logout
echo "Script processing completed."

```

Linux-/UNIX-Skript

Erstellen Sie die Datei `transaction_match.sh`, indem Sie das folgende Skript kopieren. Speichern Sie sie im selben Ordner, in dem die Datei `input.properties` gespeichert ist.

```

#!/bin/sh

. ./input.properties
export JAVA_HOME=${javahome}

echo "Running processes to archive and purge transaction matching
transactions..."
if [[ "${proxyserverusername}" != "" ]]
then
    echo "Running epmautomate login ${epmusername} ${epmpassword} ${epmurl}
ProxyServerUserName=${proxyserverusername} ProxyServerPassword=${
proxyserverpassword} ProxyServerDomain=${proxyserverdomain}"
    ${epmautomatescript} login "${epmusername}" "${epmpassword}" "${epmurl}"
    "ProxyServerUserName=${proxyserverusername}" "ProxyServerPassword=${
proxyserverpassword}" "ProxyServerDomain=${proxyserverdomain}"
else
    echo "Running epmautomate login ${epmusername} ${epmpassword} ${epmurl}"
    ${epmautomatescript} login "${epmusername}" "${epmpassword}" "${epmurl}"
fi
echo "Running epmautomate archiveTmTransactions \"${matchtype}\" ${age}
filterOperator=${filteroperator} ${filtervalues}"
${epmautomatescript} archiveTmTransactions "${matchtype}" "${age}"
"filterOperator=${filteroperator}" "${filtervalues}" > ./output.txt.tmp

jobid=$(grep "Job ID" ./output.txt.tmp | cut -d':' -f3 | cut -d' ' -f2)
logfile="Archive_Transactions_${matchtype}_${jobid}.log"
transactionsfile="Archive_Transactions_${matchtype}_${jobid}.zip"

```

```

${epmautomatescript} listfiles > ./files.txt.tmp
transactionslogfound=$(grep "${logfile}" ./files.txt.tmp | wc -l)
transactionsfilefound=$(grep "${transactionsfile}" ./files.txt.tmp |
wc -l)
rm ./files.txt.tmp
rm ./output.txt.tmp

if [ ${transactionslogfound} -eq 0 ]
then
    echo "Matchtype ID \"${matchtype}\" not found. Archive file
download, copy to object storage, and purging of archived matched
transactions will not be attempted."
else
    echo "Running epmautomate downloadfile \"${logfile}\""
    ${epmautomatescript} downloadfile "${logfile}"
    if [ ${transactionsfilefound} -eq 0 ]
    then
        echo "No matched transactions found. Archive file download,
copy to object storage, and purging of archived matched transactions
will not be attempted."
    else
        echo "Running epmautomate downloadfile ${transactionsfile}"
        ${epmautomatescript} downloadfile "${transactionsfile}"
        if [ $? -eq 0 ]
        then
            echo "Running epmautomate copyToObjectStorage $
{transactionsfile} ${objectstorageusername} ${objectstoragepassword} $
{objectstorageurl}"
            ${epmautomatescript} copyToObjectStorage "$
{transactionsfile}" "${objectstorageusername}" "${
objectstoragepassword}" "${objectstorageurl}"
            if [ $? -eq 0 ]
            then
                echo "Running epmautomate purgeArchivedTMTransactions
JOBID=${jobid}"
                ${epmautomatescript} purgeArchivedTMTransactions
"JobID=${jobid}"
            else
                echo "EPMAutomate copyToObjectStorage failed. Purging
of archived matched transactions will not be attempted."
            fi
        else
            echo "Download of transactions file ${transactionsfile}
failed. Copy to object storage, and purging of archived matched
transactions will not be attempted."
        fi
    fi
fi

echo "Running epmautomate logout"
${epmautomatescript} logout
echo "Script processing completed."

```

Beispielszenarios für Profitability and Cost Management

Diese Szenarios untersuchen die Verwendung der EPM Automate-Befehle zum Ausführen einiger gängiger Profitability and Cost Management-Aufgaben.

Siehe auch:

- [Metadaten in Anwendungen importieren](#)
Verwenden Sie diese Skripte, um eine Metadatenfile hochzuladen und Dimensionsmetadaten daraus in eine Profitability and Cost Management-Anwendung zu importieren.
- [Daten importieren und Programmregeln ausführen](#)
Verwenden Sie diese Skripte, um Datendateien hochzuladen und Daten aus den hochgeladenen Dateien in einen Profitability and Cost Management-Geschäftsprozess zu importieren.

Metadaten in Anwendungen importieren

Verwenden Sie diese Skripte, um eine Metadatenfile hochzuladen und Dimensionsmetadaten daraus in eine Profitability and Cost Management-Anwendung zu importieren.

Diese Skripte führen die folgenden Vorgänge aus:

- Anmeldung bei der Umgebung
- Hochladen einer Metadatenfile
- Import von Metadaten aus der hochgeladenen Datei in eine Anwendung
- Aktivierung der Anwendung
- Abmeldung

Windows-Skript

Erstellen Sie die Datei `replicateusers.ps1`, indem Sie das folgende Skript kopieren.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -
raw)
$username="$($inputproperties.username)"
$passwordfile="$($inputproperties.passwordfile)"
$serviceURL="$($inputproperties.serviceURL)"
$applicationName="$($inputproperties.applicationName)"
$dataFileName="$($inputproperties.dataFileName)"
$dataFileNameDestination="$($inputproperties.dataFileNameDestination)"

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile "${dataFileName}" ${dataFileNameDestination}
epmautomate loadaddimdata ${applicationName} dataFileName=${dataFileName}
epmautomate enableapp ${applicationName}
epmautomate logout
```

Linux-/UNIX-Skript

Erstellen Sie die Datei `importMetadata.sh`, indem Sie das folgende Skript kopieren.

```
#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "${serviceURL}"
${epmautomatescript} uploadfile "${dataFileName}" "${dataFileNameDestination}"
${epmautomatescript} loadaddimdata "${applicationName}" "dataFileName=${dataFileName}"
${epmautomatescript} enableapp "${applicationName}"
${epmautomatescript} logout
```

Datei "input.properties" erstellen

Um die `importMetadata`-Skripte auszuführen, erstellen Sie die Datei `input.properties`, und aktualisieren Sie sie mit den Informationen für Ihre Umgebung. Speichern Sie die Datei im selben Verzeichnis, in dem die Datei `importMetadata.psl` oder `importMetadata.sh` gespeichert ist.

Windows

```
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
applicationName=APPLICATION_NAME
dataFileName=DATA_FILE.txt
dataFileNameDestination=profitinbox
```

Linux/UNIX

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
applicationName=APPLICATION_NAME
dataFileName=DATA_FILE.txt
dataFileNameDestination=profitinbox
```

Tabelle 3-25 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis <code>JAVA_HOME</code> . Nur für Linux/UNIX.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (<code>epmautomate.sh</code>). Nur für Linux/UNIX.
username	Benutzername eines Serviceadministrators, der auch über die Rolle Identitätsdomainadministrator verfügt.

Tabelle 3-25 (Fortsetzung) Parameter für "input.properties"

Parameter	Beschreibung
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
serviceURL	URL der Umgebung, aus der Sie den Snapshot generieren möchten.
applicationName	Name der Profitability and Cost Management-Anwendung, in die Daten geladen werden sollen.
dataFileName	Name der Datei, die die zu importierenden Metadaten enthält.
dataFileNameDestination	Speicherort zum Hochladen der Metadatendatei.

Skripte ausführen

1. Erstellen Sie die Datei `importMetadata.ps1` oder `importMetadata.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
2. Erstellen Sie die Datei `input.properties`, und speichern Sie sie im selben Verzeichnis, in dem sich das Skript `importMetadata` befindet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Informationen hierzu finden Sie unter [Datei "input.properties" erstellen](#).
Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.
3. Starten Sie das Skript.
 - **Windows PowerShell:** Führen Sie die Datei `importMetadata.ps1` aus.
 - **Linux/UNIX:** Führen Sie die Datei `./importMetadata.sh` aus.

Daten importieren und Programmregeln ausführen

Verwenden Sie diese Skripte, um Datendateien hochzuladen und Daten aus den hochgeladenen Dateien in einen Profitability and Cost Management-Geschäftsprozess zu importieren.

Diese Skripte führen die folgenden Schritte aus:

- Anmeldung bei der Umgebung
- Upload einer Datendatei mit den zu ladenden Daten
- Upload einer Regeldatei mit Datenregeln
- Laden von Daten aus der Datendatei in die Anwendung, um vorhandene Werte zu überschreiben
- Ausführen aller Regeln in der Regeldatei
- Abmeldung

Windows-Skript

Erstellen Sie eine Datei namens `importData.ps1`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./
input.properties -raw)
$username="$($inputproperties.username) "
$passwordfile="$($inputproperties.passwordfile) "
$serviceURL="$($inputproperties.serviceURL) "
$applicationName="$($inputproperties.applicationName) "
$dataFileName="$($inputproperties.dataFileName) "
$rulesFileName="$($inputproperties.rulesFileName) "
$fileDestination="$($inputproperties.fileDestination) "
$clearAllDataFlag="$($inputproperties.clearAllDataFlag) "
$dataLoadValue="$($inputproperties.dataLoadValue) "

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile "${dataFileName}" ${fileDestination}
epmautomate uploadfile "${rulesFileName}" ${fileDestination}
epmautomate loaddata ${applicationName} clearAllDataFlag=${
clearAllDataFlag} dataLoadValue=${dataLoadValue} rulesFileName=${
rulesFileName} dataFileName="${dataFileName}"
epmautomate logout
```

Linux-/UNIX-Skript

Erstellen Sie eine Datei namens `importData.sh`, indem Sie das folgende Skript kopieren. Speichern Sie die Datei in einem lokalen Verzeichnis.

```
#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "$
{serviceURL}"
${epmautomatescript} uploadfile "${dataFileName}" "${fileDestination}"
${epmautomatescript} uploadfile "${rulesFileName}" "${fileDestination}"
${epmautomatescript} loaddata "${applicationName}" "clearAllDataFlag=${
clearAllDataFlag}" "dataLoadValue=${dataLoadValue}" rulesFileName=${
rulesFileName}" dataFileName="${dataFileName}"
${epmautomatescript} logout
```

Datei "input.properties" erstellen

Um die `importData`-Skripte auszuführen, erstellen Sie die Datei `input.properties`, und aktualisieren Sie sie mit den Informationen für Ihre Umgebung. Speichern Sie die Datei im selben Verzeichnis, in dem die Datei `importData.ps1` oder `importData.sh` gespeichert ist.

Windows

```
username=exampleAdmin
passwordfile=examplePassword.epw
```

```

serviceURL=exampleURL
applicationName=APPLICATION_NAME
dataFileName=DATA_FILE.txt
rulesFileName=RULE_FILE.txt
fileDestination=profitinbox
clearAllDataFlag=true
dataLoadValue=OVERWRITE_EXISTING_VALUES

```

Linux/UNIX

```

javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
applicationName=APPLICATION_NAME
dataFileName=DATA_FILE.txt
rulesFileName=RULE_FILE.txt
fileDestination=profitinbox
clearAllDataFlag=true
dataLoadValue=OVERWRITE_EXISTING_VALUES

```

Tabelle 3-26 Parameter für "input.properties"

Parameter	Beschreibung
javahome	Verzeichnis JAVA_HOME. Nur für Linux/UNIX.
epmautomatescript	Absoluter Pfad der ausführbaren Datei für EPM Automate (epmautomate.sh). Nur für Linux/UNIX.
username	Benutzername eines Serviceadministrators, der auch über die Rolle Identitätsdomainadministrator verfügt.
password	Kennwort des Serviceadministrators oder Name und Speicherort der verschlüsselten Kennwortdatei.
serviceURL	URL der Umgebung, aus der Sie den Snapshot generieren möchten.
applicationName	Name der Profitability and Cost Management-Anwendung, in die Daten geladen werden sollen.
dataFileName	Name der Datei, die die zu importierenden Daten enthält.
rulesFileName	Name der Datei, die die Regeln enthält, die für die importierten Daten ausgeführt werden sollen.
fileDestination	Speicherort, an den die Daten und Regeldateien hochgeladen werden sollen.
clearAllDataFlag	Gibt an, ob vorhandene Daten im Anwendungs-Cube gelöscht werden sollen. Wenn Sie nicht möchten, dass vorhandene Daten gelöscht werden, setzen Sie diesen Parameter auf false.
dataLoadValue	Gibt an, wie vorhandene Daten behandelt werden sollen. Wenn Sie möchten, dass vorhandene Daten im Cube beibehalten werden, setzen Sie diesen Parameter auf ADD_TO_EXISTING.

Skripte ausführen

1. Erstellen Sie die Datei `importData.ps1` oder `importData.sh`, indem Sie das Skript aus einem vorhergehenden Abschnitt kopieren.
2. Erstellen Sie die Datei `input.properties`, und speichern Sie sie im selben Verzeichnis, in dem sich das Skript `importData` befindet. Die Inhalte dieser Datei unterscheiden sich je nach Betriebssystem. Informationen hierzu finden Sie unter [Datei "input.properties" erstellen](#).
Stellen Sie sicher, dass Sie Schreibrechte in diesem Verzeichnis haben. Unter Windows müssen Sie möglicherweise PowerShell mit der Option **Als Administrator ausführen** starten, um das Skript ausführen zu können.
3. Starten Sie das Skript.
 - **Windows PowerShell:** Führen Sie die Datei `importData.ps1` aus.
 - **Linux/UNIX:** Führen Sie die Datei `./importData.sh` aus.

Beispielszenarios für Oracle Enterprise Data Management Cloud

In diesen Beispielszenarios wird die Verwendung von EPM Automate-Befehlen zum Synchronisieren von Anwendungsdimensionen zwischen Oracle Enterprise Data Management Cloud und Oracle Enterprise Performance Management Cloud untersucht.

Siehe auch:

- [Dimensionen und Zuordnungen von Oracle Enterprise Data Management Cloud mit EPM Cloud-Anwendungen synchronisieren](#)
In diesem Beispielszenario wird das Synchronisieren einer Dimension zwischen einer Oracle Enterprise Data Management Cloud-Anwendung und einer Oracle Enterprise Performance Management Cloud-Anwendung untersucht.
- [EPM Cloud-Dimensionen mit Oracle Enterprise Data Management Cloud-Anwendungen synchronisieren](#)
In diesem Beispielszenario wird das Synchronisieren einer Dimension zwischen einer Oracle Enterprise Performance Management Cloud-Anwendung und einer Oracle Enterprise Data Management Cloud-Anwendung untersucht.

Dimensionen und Zuordnungen von Oracle Enterprise Data Management Cloud mit EPM Cloud-Anwendungen synchronisieren

In diesem Beispielszenario wird das Synchronisieren einer Dimension zwischen einer Oracle Enterprise Data Management Cloud-Anwendung und einer Oracle Enterprise Performance Management Cloud-Anwendung untersucht.

Mit den Skripten in diesem Abschnitt können Sie folgende Aufgaben ausführen:

- Eine Dimension aus einer Oracle Enterprise Data Management Cloud-Anwendung exportieren
- Zuordnungen aus einer Oracle Enterprise Data Management Cloud-Anwendungsdimension exportieren

- Die Exportdateien in eine EPM Cloud-Umgebung kopieren
- Dimensionsmetadaten und -zuordnungen in die EPM Cloud-Anwendung importieren

So synchronisieren Sie eine Dimension und Zuordnungen zwischen einer Oracle Enterprise Data Management Cloud-Anwendung und einer EPM Cloud-Anwendung:

1. Erstellen Sie eine Skriptdatei, indem Sie das folgende Skript kopieren:

```
rem Integration example to sync application dimensions between EDM and
EPM Cloud
rem Windows script for demonstration purposes only; do not use in
production environments

set EDMUSER=userid
set EDMSVR=https://hostname
set EDMPWDFILE=example_EDM
set EDMAPP=appname
set EDMDIM=dimname
set EDMLOC=location

set EPMUSER=userid
set EPMSVR=https://hostname
set EPMIMPJOB=importjobname
set PWDFILE=C:\Oracle\EPM.epw
set DIMFILE=dimension.csv
set MAPFILE=mapping.csv

rem Synchronizing EDM ---> EPM
rem Export Dimension and Mappings from EDM

call epmautomate login %EDMUSER% %EDMPWDFILE% %EDMSVR%
call epmautomate exporthdimension %EDMAPP% %EDMDIM% %DIMFILE%
call epmautomate exporthdimensionmapping %EDMAPP% %EDMDIM% %EDMLOC%
%MAPFILE%
call epmautomate logout

rem Log into the EPM Cloud environment
call epmautomate login %EPMUSER% %PWDFILE% %EPMSVR%

rem Copy exported files from EDM environment to EPM and import metadata
and mappings
call epmautomate copyfilefrominstance %DIMFILE% %EDMUSER% %EDMPWDFILE%
%EDMSVR% inbox/%DIMFILE%
call epmautomate importmetadata %EPMIMPJOB%

call epmautomate copyfilefrominstance %MAPFILE% %EDMUSER% %EDMPWDFILE%
%EDMSVR% inbox/%MAPFILE%
call epmautomate importmapping %EDMDIM% %MAPFILE% REPLACE FALSE %EDMLOC%

call epmautomate logout
```

2. Ändern Sie die Skriptdatei, und legen Sie die erforderlichen Parameterwerte fest. Eine Erläuterung sowie Beispiele für die Parameter finden Sie unter [Parameter für die Skriptausführung](#).

3. Führen Sie das Skript manuell aus oder planen Sie seine Ausführung nach Bedarf. Informationen hierzu finden Sie unter [Skriptausführung automatisieren](#).

Parameter für die Skriptausführung

Für die Skriptdateien in diesem Abschnitt müssen Sie einige der Parameterwerte angeben, die in der folgenden Tabelle beschrieben sind. Nicht alle diese Parameter werden in allen Skripten verwendet.

Tabelle 3-27 Parameterwerte für Skriptdateien

Parameter	Beschreibung
EDMUSER	Benutzeranmelde-ID eines Oracle Enterprise Data Management Cloud-Serviceadministrators. Beispiel: EDMUSER=jdoe@example.com
EDMSVR	URL der Oracle Enterprise Data Management Cloud-Umgebung. Beispiel: EDMSVR=https:// example.oraclecloud.com
EDMPWDFILE	Name und Speicherort der verschlüsselten Kennwortdatei (EPW) des Oracle Enterprise Data Management Cloud-Serviceadministrators. Beispiel: EDMPWDFILE=edm_jdoe.epw
EDMAPP	Name einer Oracle Enterprise Data Management Cloud-Anwendungsdimension. Beispiel: EDMAPP=USOperations
EDMDIM	Name der Dimension, die exportiert oder importiert werden soll. Beispiel: EDMDIM=entity
EDMLOC	Name des Speicherorts für den Export. Beispiel: EDMLOC=Loc1
EPMUSER	Anmeldename eines EPM Cloud-Serviceadministrators. Beispiel: EPMUSER=john.doe@example.com
EPMSVR	URL der EPM Cloud-Umgebung. Beispiel: EPMSVR=https://example.oraclecloud.com
EPMIMPJOB	Name eines vorhandenen Jobs zum Importieren von Metadaten in der EPM Cloud-Umgebung. Beispiel: EPMIMPJOB=imp_DIMMetadata
EPMEXPJOB	Name eines vorhandenen Jobs zum Exportieren von Metadaten in der EPM Cloud-Umgebung. Beispiel: EPMEXPJOB=Exp_DIMMetadata
PWDFILE	Name und Speicherort der verschlüsselten Kennwortdatei (EPW) des EPM Cloud-Serviceadministrators. Informationen hierzu finden Sie im Abschnitt zum Befehl encrypt . Beispiel: PWDFILE=pwd_jdoe.epw
DIMFILE	Name der Datei, in der exportierte Dimensionsdaten gespeichert werden sollen. Beispiel: DIMFILE=entity_file.CSV
MAPFILE	Name der Datei, in der exportierte Zuordnungsdaten gespeichert werden sollen. Beispiel: MAPFILE=map_file.CSV

EPM Cloud-Dimensionen mit Oracle Enterprise Data Management Cloud-Anwendungen synchronisieren

In diesem Beispielszenario wird das Synchronisieren einer Dimension zwischen einer Oracle Enterprise Performance Management Cloud-Anwendung und einer Oracle Enterprise Data Management Cloud-Anwendung untersucht.

Mit den Skripten in diesem Abschnitt können Sie folgende Aufgaben ausführen:

- Metadaten (Dimensionen) aus einer EPM Cloud-Anwendung exportieren
- Exportdateien mit Dimensionsdaten in eine Oracle Enterprise Data Management Cloud-Umgebung kopieren
- Dimensionsmetadaten in die Oracle Enterprise Data Management Cloud-Anwendung importieren

So synchronisieren Sie eine Dimension zwischen einer EPM Cloud-Anwendung und einer Oracle Enterprise Data Management Cloud-Anwendung:

1. Erstellen Sie eine Windows-Skriptdatei, indem Sie das folgende Skript kopieren:

```
rem Integration example to sync an application dimension between EPM
Cloud and EDM
rem Windows script for demonstration purposes only; do not use in
production environments

set EDMUSER=userid
set EDMSVR=https://hostname
set EDMPWDFILE=example_EDM.epw
set EDMAPP=appname
set EDMDIM=dimname

set EPMUSER=userid
set EPMSVR=https://hostname
set PWDFILE=example_epm.epw
set EPMEXPJOB=exportjobname

rem Synchronizing EPM ---> EDM

rem Export Metadata from EPM
call epmautomate login %EPMUSER% %PWDFILE% %EPMSVR%
call epmautomate exportmetadata %EPMEXPJOB%
call epmautomate logout

rem Import Dimension to EDM
rem Log into the EDM environment
call epmautomate login %EDMUSER% %EDMPWDFILE% %EDMSVR%
rem Copy exported metadata file into the EDM environment
call epmautomate copyfilefrominstance %EPMEXPJOB%.zip %EPMUSER% %PWDFILE%
%EPMSVR% %EPMEXPJOB%.zip
call epmautomate importdimension %EDMAPP% %EDMDIM% ReplaceNodes
%EPMEXPJOB%.zip
call epmautomate logout
```

Ändern Sie die Skriptdatei, und legen Sie die erforderlichen Parameterwerte fest. Eine Erläuterung sowie Beispiele für die Parameter finden Sie unter [Parameter für die Skriptausführung](#).

2. Führen Sie das Skript manuell aus oder planen Sie seine Ausführung nach Bedarf. Informationen hierzu finden Sie unter [Skriptausführung automatisieren](#).

Skriptausführung automatisieren

Ein Serviceadministrator plant Skripte in der Windows-Aufgabenplanung oder verwendet einen Cron-Job, um Aktivitäten mit EPM Automate zu automatisieren.

So planen Sie die EPM Automate-Skriptausführung mit der Windows-Aufgabenplanung:

1. Klicken Sie auf **Start, Systemsteuerung, Verwaltung**.
2. Öffnen Sie die **Aufgabenplanung**.
3. Wählen Sie **Aktion, Einfache Aufgabe erstellen** aus.
4. Geben Sie einen Namen und optional eine Beschreibung für die Aufgabe ein, und klicken Sie auf **Weiter**.
5. Wählen Sie unter **Aufgabentrigger** einen Plan für die Ausführung des Skripts aus, und klicken Sie auf **Weiter**.
6. Geben Sie im nächsten Fenster andere Zeitplanparameter an, und klicken Sie auf **Weiter**.
7. Stellen Sie sicher, dass unter "Aktion" die Option **Programm starten** aktiviert ist.
8. Führen Sie unter **Programm starten** die folgenden Schritte aus:
 - a. Wählen Sie über "Durchsuchen" unter **Programm/Skript** das Skript aus, für das Sie die Ausführung planen möchten.
 - b. Geben Sie unter **Argumente hinzufügen (optional)** das Kennwort für den Serviceadministrator ein, der im Skriptparameter `SET user` angegeben ist.
 - c. Geben Sie unter **Starten in (optional)** den Speicherort ein, an dem EPM Automate installiert ist (in der Regel `C:/Oracle/EPMAutomate/bin`).
 - d. Klicken Sie auf **Weiter**.
9. Wählen Sie in der **Zusammenfassung** die Option **Beim Klicken auf "Fertig stellen", die Eigenschaften für diese Aufgabe öffnen** aus, und klicken Sie auf **Fertig stellen**.
10. Wählen Sie unter **Allgemein** die folgenden Sicherheitsoptionen aus, und klicken Sie auf **OK**.
 - **Unabhängig von der Benutzeranmeldung ausführen**
 - **Mit höchsten Privilegien ausführen**

EPM Automate-Aktivitäten überwachen

Um den Status des initialisierten Vorgangs zu bestimmen, zeigt EPM Automate Statuscodes in der Konsole an, über die Sie das Utility ausführen.

Informationen hierzu finden Sie unter [Exit-Codes](#).

Mit der Jobkonsole können Sie die Jobs überwachen, die Sie mit EPM Automate ausführen. Weitere Details finden Sie unter Jobs verwalten in der Dokumentation *Planning verwalten*.

4

Befehle ohne Installation von EPM Automate ausführen

Mit Groovy können Sie ausgewählte Befehle direkt in Oracle Enterprise Performance Management Cloud ausführen. Es ist keine Installation von EPM Automate erforderlich, um serverseitige Befehle auszuführen.



Note:

In diesem Szenario wird das Groovy-Skript so geschrieben, dass es direkt in EPM Cloud und nicht auf einem Clientcomputer ausgeführt wird.

In diesem Kapitel finden Sie Informationen zu Folgendem:

- [Umgebungen, die die serverseitige Befehlsausführung unterstützen](#)
- [Informationsquellen](#)
- [Unterstützte Befehle](#)
- [Methoden zur Ausführung von EPM Automate mit serverseitigen Groovy-Skripten](#)
- [Umgebungen mit einem serverseitigen Groovy-Skript klonen](#)
- [Aktivitätsbericht mit serverseitigem Groovy-Skript per E-Mail versenden](#)

Umgebungen, die die serverseitige Befehlsausführung unterstützen

Die Unterstützung des Groovy-Skripting für die serverseitige Ausführung der EPM Automate-Befehle steht nur in den folgenden Umgebungen zur Verfügung:

- Geschäftsprozesse in Planning und Planning Modules, die in EPM Enterprise Cloud Service-Umgebungen bereitgestellt werden.
- Enterprise Planning and Budgeting Cloud
- Planning and Budgeting Cloud mit Plus One-Option
- FreeForm
- Enterprise Profitability and Cost Management
- Financial Consolidation and Close in EPM Enterprise Cloud Service-Umgebungen.
- Tax Reporting in EPM Enterprise Cloud Service-Umgebungen.
- Strategische Personalplanung
- Sales Planning

Groovy-Skripte, die EPM Automate-Befehle beinhalten, können nur in den oben beschriebenen Oracle Enterprise Performance Management Cloud-Umgebungen geschrieben und ausgeführt werden. Skripte, die in solchen Umgebungen geschrieben wurden, können jedoch EPM Automate-Befehle in beliebigen EPM Cloud-Umgebungen ausgeben. Beispiel: Sie können das Skript in einer Planning EPM Enterprise Cloud Service-Umgebung erstellen und festlegen, dass Befehle in einer Narrative Reporting-Umgebung ausgegeben werden sollen, die kein Groovy-Skripting unterstützt.

Informationsquellen

Ausführliche Informationen hierzu finden Sie in den nachstehenden Quellen in der Dokumentation *Mit Calculation Manager für Oracle Enterprise Performance Management Cloud entwerfen*:

- Informationen zu Groovy-Geschäftsregeln
- Groovy-Geschäftsregeln erstellen

Unterstützte Befehle

Mit Ausnahme der folgenden Befehle können alle EPM Automate-Befehle mit Groovy ausgeführt werden:

- [downloadFile](#)
- [upgrade](#)
- [uploadFile](#)

Die folgenden Befehle können nicht in der Umgebung ausgeführt werden, in der das Groovy-Skript ausgeführt wird:

- [recreate](#)
- [replay](#)
- [resetService](#)
- [restructureCube](#)

Note:

- Bei Ausführung des Befehls [encrypt](#) wird die verschlüsselte Kennwortdatei auf dem Server erstellt. Sie wird nach sieben Tagen ohne Verwendung gelöscht.
- Damit der Befehl [feedback](#) funktioniert, müssen alle als Anhang verwendeten Dateien und Screenshots am Standardspeicherort für Uploads verfügbar sein. Informationen hierzu finden Sie unter [Standardspeicherort zum Hochladen](#). Hierbei handelt es sich um den Speicherort, an dem Dateien gespeichert werden, wenn Sie im Befehl [uploadFile](#) keinen Speicherort angeben.

Methoden zur Ausführung von EPM Automate mit serverseitigen Groovy-Skripten

- `getEPMAutomate ()`: Diese statische Methode stellt eine Instanz der Klasse `EpmAutomate` bereit, die dann zum Aufrufen der EPM Automate-Befehle verwendet werden kann.
- `execute ()`: Diese Methode der Klasse `EpmAutomate` wird zum Ausführen eines EPM Automate-Befehls verwendet. Übergeben Sie den Namen des EPM Automate-Befehls als ersten Parameter und die Befehlsoptionen als nachfolgende Parameter. Diese Methode gibt eine Instanz der Klasse `EpmAutomateStatus` zurück.
- `getStatus ()`: Diese Methode der Klasse `EPMAutomateStatus` gibt den Ausführungsstatus zurück, der von dem Befehl zurückgegeben wird. Der Rückgabewert "0" gibt an, dass der Vorgang erfolgreich war, während andere Werte bedeuten, dass der Befehl nicht erfolgreich war.
- `getOutput ()`: Diese Methode der Klasse `EPMAutomateStatus` gibt die Zeichenfolgenausgabe des Befehls zurück. Beispiel: Mit dieser Methode können Sie die Ausgabe der Befehle `getApplicationAdminMode` und `getDailyMaintenanceStartTime` zurückgeben. Wenn der Rückgabestatus des Befehls nicht null ist, gibt diese Methode eine Fehlermeldung zurück.
- `getItemsList ()`: Diese Methode der Klasse `EPMAutomateStatus` gibt die Listenausgabe des Befehls zurück. Beispiel: Mit dieser Methode können Sie die Ausgabe der Befehle `getSubstVar`, `listBackups` und `listFiles` zurückgeben.

Umgebungen mit einem serverseitigen Groovy-Skript klonen

Sie können EPM Automate-Befehle in serverseitige Groovy-Skripte einschließen, um Umgebungen für Disaster Recovery-Zwecke zu klonen. So können Sie eine Disaster Recovery ohne On-Premise-Footprint einrichten.

Wenn Kennwörter Sonderzeichen enthalten, finden Sie weitere Informationen unter [Sonderzeichen verarbeiten](#). Stellen Sie zudem sicher, dass die folgenden Parameterwerte so ersetzt werden, dass Sie zu Ihren Umgebungen passen:

Table 4-1 Zu ändernde Parameter

Parameter	Beschreibung
<code>password</code>	Das Kennwort des Serviceadministrators, der den Klonvorgang in der Quellumgebung durchführt.
<code>targetpassword</code>	Das Kennwort des Serviceadministrators, der den Klonvorgang in der Zielumgebung durchführt.
<code>username</code>	Die Benutzer-ID des Serviceadministrators in der Quellumgebung.
<code>targetusername</code>	Die Benutzer-ID des Serviceadministrators in der Zielumgebung. Diesem Benutzer muss zudem die Rolle Identitätsdomainadministrator in der Zielumgebung zugewiesen sein.
<code>email_id</code>	Die E-Mail-Adresse, an die Sie Informationen zum Klonprozess senden möchten.

Skript zum Verschlüsseln von Kennwörtern

```
EpmAutomate automate = getEpmAutomate()

//Encrypt the password of a Service Administrator in the source
environment
EpmAutomateStatus encryptstatus1 = automate.execute('encrypt',
'password', 'encryptionKey', 'sourcePassword.epw')
if(encryptstatus1.getStatus() != 0)
throwVetoException(encryptstatus1.getOutput())
println(encryptstatus1.getOutput())

//Encrypt the password of a Service Administrator in the target
environment
//This user must also have the Identity Domain Administrator
//role in the target environment

EpmAutomateStatus encryptstatus2= automate.execute('encrypt',
'targetpassword',
'encryptionKey', 'targetPassword.epw')
if(encryptstatus2.getStatus() != 0)
throwVetoException(encryptstatus2.getOutput())
println(encryptstatus2.getOutput())
```

Skript zum Klonen von Umgebungen

Dieses Skript verwendet die verschlüsselten Kennwortdateien, die mit dem vorherigen Skript erstellt wurden.

```
EpmAutomate automate = getEpmAutomate()

//Log into the target environment
EpmAutomateStatus loginstatus = automate.execute('login',
'username', 'targetPassword.epw' , 'targeturl')
if(loginstatus.getStatus() != 0)
throwVetoException(loginstatus.getOutput())
println(loginstatus.getOutput())

//Recreate the target environment
EpmAutomateStatus recreatestatus = automate.execute('recreate' , '-f' )
if(recreatestatus.getStatus() != 0)
throwVetoException(recreatestatus.getOutput())
println(recreatestatus.getOutput())

//Copy Artifact Snapshot from the source environment
//to the target environment
EpmAutomateStatus copystatus =
automate.execute('copysnapshotfrominstance',
'Artifact Snapshot', 'sourceusername', 'sourcePassword.epw', 'source
url')
if(copystatus.getStatus() != 0)
throwVetoException(copystatus.getOutput())
println(copystatus.getOutput())
```

```
//import Artifact Snapshot into the target environment
EpmAutomateStatus importstatus = automate.execute('importsnapshot',
'Artifact Snapshot')
println(importstatus.getOutput())

//Send an email to a designated user with the status of the
//snapshot import process
EpmAutomateStatus emailstatus = automate.execute('sendmail',
'email_id','Status of DR', 'BODY='+ importstatus.getOutput())
println(emailstatus.getOutput())

//Sign out of the target environment
EpmAutomateStatus logoutstatus = automate.execute('logout')
println(logoutstatus.getOutput())
```

Aktivitätsbericht mit serverseitigem Groovy-Skript per E-Mail versenden

Mit diesem Skript kann der Aktivitätsbericht per E-Mail an eine Empfängerliste gesendet werden. Für dieses Skript kann dann die tägliche Ausführung zum Abrufen des täglichen Aktivitätsberichts geplant werden. Dieses Skript führt die folgenden Funktionen aus:

- Es verschlüsselt das Kennwort des Serviceadministrators, der das Groovy-Skript ausführt.
- Es führt die Anmeldung in der Oracle Enterprise Performance Management Cloud-Umgebung mit dem verschlüsselten Kennwort durch.
- Es sendet den in der Umgebung verfügbaren Aktivitätsbericht per E-Mail an eine Empfängerliste, in der Regel Serviceadministratoren.
- Es führt die Abmeldung von der Umgebung aus.

Wenn Kennwörter Sonderzeichen enthalten, finden Sie weitere Informationen unter [Sonderzeichen verarbeiten](#). Stellen Sie zudem sicher, dass die folgenden Parameterwerte so ersetzt werden, dass Sie zu Ihren Umgebungen passen:

Table 4-2 Zu ändernde Parameter

Parameter	Beschreibung
user	Die Benutzer-ID eines Serviceadministrators für die Anmeldung bei der Umgebung
password	Das Kennwort des Serviceadministrators
url	Die URL der EPM Cloud-Umgebung, in der der Aktivitätsbericht per E-Mail gesendet werden soll
emailaddresses	Eine durch Semikolon getrennte Liste von E-Mail-Adressen, an die der Aktivitätsbericht gesendet werden soll

Ausführliche Informationen zur Verwendung von Groovy-Regeln finden Sie unter Groovy-Regeln verwenden in der Dokumentation *Planning verwalten*.

```
/*RTPS: {user} {password} {url} {emailaddresses}*/
import java.text.SimpleDateFormat
```

```

String user = 'service_administrator'
String password = 'examplePWD'
String url = 'example_EPM_URL'
String emailaddresses = 'service_administrator@oracle.com'

EpmAutomate automate = getEpmAutomate()

def LogMessage(String message) {
    def date = new Date()
    def sdf = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss")
    println('[' + sdf.format(date) + '][GROOVY] ' + message);
}

def LogOperationStatus(EpmAutomateStatus opstatus) {
    def returncode = opstatus.getStatus()
    LogMessage(opstatus.getOutput())
    LogMessage('return code: ' + returncode)
}

LogMessage('Starting mail activity report processing')

// encrypt
LogMessage("Operation: encrypt " + password + " oracleKey
password.epw")
EpmAutomateStatus status =
automate.execute('encrypt',password,"oracleKey","password.epw")
LogOperationStatus(status)

// login
LogMessage("Operation: login " + user + " password.epw " + url)
status = automate.execute('login',user,"password.epw",url)
LogOperationStatus(status)

// listfiles
LogMessage('Operation: listfiles')
status = automate.execute('listfiles')
LogOperationStatus(status)

String filelist = status.getItemsList()
String[] str = filelist.split(',');
String reportfile = ''

for( String svalues : str ) {
    String[] ftr = svalues.split('/')
    for( String fvalues : ftr ) {
        if (fvalues.startsWith('2') && fvalues.endsWith('html')) {
            reportfile = fvalues
        }
    }
}

def reportdir = reportfile.tokenize(".")[0]
String reportpath = 'apr/' + reportdir + '/' + reportfile

// sendMail

```

```
LogMessage('Operation: sendMail ' + emailaddresses + ' Daily Activity Report
Body=Daily Activity Report Attachments=' + reportpath)
status = automate.execute('sendmail',emailaddresses,'Daily Activity
Report','Body=Daily Activity Report',"Attachments=${reportpath}")
LogOperationStatus(status)

// logout
LogMessage('Operation: logout')
status = automate.execute('logout')
LogOperationStatus(status)
```

5

EPM Cloud-Umgebung replizieren

Diese Schritte sind beim Konfigurieren einer Sekundärumgebung von Oracle Enterprise Performance Management Cloud erforderlich, um die Verfügbarkeit des Service sicherzustellen, wenn das primäre Data Center von Oracle aufgrund von unvorhergesehenen Umständen nicht verfügbar ist.



Hinweis:

Die in diesem Anhang erläuterten Verfahren gelten nicht für Narrative Reporting.

- [Tägliche Artefaktreplikation einrichten](#)
- [On-Demand-Replikationen einrichten](#)
- [Sekundärumgebung konfigurieren](#)

Tägliche Replikation einrichten

Verwenden Sie zum Replizieren einer Umgebung EPM Automate, um den während der täglichen Wartung erstellten Artefakt-Snapshot aus der Primärumgebung in die Sekundärumgebung zu kopieren.

Oracle führt für jede Umgebung eine tägliche routinemäßige Wartung durch. Während dieser Servicewartung erstellt Oracle einen Wartungs-Snapshot durch Sichern der Inhalte der Umgebung (vorhandene Daten und Artefakte, einschließlich Benutzer- und Rollenzuweisungen aus der Identitätsdomain).

So richten Sie die tägliche Servicereplikation ein:

1. Erstellen Sie eine Skriptdatei, die die folgenden EPM Automate-Befehle enthält. Dieses Skript repliziert den Anwendungs-Snapshot von der Primärumgebung in der Sekundärumgebung.



Hinweis:

Stellen Sie sicher, dass Sie den Benutzernamen, die Kennwortdatei, die Identitätsdomainnamen und die Service-URLs ändern. Informationen zum Erstellen einer verschlüsselten Kennwortdatei finden Sie im Abschnitt zum Befehl [encrypt](#).

```
REM Sign in to the secondary instance
epmautomate login serviceAdmin secondaryPassword.epw secondary_URL
secondaryDomain
REM Delete the existing artifact snapshot
```



```
epmautomate deletefile "Artifact Snapshot"  
REM Copy the snapshot from the primary instance  
epmautomate copysnapshotfrominstance "Artifact Snapshot"  
primaryPassword.epw primary_URL primaryDomain  
REM Sign out of the secondary instance  
epmautomate logout
```

2. Erstellen Sie mit einem Scheduler (z.B. Windows-Aufgabenplanung) einen Zeitplan für die Ausführung der Skriptdatei, sodass sie zwei Stunden lang nach Start des Wartungsfensters ausgeführt wird.
3. Legen Sie für die Primär- und die Sekundärumgebung eine identische Startzeit für das Wartungsfenster fest. Weitere Informationen finden Sie unter *Wartungszeit für Services* festlegen in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*.

On-Demand-Replikationen einrichten

Zur Reduzierung des RPOs können Sie On-Demand-Snapshots der Primärumgebung erstellen und diese in die Sekundärumgebung kopieren.

Beispiel: Sie können ein EPM Automate-Skript erstellen und so planen, dass es alle sechs Stunden zwischen den täglichen Replikationen ausgeführt wird, um das RPO von 24 auf 6 Stunden zu reduzieren.

Hinweis:

Während der Erstellung eines On-Demand-Snapshots wird die primäre Serviceumgebung einige wenige Minuten lang in den schreibgeschützten Modus versetzt.

So richten Sie On-Demand-Replikationen ein:

1. Erstellen Sie eine Skriptdatei, die die folgenden EPM Automate-Befehle enthält. Dieses Skript repliziert den Anwendungs-Snapshot von der Primärumgebung in der Sekundärumgebung.

Hinweis:

Stellen Sie sicher, dass Sie den Benutzernamen, die Kennwortdatei, die Identitätsdomainnamen und die Service-URLs ändern. Informationen zum Erstellen einer verschlüsselten Kennwortdatei finden Sie im Abschnitt zum Befehl [encrypt](#).

```
REM Sign in to the primary instance  
epmautomate login serviceAdmin primaryPassword.epw primary_URL  
primaryDomain  
REM Create a snapshot and then sign out  
epmautomate exportsnapshot "Artifact Snapshot"  
epmautomate logout
```

```
REM Sign in to the secondary instance
epmautomate login serviceAdmin secondaryPassword.epw secondary_URL
secondaryDomain
REM Copy the snapshot from the primary instance
epmautomate copysnapshotfrominstance "Artifact Snapshot"
primaryPassword.epw primary_URL primaryDomain
REM Sign out of the secondary instance
epmautomate logout
```

2. Erstellen Sie mit einem Scheduler (z.B. Windows-Aufgabenplanung) einen Zeitplan für die Ausführung der Skriptdatei, damit sie so ausgeführt wird, dass das gewünschte RPO erreicht wird.

Sekundärumgebung konfigurieren

Sie konfigurieren die Sekundärumgebung, um sie zu aktivieren.

Führen Sie diese Prozedur nur aus, wenn die Sekundärumgebung aktiviert werden muss, wenn die Primärumgebung längere Zeit nicht verfügbar ist. Lesen Sie vor dem Konfigurieren der Sekundärumgebung die folgenden Themen in der Dokumentation *Erste Schritte mit Oracle Enterprise Performance Management Cloud for Administrators*:

- Migrationspfade für EPM Cloud-Legacy-Snapshots
- Migrationspfade für EPM Standard Cloud Service- und EPM Enterprise Cloud Service-Snapshots

So konfigurieren Sie die Sekundärumgebung:

1. Starten Sie eine EPM Automate-Session, und führen Sie die folgenden Aktivitäten aus.

- Melden Sie sich bei der Sekundärumgebung mit einem Konto an, das die Rollen "Serviceadministrator" und "Identitätsdomainadministrator" aufweist. Stellen Sie sicher, dass Sie den entsprechenden Benutzernamen, das Kennwort, den Domainnamen und die Service-URL angeben.
- Erstellen Sie die Sekundärumgebung neu.
 - Wenn die Primärumgebung eine Planning-, Tax Reporting-, Financial Consolidation and Close- oder Enterprise Profitability and Cost Management-Umgebung ist, verwenden Sie folgenden Befehl:
`epmautomate recreate -f`
 - Wenn die Primärumgebung keine Planning-, Tax Reporting-, Financial Consolidation and Close- oder Enterprise Profitability and Cost Management-Umgebung ist, verwenden Sie folgenden Befehl:
`epmautomate recreate -f TempServiceType=PRIMARY_APPLICATION_TYPE,`
wobei `PRIMARY_APPLICATION_TYPE` den Wert `ARCS`, `EDMCS`, `PCMCS` oder `EPRCS` aufweist.
- Importieren Sie Anwendungs- und Identitätsdomainartefakte aus dem Snapshot.
- Melden Sie sich ab.

Sie können die zuvor aufgeführten Aktivitäten durchführen, indem Sie die folgenden Befehle ausführen. Informationen hierzu finden Sie in den folgenden Themen:

- Befehl [login](#)
- Befehl [recreate](#)

- Befehl `importSnapshot`

```
epmautomate login serviceAdmin secondaryPassword.epw secondary_URL
epmautomate recreate -f
epmautomate importsnapshot "Artifact Snapshot" importUsers=true
epmautomate logout
```

2. Melden Sie sich bei der Sekundärumgebung an, und prüfen Sie, ob alle Daten verfügbar sind.
3. Senden Sie allen Benutzern die URL der Sekundärumgebung.

A

Ausführen des Befehls "simulateConcurrentUsage" vorbereiten

Der Befehl [simulateConcurrentUsage](#) unterstützt die folgenden Vorgänge zur Simulation eines Ladevorgangs in eine Umgebung:

- Formulare öffnen
- Formulare speichern
- Geschäftsregeln ausführen
- Datenregeln ausführen
- Ad-hoc-Raster öffnen
- Bücher mit Managementberichten ausführen
- Berichte mit Managementberichten ausführen

Nötige Schritte

1. Erstellen Sie die Datei `requirement.csv`. Informationen hierzu finden Sie unter [Datei "requirement.csv" erstellen](#).
2. Erstellen Sie Eingabedateien, in denen die Details der in der Datei `requirement.csv` enthaltenen Vorgänge angegeben sind. Informationen hierzu finden Sie unter:
 - [Eingabedatei für Formulare öffnen](#)
 - [Eingabedatei für Formulare speichern](#)
 - [Eingabedatei für Geschäftsregeln ausführen](#)
 - [Eingabedatei für Datenregeln ausführen](#)
 - [Eingabedatei für Ad-hoc-Raster](#)
 - [Eingabedatei für Bücher ausführen](#)
 - [Eingabedatei für Berichte ausführen](#)
3. Erstellen Sie die Datei `UserVarMemberMapping.csv`, die die Details zu den Benutzervariablen enthält. Informationen hierzu finden Sie unter [Datei "UserVarMemberMapping.csv" erstellen](#).
4. Erstellen Sie eine ZIP-Datei, die die Dateien enthält, die Sie in den vorherigen Schritten erstellt haben, und laden Sie sie in die Umgebung hoch. Informationen hierzu finden Sie unter [Eingabe-ZIP-Datei erstellen und in die Umgebung hochladen](#).
5. Führen Sie den Befehl [simulateConcurrentUsage](#) mit der hochgeladenen ZIP-Datei aus.

Datei "requirement.csv" erstellen

Beginnen Sie mit der Erstellung der Datei `requirement.csv`, in der die Details der Anwendungsfälle aufgelistet werden, die Sie testen möchten. In jeder Zeile dieser CSV-Datei wird der Typ des auszuführenden Vorgangs, der Artefaktnamen, die Anzahl gleichzeitiger

Benutzer, die Eingabedatei mit den Details zum Vorgang und eventuell zusätzliche Informationen zum Vorgang angeben. Beispiel: Um 2 Formulare zu öffnen, 2 Formulare zu speichern und 2 Geschäftsregeln auszuführen, geben Sie 6 Zeilen in der Eingabe-CSV-Datei an. Die erste Zeile der Datei `requirement.csv` muss die folgenden Informationen enthalten:

```
#Type of Operation,Artifact Name,Number of Users,Input File,Additional Info
```

Jede der folgenden Zeilen der Datei enthält einen einzelnen Vorgang und die zugehörigen Parameter. Für einige Vorgänge sind nicht alle diese Parameterwerte erforderlich. Die erwarteten Dateieinträge werden in der folgenden Tabelle erläutert.



Note:

Alle Werte sind erforderlich, sofern in der Tabelle nicht anders angegeben.

Table A-1 Format der Datei "requirement.csv"

Feld	Beschreibung
Vorgangstyp	Muss einer der folgenden Typen sein: <ul style="list-style-type: none"> • Open Form • Save Form • Run Business Rule • Run Data Rule • Ad Hoc Grid • Execute Report • Execute Book
Artefaktnamen	Dieser Wert ist abhängig vom Vorgangstyp: <ul style="list-style-type: none"> • Open Form: Name und Speicherort des zu öffnenden Formulars. • Save Form: Name und Speicherort des zu speichernden Formulars. • Run Business Rule: Name der Geschäftsregel. • Data Rule: Name der Datenregel. • Ad Hoc Grid: nicht anwendbar (leer lassen). • Execute Report: Name und Speicherort des Berichts. • Execute Report: Name und Speicherort des Buches.
Anzahl an Benutzern	Anzahl der Benutzer zur Simulation der gleichzeitigen Verwendung.
Eingabedatei	Name der CSV-Datei, die die zu verwendenden POV-Werte, Runtime Prompts oder andere anwendungsfallsspezifische Werte enthält.
Weitere Informationen	Weitere für den Vorgang erforderliche Parameter. Gilt nur für Ad-hoc-Raster. Lassen Sie dieses Feld in anderen Anwendungsfällen leer.

Hinweis: Die Artefaktnamen müssen den Namen in der Anwendung entsprechen, wobei die Groß-/Kleinschreibung berücksichtigt wird.

Beispiel einer `requirement.csv`-Datei:

```
# Type of Operation,Artifact Name,Number of Users,Input File,Additional Info
Open Form, Library/Global Assumption/Revenue Forecast
Assumptions,10,openform_input.csv,
Save Form, Library/Global Assumption/ExchangeRates,5,saveform_input.csv,
Run Business Rule, Run_FinStatement - Copy Budget to Prior Year
Budget,4,runbusinessrule_input.csv,
Run Data Rule, Delimited_file_DL,5,rundatarule_input.csv,
Ad Hoc Grid,,3,runadhocgrid_input.csv,cube=FinStmt
Execute Book,Review Books/Revenue Reports,10,book_input.csv
Execute Report,Review Reports/Executive Report,10,report_input.csv,
```

Eingabedateien erstellen

Für jeden in der Datei `requirement.csv` angegebenen Anwendungsfall muss eine entsprechende Eingabedatei vorhanden sein, in der alle Parameter für dessen Ausführung angegeben sind.

Die Eingabedatei sollte jeweils einen Eintrag für jeden Benutzer enthalten, der in der Datei `requirement.csv` für diesen Anwendungsfall angegeben ist.

Wenn die Anzahl der Einträge in der Eingabedatei geringer ist als die in der Datei `requirement.csv` angegebene Anzahl der gleichzeitigen Benutzer für diesen Anwendungsfall, wiederholt EPM Automate einige Einträge aus der Eingabedatei, um den Anwendungsfall so lange auszuführen, bis der Vorgang für die in der Datei `requirement.csv` angegebene Anzahl von Benutzern ausgeführt wurde.

Beispiel: Angenommen, der Eintrag zum Anwendungsfall in der Datei `requirement.csv` für den Vorgang "Geschäftsregeln ausführen" lautet wie folgt:

```
Run Business Rule, Copy Budget,10,br_input_file.csv,
```

In diesem Fall muss die Datei `br_input_file.csv` 10 Einträge enthalten. Wenn `br_input_file.csv` nur 6 Einträge enthält, verwendet EPM Automate diese für die ersten 6 Benutzer. Für die folgenden 4 Benutzer verwendet EPM Automate erneut die ersten 4 Einträge aus der Datei `br_input_file.csv`.

Wenn die Anzahl der Einträge in der Eingabedatei höher als die Anzahl der für den Anwendungsfall angegebenen Benutzer ist, ignoriert EPM Automate die überzähligen letzten Einträge in der Eingabedatei.

- [Eingabedatei für Formulare öffnen](#)
- [Eingabedatei für Formulare speichern](#)
- [Eingabedatei für Geschäftsregeln ausführen](#)
- [Eingabedatei für Datenregeln ausführen](#)
- [Eingabedatei für Ad-hoc-Raster](#)
- [Eingabedatei für Berichte ausführen](#)
- [Eingabedatei für Bücher ausführen](#)

Eingabedatei für Formulare öffnen

Diese Datei, auf die in der Datei `requirement.csv` zur Unterstützung des Öffnens von Formularen verwiesen wird, enthält POV-Einträge im folgenden Format: `pov=[DIM 1:MEMBER 1],[DIM 2:MEMBER 2],[DIM 3:MEMBER 3], usw.`

Dabei sind DIM 1, DIM 2 usw. die Dimensionsnamen, und MEMBER 1, MEMBER 2 usw. sind die entsprechenden Dimensionselementwerte für den POV.

Eine Beispielseingabedatei:

```
pov=[Account:APL_RATE_AED],[Scenario:Budget],[Years:FY20]
pov=[Account:APL_RATE_AED],[Scenario:Budget],[Years:FY19]
pov=[Account:APL_RATE_AED],[Scenario:Budget],[Years:FY18]
pov=[Account:APL_RATE_AED],[Scenario:Budget],[Years:FY17]
pov=[Account:APL_RATE_AED],[Scenario:Budget],[Years:FY16]
```



Note:

Sie müssen die Datei `UserVarMemberMapping.csv` auch erstellen, wenn die in der Datei `requirement.csv` angegebenen Formulare das Festlegen von Benutzervariablen erfordern. Informationen hierzu finden Sie unter [Datei "UserVarMemberMapping.csv" erstellen](#).

Eingabedatei für Formulare speichern

Diese Datei, auf die in der Datei `requirement.csv` zur Unterstützung des Speicherns von Formularen verwiesen wird, enthält POVs und Zelleingabewerte im folgenden Format:

Eine Beispielseingabedatei:

```
pov=[DIM 1:MEMBER1],[DIM 2:MEMBER2],[DIM 3:MEMBER3],...;cells=[CELL
COLUMN HEADER 1 -> CELL COLUMN HEADER 2 -> CELL COLUMN HEADER 3 ->.. |
CELL ROW HEADER 1-> CELL ROW HEADER 2-> CELL ROW HEADER 3->..| CELL 1
DATA], [CELL COLUMN HEADER 11 -> CELL COLUMN HEADER 22 -> CELL COLUMN
HEADER 33 ->.. | CELL ROW HEADER 11-> CELL ROW HEADER 22-> CELL ROW
HEADER 33->..| CELL 2 DATA]
```

In diesem Beispiel gilt:

- "DIM" ist der Name einer Dimension und "MEMBER" ein Dimensionselementwert.
- "CELL COLUMN HEADER" ist der Name des Spalten-Headers und "CELL ROW HEADER" der Name eines Zeilen-Headers.

Beispiel:

```
pov=[Version View:Working],[Sales Entity:International Sales];
cells=[FY16->x-----x->Pct|P293:Maintenance->4120:Support|1]
```

 **Note:**

Sie müssen die Datei `UserVarMemberMapping.csv` auch erstellen, wenn die in der Datei `requirement.csv` angegebenen Formulare das Festlegen von Benutzervariablen erfordern. Informationen hierzu finden Sie unter [Datei "UserVarMemberMapping.csv" erstellen](#).

Eingabedatei für Geschäftsregeln ausführen

Diese Datei, auf die in der Datei `requirement.csv` zur Unterstützung der Ausführung von Geschäftsregeln verwiesen wird, enthält Laufzeitparameterwerte im folgenden Format:
`rtp=[RTP1:Value1],[RTP2:Value2]` usw.

Wenn für die Geschäftsregel kein Laufzeitparameterwert erforderlich ist, schließen Sie den Ausdruck `rtp=[]` ein. In diesem Beispiel stehen RTP1, RTP2 usw. für die Namen der Runtime Prompts, und VALUE1, VALUE2 usw. geben deren Werte an. Achten Sie darauf, so viele Runtime Prompts wie erforderlich hinzuzufügen.

Eine Beispielseingabedatei:

```
rtp=[Period:Q1],[Entity:USA]
rtp=[Period:Q2],[Entity:USA]
rtp=[Period:Q3],[Entity:USA]
rtp=[Period:Q4],[Entity:USA]
```

Eingabedatei für Datenregeln ausführen

In dieser Datei, auf die in der Datei `requirement.csv` zur Unterstützung der Ausführung von Datenregeln verwiesen wird, muss die Startperiode, die Endperiode, der Importmodus, der Exportmodus und optional ein in der Umgebung verfügbarer Importdateiname angegeben werden. Wenn kein Dateiname angegeben wird, wird der in der Datenregel angegebene Dateiname verwendet. Das Format der einzelnen Zeilen lautet: `startperiod=START PERIOD;endperiod=END PERIOD;importmode=IMPORT_MODE;exportmode=EXPORT_MODE;filename=FILE NAME`

Eine Beispielseingabedatei:

```
startperiod=Dec-15;endperiod=Dec-15;importmode=REPLACE;exportmode=STORE_DATA;
filename=comma_delim_file1.csv
startperiod=Dec-16;endperiod=Dec-16;importmode=REPLACE;exportmode=STORE_DATA;
filename=comma_delim_file2.csv
startperiod=Dec-17;endperiod=Dec-17;importmode=REPLACE;exportmode=STORE_DATA;
filename=comma_delim_file3.csv
startperiod=Dec-18;endperiod=Dec-18;importmode=REPLACE;exportmode=STORE_DATA;
filename=comma_delim_file4.csv
startperiod=Dec-19;endperiod=Dec-19;importmode=REPLACE;exportmode=STORE_DATA;
filename=comma_delim_file5.csv
```


Eingabedatei für Ad-hoc-Raster

In dieser Datei, auf die in der Datei `requirement.csv` zur Unterstützung des Öffnens von Ad-hoc-Rastern verwiesen wird, muss das zu öffnende Raster angegeben werden. Jede Zeile der Datei muss das folgende Format aufweisen:

```
filename=xlsx filename#sheet name; pov=[DIM 1:MEMBER 1],[DIM 2:MEMBER 2]..; rows=[ROW HEADER 1, ROW HEADER 2,..]; cols=[COL HEADER 1, COL HEADER 2,..]
```

Eine Beispieleingabedatei:

```
fileName=dropdown.xlsx#sheet4;pov=[HSP_View:BaseData],[Scenario:Forecast],[Product:No Product],[Entity:Sales Mid-Atlantic];rows = [ Account ]; cols= [Year, Period, Version]
```

Note:

Die Datei mit dem Namen `fileName` in der Eingabe-CSV-Datei muss die Definition des Ad-hoc-Rasters im angegebenen Arbeitsblatt enthalten. Beispiel: In der oben angegebenen Beispieleingabedatei ist `sheet4` aus der Datei `dropdown.xlsx` als Quelle der Definition des Ad-hoc-Rasters angegeben. Diese Excel-Datei muss gemeinsam mit der Datei `requirement.csv` und der Eingabe-CSV-Datei in der Datei `INPUT_FILE.zip` verfügbar sein, die zur Ausführung des Befehls [simulateConcurrentUsage](#) verwendet wird.

Eingabedatei für Berichte ausführen

In dieser Datei, auf die in der Datei `requirement.csv` zur Unterstützung des Öffnens von Managementberichten verwiesen wird, muss der zu öffnende Bericht angegeben werden. Jede Zeile der Datei muss das folgende Format aufweisen:

```
format=[REPORT_FORMAT];globalPov=[POV];prompts=[PROMPT_1],[PROMPT_2]
```

`globalPov` und `prompts` sind optional.

Note:

- Unterstützt werden die Berichtsformate PDF und eingebettetes Format.
- Wenn der Name der `globalPov`-Dimension oder eines ihrer Elemente einen Doppelpunkt (:) oder ein Semikolon (;) enthält, setzen Sie das Escape-Zeichen `\\` davor. Beispiel: Der Dimensionsname `Version:View` muss wie folgt angegeben werden: `Version\\:View`.

Eine Beispieleringabedatei zum Generieren einer PDF aus einem Bericht mit dem globalen POV [Version View:Working],[Sales Entity:International Sales] und den Prompts [Actual;Budget],[Year:2018]:

```
format=pdf;globalPov=[Version View:Working],[Sales Entity:International Sales];prompts=[Actual;Budget],[Year:2018]
```

Eingabedatei für Bücher ausführen

In dieser Datei, auf die in der Datei `requirement.csv` zur Unterstützung des Öffnens von Büchern in Berichten verwiesen wird, muss das zu öffnende Buch angegeben werden. Jede Zeile der Datei muss das folgende Format aufweisen:

```
format=BOOK_FORMAT
```

oder

```
format=BOOK_FORMAT;globalPov=POV
```

Eine Beispieleringabedatei zum Generieren einer PDF aus einem Buch mit dem globalen POV [Version View:Working],[Sales Entity:International Sales]:

```
format=pdf;globalPov=[Version View:Working],[Sales Entity:International Sales]
```

Note:

- Unterstützt werden die Buchformate PDF und XLSX.
- Wenn der Name der globalPov-Dimension oder eines ihrer Elemente einen Doppelpunkt (:) oder ein Semikolon (;) enthält, setzen Sie das Escape-Zeichen \ davor. Beispiel: Der Dimensionsname `Version:View` muss wie folgt angegeben werden: `Version\\:View`.

Datei "UserVarMemberMapping.csv" erstellen

Diese Datei ist erforderlich, wenn die Formulare, die Sie in der Eingabedatei für die Anwendungsfälle "Open Form" oder "Save Form" angegeben haben, die Festlegung von Benutzervariablen erfordern. Diese Datei ist für andere Anwendungsfälle nicht erforderlich.

Die erste Zeile dieser Datei enthält den Header `#Dimension,User Variable,Member`.

Die folgenden Einträge enthalten die Zuordnung von Dimension, Benutzervariable und Dimensionselement.

Beispiel einer `UserVarMemberMapping.csv`-Datei:

```
#Dimension,User Variable,Member
Account,Account View,Revenue Driver Assumptions
```

```
Entity,Entity,No Entity
Entity,Entity View,Total Entity
HSP_View,HSP_View,BaseData
Market Size,Market View,Large Market
Period,Period,Jan
```

Eingabe-ZIP-Datei erstellen und in die Umgebung hochladen

Erstellen Sie mit einem Tool wie 7-Zip eine ZIP-Datei, die die Datei `requirement.csv`, entsprechende Eingabedateien zum Anwendungsfall und die Datei `UserVarMemberMapping.csv` (falls erforderlich) enthält.

Laden Sie diese ZIP-Datei mit dem Befehl `uploadFile` in die Inbox der Umgebung hoch, in der Sie die Simulation ausführen möchten. (Beispiel für die Befehlsyntax: `epmautomate uploadFile "C:/uploads/INPUT_FILE.zip" inbox.`)

Beispiel für einen Bericht zur Simulation der gleichzeitigen Verwendung

Die Simulation der gleichzeitigen Verwendung wird standardmäßig an Benutzer gesendet, die den Befehl `simulateConcurrentUsage` ausführen. Wenn Sie E-Mail-Empfänger angeben, wird der Bericht per E-Mail an die betreffenden Empfänger gesendet.

Simulate Concurrent Usage Report									
Simulate Concurrent Usage report for customer requirements file demo.zip									
Operation #	Operation	Artifact Name	Users	Iterations	Min. Duration	Max. Duration	Avg. Duration	Return Status	
1	Open Form	Set Services Revenue Forecast Assumptions	2	1	00:00:04.57	00:00:04.57	00:00:04.57	Passed	
2	Save Form	Set Services Revenue Forecast Assumptions	3	1	00:00:05.15	00:00:05.15	00:00:05.15	Passed	
3	Run Business Rule	Operating Expense Adj Plan	2	1	00:00:00.74	00:00:00.74	00:00:00.74	Passed	
4	Run Data Rule	DelimitedFileDL	2	1	00:00:03.78	00:00:03.78	00:00:03.78	Passed	
5	Ad Hoc Grid		1	1	00:00:00.52	00:00:00.52	00:00:00.52	Passed	

Dieser Bericht enthält die folgenden Angaben:

Spalte	Beschreibung
Vorgangsnummer	Die Folgenummer des Anwendungsfalls in der Datei <code>requirement.csv</code> .
Vorgang	Der Vorgangstyp, wie in der Datei <code>requirement.csv</code> angegeben.
Artefaktnamen	Der Artefaktnamen, wie in der Datei <code>requirement.csv</code> angegeben.

Spalte	Beschreibung
Benutzer	Die Anzahl der Benutzer, wie in der Datei <code>requirement.csv</code> angegeben.
Iterationen	Wie oft der Anwendungsfall gemäß Angabe im Parameter <code>iterations</code> ausgeführt wurde.
Minstdauer	Die Minstdauer der Ausführung dieses Anwendungsfalls durch einen Benutzer.
Höchstdauer	Die Höchstdauer der Ausführung dieses Anwendungsfalls durch einen Benutzer.
Durchschnittsdauer	Die durchschnittliche Dauer der Ausführung dieses Anwendungsfalls durch einen Benutzer.
Rückgabestatus	Der Status des Anwendungsfalls. <code>Failed</code> wird angezeigt, wenn die Ausführung des Anwendungsfalls nicht erfolgreich war.

B

Ausführen des Wiedergabebefehls vorbereiten

Der Wiedergabebefehl wird verwendet, um die Performance in einer Umgebung unter starker Last zu testen. Dadurch wird geprüft, ob die Benutzererfahrung akzeptabel ist, wenn sich der Service unter einer bestimmten Last befindet. Sie müssen vor dem Durchführen von Belastungstests für Umgebungen einige Schritte ausführen.

In diesem Anhang werden die Schritte beschrieben, die Serviceadministratoren durchführen müssen, bevor der Wiedergabebefehl von EPM Automate ausgeführt wird.

- [Informationen zum Wiedergabebefehl](#)
- [Voraussetzungen](#)
- [HAR-Dateien erstellen](#)
- [Wiedergabedateien erstellen](#)
- [Tracedateien generieren](#)
- [Wiedergabesession - Beispiel](#)

Informationen zum Wiedergabebefehl

Der Wiedergabebefehl gibt die Auslastung von Oracle Smart View for Office, der REST-API von Oracle Enterprise Performance Management Cloud oder von EPM Automate in einer Umgebung wieder, um Performancetests unter starker Last zu ermöglichen. Dadurch wird geprüft, ob die Benutzererfahrung akzeptabel ist, wenn sich die Umgebung unter einer bestimmten Last befindet.

Beispiel: Sie können die Benutzererfahrung unter starker Last in einer Testumgebung testen, um sicherzustellen, dass die Performance gut ist, nachdem Sie die Anwendung aus der Testumgebung in die Produktionsumgebung migriert haben.

Voraussetzungen

Wenn Sie den Befehl mit einer Wiedergabedatei ausführen, führt EPM Automate alle Zeilen in der Wiedergabedatei parallel aus, um den Service auszulasten. So können Sie Tests ausführen und überprüfen, ob die Benutzererfahrung akzeptabel ist, wenn sich der Service unter Last befindet.

- Identifizieren Sie Formulare, für die eine umfangreiche Verarbeitung in der Umgebung erforderlich ist. Dazu eignen sich Formulare mit großen Datenmengen oder Formulare, die komplexe Berechnungen enthalten. Beispiel: Formulare, mit denen Prognosen und Prozesse weitergeleitet werden, die beim Erstellen von Ad-hoc- und statischen Berichten beteiligt sind, können eine starke Auslastung des Service bewirken. Aktivitäten wie das Ausführen von Geschäftsregeln, Berichten und ressourcenintensiven REST-APIs sowie EPM Automate-Befehle (z.B. runBusinessrule, runDataRule, exportData, exportMetadata, restructureCube) können zu einer starken Auslastung der Umgebung führen und somit zu Kandidaten für Belastungstests werden.

- Installieren Sie Fiddler bei Bedarf. EPM Automate erfordert eine Datei des Typs HAR 1.1 (HAR - HTTP-Archiv-Format), die Datensätze für Oracle Smart View for Office, die REST-API von Oracle Enterprise Performance Management Cloud oder die EPM Automate-Interaktion mit der EPM Cloud-Umgebung enthält. Üblicherweise verwenden Sie Fiddler, um die HAR-Datei zu generieren, in der das Log Ihrer Interaktion mit EPM Cloud erfasst wird.
- Führen Sie die Hauptaktivitäten aus, die Sie zuvor identifiziert haben. Verwenden Sie Smart View, um Aktivitäten wie das Öffnen und Speichern von Formularen, das Ausführen von Geschäftsregeln und das Erstellen von Berichten auszuführen. Mit Fiddler können Sie Aktivitätsdetails erfassen und diese in HAR-Dateien exportieren. Führen Sie REST-APIs und EPM Automate-Befehle aus, und sorgen Sie dafür, dass Fiddler die Details erfasst. Details hierzu finden Sie unter [HAR-Dateien erstellen](#).
- Erstellen Sie eine CSV-Datei zur Wiedergabe, die die Zugangsdaten (Benutzernamen und Kennwörter) sowie den Namen der auszuführenden HAR-Dateien aufführt. Jede Zeile in der Datei kann den Benutzernamen und das Kennwort eines eindeutigen Benutzers enthalten, um mehrere gleichzeitige Benutzersessions zu simulieren. Details hierzu finden Sie unter [Wiedergabedateien erstellen](#).
Der Benutzer, dessen Zugangsdaten in einer Zeile zum Ausführen einer HAR-Datei angegeben sind, muss nicht mit dem Benutzer identisch sein, der die Session zum Erstellen der HAR-Datei ausgeführt hat. Allerdings muss dieser Benutzer über die Berechtigungen zum Ausführen dieser Aktivitäten in der Umgebung verfügen.

Detaillierte Anweisungen zum Ausführen des Wiedergabebefehls finden Sie unter [Wiedergabesession - Beispiel](#).

HAR-Dateien erstellen

Die HAR-Datei erfasst Traces von Oracle Smart View for Office, der REST-API oder der EPM Automate-Interaktion mit Oracle Enterprise Performance Management Cloud.

Da Fiddler beim Erstellen der HAR-Dateien Informationen zu jeglichem HTTP(S)-Datenverkehr erfasst, vermeiden Sie Aktivitäten, die Fiddler möglicherweise unnötige Traces hinzufügen.

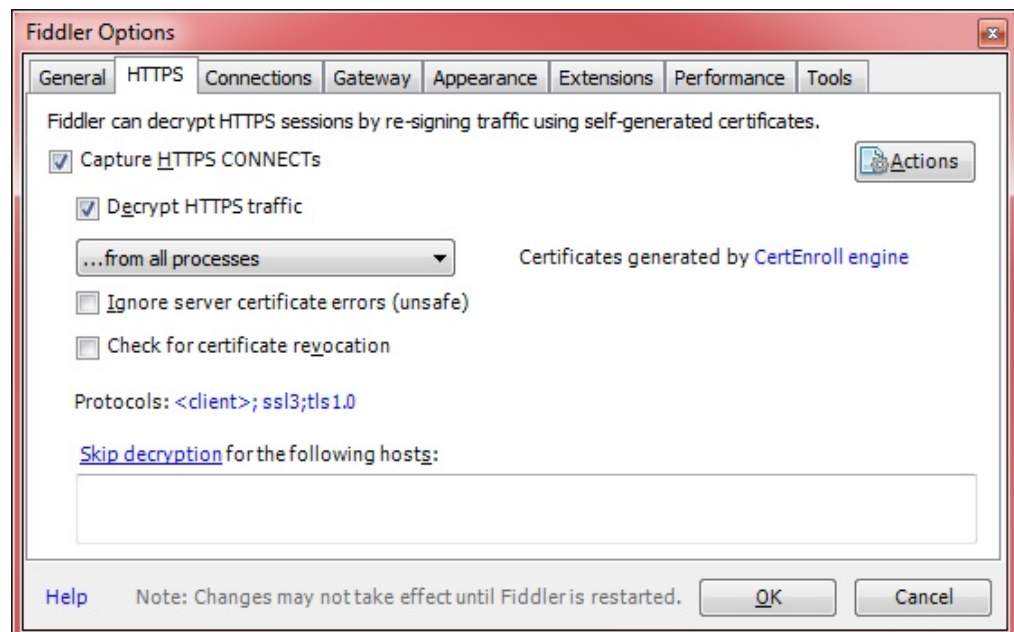
So erstellen Sie eine HAR-Datei:

1. Starten Sie Fiddler.
2. Stellen Sie sicher, dass Fiddler so konfiguriert ist, dass HTTPS-Datenverkehr von allen Prozessen entschlüsselt wird.
 - a. Wählen Sie **Extras, Optionen, HTTPS** aus.
 - b. Wählen Sie **Decrypt HTTPS traffic** (HTTPS-Datenverkehr entschlüsseln) aus, sofern nicht bereits ausgewählt.

Fiddler zeigt Informationen zum Stammzertifikat an, das zum Abfangen von HTTPS-Datenverkehr verwendet wird. Normalerweise kann diesem Zertifikat vertraut werden.

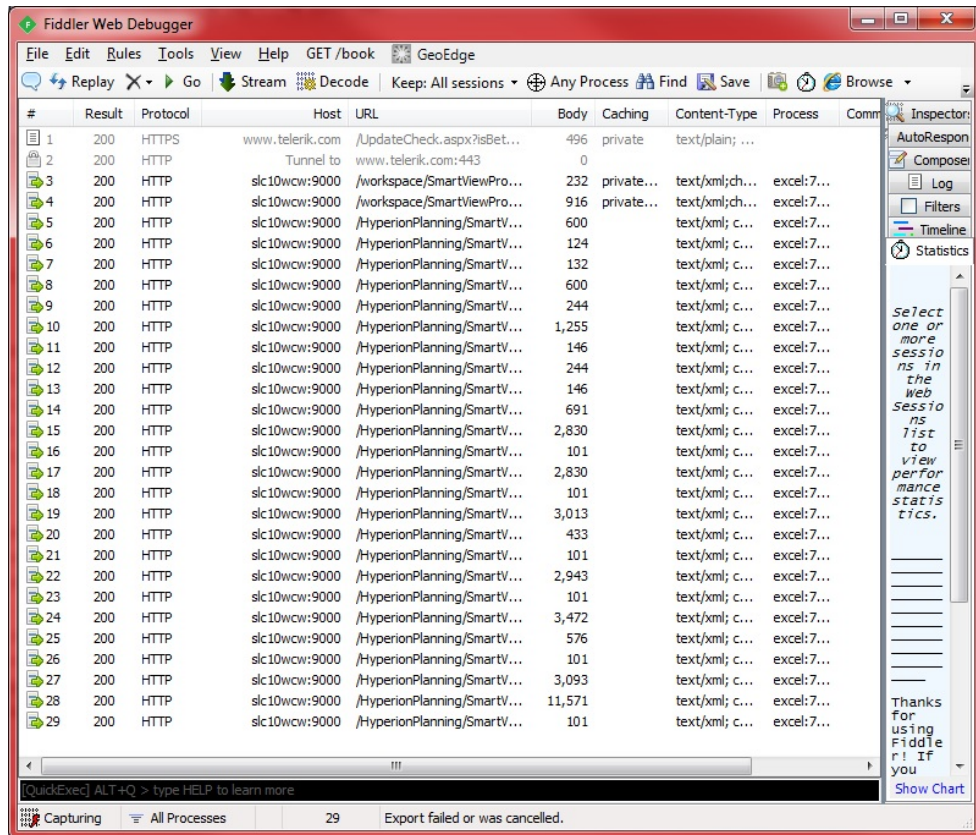


- c. Klicken Sie auf **Yes** (Ja), wenn Sie das Stammzertifikat der Liste der Trusted Certificate Authorities hinzufügen möchten. Wählen Sie andernfalls **No** (Nein) aus.
- d. **Optional:** Wenn Sie im vorherigen Schritt **No** (Nein) ausgewählt haben, können Sie **Ignore server certificate errors** (Serverzertifikatsfehler ignorieren) auswählen, um Sicherheitswarnungen von Fiddler in Bezug auf Entschlüsselung von HTTPS-Datenverkehr zu unterdrücken.
- e. Klicken Sie auf **OK**.



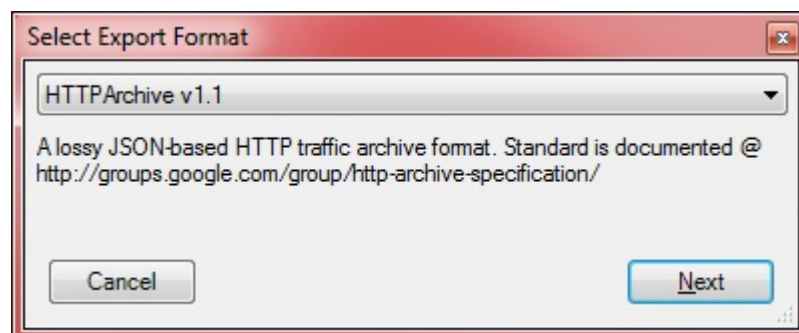
3. Starten Sie Smart View, und greifen Sie auf die Umgebung zu, für die Traces erfasst werden sollen.
4. Mit Smart View, REST-API oder EPM Automate können Sie die Aktivitäten ausführen, die eine starke Auslastung in der Umgebung bewirken. Beispiel: Öffnen Sie Formulare in Smart View, damit Fiddler Ihre Aktivität aufzeichnen kann.

Fiddler zeichnet die von Ihnen initiierten Prozesse auf.



5. Führen Sie in Fiddler die folgenden Schritte aus:

- a. Wählen Sie File (Datei), Export Sessions (Sessions exportieren) und anschließend entweder All Sessions (Alle Sessions) oder Selected Sessions (Ausgewählte Sessions) aus. Wenn Sie Fiddler ausführen und eine Verbindung mit anderen Websites vorhanden ist, wählen Sie Selected Sessions (Ausgewählte Sessions) aus, und wählen Sie die für die Umgebung relevanten Sessions aus.
- b. Wählen Sie unter Select Export Format (Exportformat auswählen) die Option HTTPArchive v1.1 als Exportformat aus.
- c. Klicken Sie auf Next (Weiter).



- d. Wählen Sie unter `Export As HTTPArchive v1.1` (Als HTTPArchive v1.1 exportieren) das Verzeichnis aus, in dem die Datei gespeichert werden soll, und geben Sie einen Dateinamen an.
- e. Klicken Sie auf `Save` (Speichern).

Wiedergabedateien erstellen

Eine Wiedergabedatei ist eine CSV-Datei mit den Zugangsdaten (Benutzername und Kennwort) und mit den Namen der HAR-Dateien, die ausgeführt werden sollen, um das System mit dem Wiedergabebefehl von EPM Automate auszulasten.

Stellen Sie sicher, dass der von Ihnen angegebene Benutzername und das Kennwort über die Berechtigungen zum Ausführen der Aktivitäten in der HAR-Datei verfügen.

Beim Ausführen des Wiedergabebefehls führt EPM Automate alle Zeilen in der Wiedergabedatei parallel aus, um den Service auszulasten. Beispiel: Wenn Ihre Wiedergabedatei 10 Zeilen enthält, gibt EPM Automate 10 Sessions wieder. So können Sie Tests ausführen und überprüfen, ob die Benutzererfahrung akzeptabel ist, wenn sich der Service unter einer bestimmten Last befindet. Jede Aktivität, die in der HAR-Datei enthalten ist, wird seriell ausgeführt.

Informationen zum Ausführen des Wiedergabebefehls finden Sie unter [replay](#) .

So erstellen Sie eine Wiedergabedatei:

1. Öffnen Sie Microsoft Office Excel, und starten Sie ein neues Arbeitsblatt.
2. Geben Sie in den Spalten A, B und C der Zeile 1 einen Benutzernamen, ein Kennwort und den Speicherort einer HAR-Datei ein.

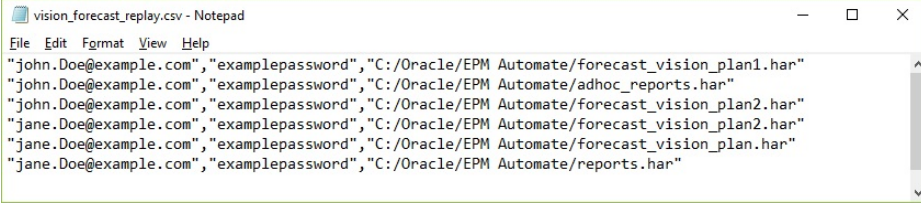
Wiederholen Sie diesen Schritt, um weitere Zeilen zu erstellen.

Hinweis:

Sie müssen den absoluten Pfad zum Speicherort der HAR-Datei angeben. Verwenden Sie Schrägstriche (/) als Verzeichnistrennzeichen in Dateipfaden. Verwenden Sie keine umgekehrten Schrägstriche (\).

3. Speichern Sie die Datei.
4. Führen Sie unter `Speichern` unter die folgenden Schritte aus:
 - a. Wählen Sie das Verzeichnis aus, in dem die Wiedergabedatei gespeichert werden soll.
 - b. Geben Sie unter `Dateiname` einen Namen an, und wählen Sie unter `Dateityp` die Option `CSV` (durch Komma getrennt) (`*.csv`) aus.
 - c. Klicken Sie auf `Save` (Speichern).

Eine Beispielwiedergabedatei kann der folgenden ähneln:



```
vision_forecast_replay.csv - Notepad
File Edit Format View Help
"john.Doe@example.com","examplepassword","C:/Oracle/EPM Automate/forecast_vision_plan1.har"
"john.Doe@example.com","examplepassword","C:/Oracle/EPM Automate/adhoc_reports.har"
"john.Doe@example.com","examplepassword","C:/Oracle/EPM Automate/forecast_vision_plan2.har"
"jane.Doe@example.com","examplepassword","C:/Oracle/EPM Automate/forecast_vision_plan2.har"
"jane.Doe@example.com","examplepassword","C:/Oracle/EPM Automate/forecast_vision_plan.har"
"jane.Doe@example.com","examplepassword","C:/Oracle/EPM Automate/reports.har"
```

Tracedateien generieren

Bei der Ausführung des Wiedergabebefehls können Sie Tracedateien generieren, um sie dem Oracle Support zum Beheben von Problemen freizugeben. Oracle Support vollzieht anhand der Tracedateien nach, wie die Umgebung eine Oracle Smart View for Office-Aktivität verarbeitet hat.

Verwenden Sie den optionalen Parameter `trace=true` mit dem Wiedergabebefehl, um Tracedateien im XML-Format zu generieren. Wenn Sie diesen Parameter verwenden, erstellt EPM Automate für jede Aktivität in der HAR-Datei eine Tracedatei, die die Reaktion von Smart View auf die Aktivität enthält.

Tracedateien weisen den Namen `trace-N.xml` auf. Beispiel: `trace-1.xml`, wobei `N` ein Zähler ist, der bei 1 beginnt. Wenn in der Wiedergabedatei mehrere HAR-Dateien mit demselben Namen angegeben sind, konsolidiert EPM Automate die Tracedateien in einem Ordner.

Tracedateien, die zu einer HAR-Datei gehören, werden in einem Ordner in dem Verzeichnis gespeichert, aus dem EPM Automate ausgeführt wird. EPM Automate erstellt jeweils einen Ordner für jede HAR-Datei, die in der Wiedergabedatei aufgeführt ist. EPM Automate verwendet zur Benennung der Ordner eine Kombination aus der aktuellen Serversystemzeit und dem HAR-Dateinamen im Format `YYYY_MM_DD_HH_MM_SS_HAR_FILE_NAME`. Beispiel: Wenn der HAR-Dateiname `forecast1.har` lautet, lautet der Ordnername möglicherweise `2016_06_08_10_21_42_forecast1`.

Wiedergabesession - Beispiel

Beschreibt, wie Sie den Wiedergabebefehl mit mehreren HAR-Dateien ausführen.

In diesem Abschnitt wird Folgendes vorausgesetzt:

- Sie haben die folgenden HAR-Dateien erstellt: Jede HAR-Datei kann denselben Satz an Aktivitäten enthalten. Ausführliche Informationen finden Sie unter [HAR-Dateien erstellen](#).
 - `C:\Oracle\EPM Automate\forecast_vision_plan1.har`
 - `C:\Oracle\EPM Automate\forecast_vision_plan2.har`
 - `C:\Oracle\EPM Automate\forecast_plan2.har`
- Sie haben die Wiedergabedatei `C:/Oracle/EPM Automate/vision_forecast_replay.csv` mit dem folgenden Inhalt erstellt (Details hierzu finden Sie unter [Wiedergabedateien erstellen](#)):

 **Hinweis:**

Verwenden Sie Schrägstriche (/) als Verzeichnistrennzeichen in Dateipfaden in der Wiedergabedatei. Verwenden Sie keine umgekehrten Schrägstriche (\).

```
john.doe@example.com,examplePwd,C:/Oracle/EPM Automate/  
forecast_vision_plan1.har  
john.doe@example.com,examplePwd,C:/Oracle/EPM Automate/  
forecast_vision_plan2.har  
john.doe@example.com,examplePwd,C:/Oracle/EPM Automate/forecast_plan2.har
```

So führen Sie den Wiedergabebefehl aus:

1. Navigieren Sie in einem Fenster für die Eingabeaufforderung zu dem Verzeichnis (z.B. C:\Oracle\EPM Automate\bin), in dem EPM Automate installiert ist.
2. Melden Sie sich bei der Umgebung als Serviceadministrator an, und führen Sie den Wiedergabebefehl aus:

```
epmautomate login john.doe@example.com examplePassword https://test-cloud-  
pln.pbcs.us1.oraclecloud.com myIdentityDomain
```

```
epmautomate replay "c:/Oracle/EPM Automate/vision_forecast_replay.csv"  
duration=12 lagTime=5.5 trace=true
```

EPM Automate zeigt Wiedergabeinformationen in der Konsole an und beendet die Verarbeitung nach der festgelegten Dauer (im obigen Beispiel 12 Minuten). Darüber hinaus werden Traceordner und -dateien erstellt, da der obige Befehl den Parameter `trace=true` aufweist.

Da der Befehl aus dem Verzeichnis C:\Oracle\EPM Automate\bin ausgeführt wurde, hat EPM Automate die Tracedateien in den folgenden Ordnern gespeichert. Beachten Sie, dass diese Ordner auf der Basis der HAR-Dateinamen benannt sind.

- C:\Oracle\EPM Automate\bin\2017_01_08-12_52_37-forecast_plan2-jdoe@example.com
- C:\Oracle\EPM Automate\bin\2017_01_08-12_52_37-forecast_vision_plan1-jdoe@example.com
- C:\Oracle\EPM Automate\bin\2017_01_08-12_52_37-forecast_vision_plan2-jdoe@example.com

3. Melden Sie sich von der Umgebung ab:

```
epmautomate logout
```

C

Sonderzeichen verarbeiten

Kennwörter, Proxykennwörter und Befehlsparameterwerte von Oracle Enterprise Performance Management Cloud enthalten möglicherweise Sonderzeichen. Eine spezielle Vorgehensweise ist erforderlich, damit EPM Automate solche Zeichen verarbeiten kann.

In den Beispielen in diesem Abschnitt wird ein Beispielkennwort verwendet, um die Verwendung von Sonderzeichen darzustellen.

Oracle empfiehlt, Paare aus Parametern und Werten in doppelte Anführungszeichen zu setzen.

Windows

Diese Sonderzeichen müssen maskiert werden. Setzen Sie hierzu das Sonderzeichen oder den Parameterwert mit dem Sonderzeichen in doppelte Anführungszeichen (").



Hinweis:

EPM Automate kann von keinem Ordner aus ausgeführt werden, dessen Name ein & enthält. Beispiel: C:\Oracle\A&B.

Tabelle C-1 Sonderzeichen verarbeiten: Windows

Zeichen	Beschreibung	Beispiel mit Escape
)	Klammer rechts	<ul style="list-style-type: none">Example") "pwd1 or"Example)pwd1 "
<	Kleiner als	<ul style="list-style-type: none">Example"<"pwd1 or"Example<pwd1 "
>	Größer als	<ul style="list-style-type: none">Example">"pwd1 or"Example>pwd1 "
&	Et-Zeichen	<ul style="list-style-type: none">Example"&"pwd1 or"Example&pwd1 "
	Pipe	<ul style="list-style-type: none">Example" "pwd1 or"Example pwd1 "
"	Anführungszeichen	<ul style="list-style-type: none">Example""pwd1 or"Example"pwd1 "

Ausrufezeichen in Nur-Text-Kennwörtern in Windows-Batchdateien verwenden

Gehen Sie bei Verwendung von Ausrufezeichen (!) in Nur-Text-Kennwörtern in Windows-Batchdateien, die mit EPM Automate verwendet werden, wie folgt vor:

1. Setzen Sie vor dem Ausrufezeichen zwei Caret-Zeichen (^) als Escape-Zeichen. Wenn z.B. das Kennwort Welc0me! lautet, codieren Sie es als Welc0me^^!

2. Aktualisieren Sie die Batchdatei so, dass "DisableDelayedExpansion" mit der folgenden Deklaration am Anfang der Datei steht:

```
setlocal DisableDelayedExpansion
```
3. Entfernen Sie die Deklaration `setlocal EnableExtensions`
`EnableDelayedExpansion`, sofern diese im Skript enthalten ist.

UNIX/Linux

Unter UNIX- und Linux-Betriebssystemen müssen Sonderzeichen mit einem umgekehrten Schrägstrich (\) maskiert werden.

Hinweis:

- Um ! (Ausrufezeichen) zu maskieren, setzen Sie das Kennwort in einfache Anführungszeichen, oder verwenden Sie einen umgekehrten Schrägstrich (\) als Escapezeichen.
- Um \, \$, ' und " zu maskieren, setzen Sie das Kennwort in doppelte Anführungszeichen, oder verwenden Sie den umgekehrten Schrägstrich (\) als Escapezeichen.

Tabelle C-2 Sonderzeichen verarbeiten: UNIX/Linux

Zeichen	Beschreibung	Beispiel mit Escape
(Klammer links	Example \(pwd1
)	Klammer rechts	Example\)pwd1
<	Kleiner als	Example \(<pwd1
>	Größer als	Example \(>pwd1
'	Apostroph	Example \('pwd1
!	Ausrufezeichen	<ul style="list-style-type: none"> • 'Example!pwd1' or • Example \(!pwd1
#	Rautezeichen	Example \(#pwd1
&	Et-Zeichen	Example \(&pwd1
	Pipe	Example \(pwd1
;	Semikolon	Example \(;pwd1
.	Punkt	Example \(.pwd1
"	Anführungszeichen	<ul style="list-style-type: none"> • Example \("pwd1 oder • "Example \("pwd1"
'	Einfaches Anführungszeichen	<ul style="list-style-type: none"> • Example \('pwd1 or • "Example \('pwd1"
\$	Dollarzeichen	<ul style="list-style-type: none"> • Example \(\$pwd1 or • "Example \(\$pwd1"
\	Umgekehrter Schrägstrich	<ul style="list-style-type: none"> • Example \(\(pwd1 or • "Example \(\(pwd1"

Ausrufezeichen in Nur-Text-Kennwörtern in UNIX- oder Linux-Skripten verwenden

Wenn in UNIX/Linux-Skripten ein in einer Shell-Variable gespeichertes EPM Automate-Kennwort Sonderzeichen enthält, verwenden Sie drei umgekehrte Schrägstriche als Escape-Sequenz und setzen Sie sie anschließend in doppelte Anführungszeichen. Beispiel: Das Kennwort `lzi[ACO(e*7Qd)jE`, das in der Shell-Variable `password` enthalten ist, muss folgendermaßen in das Skript geschrieben werden:

```
password="lzi[ACO\\(e*7Qd)jE"
```

D

Spezifische Befehle für jeden EPM Cloud-Service

- [Account Reconciliation-Befehle](#)
- [Befehle für Financial Consolidation and Close](#)
- [Narrative Reporting-Befehle](#)
- [Oracle Enterprise Data Management Cloud-Befehle](#)
- [Befehle für Planning, Planning Modules, FreeForm, Strategische Personalplanung und Sales Planning](#)
- [Profitability and Cost Management-Befehle](#)
- [Enterprise Profitability and Cost Management-Befehle](#)
- [Tax Reporting-Befehle](#)

Account Reconciliation-Befehle

EPM Automate-Befehle für Account Reconciliation

addUsers	groupAssignmentAuditReport	renameSnapshot
addUsersToGroup	help	replay
addUsersToTeam	importARApplicationProperties	resetService
addUserToGroups	importBackgroundImage	restoreBackup
archiveTmTransactions	importLogoImage	roleAssignmentAuditReport
assignRole	importBalances	roleAssignmentReport
cloneEnvironment	importDataManagement	runAutomatch
copyFileFromInstance	importMapping	runBatch
copyFromObjectStorage	importPreMappedBalances	runComplianceReport
copySnapshotFromInstance	importPreMappedTransactions	runDailyMaintenance
copyToObjectStorage	importProfiles	runDataRule
createGroups	importRates	runDMReport
createReconciliations	importRCAAttributeValues	runIntegration
deleteFile	importReconciliationAttributes	runMatchingReport
deleteGroups	importSnapshot	sendMail
downloadFile	importTMAttributeValues	setApplicationAdminMode
encrypt	importTmPremappedTransactions	setDailyMaintenanceStartTime
exportAccessControl	invalidLoginReport	setDemoDates
exportARApplicationProperties	listBackups	setEncryptionKey
exportBackgroundImage	listFiles	setIdleSessionTimeout
exportDataManagement	login	setIPAllowlist
exportLogoImage	logout	setManualDataAccess
exportMapping	provisionReport	setPeriodStatus
exportSnapshot	purgeArchivedTmTransactions	setVirusScanOnFileUploads
feedback	purgeTmTransactions	skipUpdate
getApplicationAdminMode	recreate	unassignRole
getDailyMaintenanceStartTime	refreshCube	updateUsers
getIdleSessionTimeout	removeUserFromGroups	upgrade
getIPAllowlist	removeUsers	uploadFile
getVirusScanOnFileUploads	removeUsersFromGroup	userAuditReport
	removeUsersFromTeam	userGroupReport

Befehle für Financial Consolidation and Close

EPM Automate-Befehle für Financial Consolidation and Close

addUsers	exportTaskManagerAccessControl	renameSnapshot
addUsersToGroup	exportValidIntersections	replay
addUsersToTeam	feedback	resetService
addUserToGroups	getApplicationAdminMode	restoreBackup
applicationAdminMode	getDailyMaintenanceStartTime	restructureCube
assignRole	getEssbaseQryGovExecTime	roleAssignmentAuditReport
clearDataByProfile	getIdleSessionTimeout	roleAssignmentReport
cloneEnvironment	getIPAllowlist	runBatch
copyDataByProfile	getSubstVar	runBusinessRule
copyFileFromInstance	getVirusScanOnFileUploads	runDailyMaintenance
copyFromObjectStorage	groupAssignmentAuditReport	runDataRule
copyOwnershipDataToNextYear	help	runDMReport
copySnapshotFromInstance	importAppSecurity	runIntegration
copyToObjectStorage	importConsolidationJournals	runRuleSet
createGroups	importData	runSupplementalDataReport
deleteFile	importDataManagement	runTaskManagerReport
deleteGroups	importJobConsole	sendMail
deployEJTemplates	importMapping	setApplicationAdminMode
deployFormTemplates	importMetadata	setDailyMaintenanceStartTime
deployTaskManagerTemplate	importOwnershipData	setDemoDates
downloadFile	importSnapshot	setEJJournalStatus
executeReportBurstingDefinition	importSupplementalCollectionData	setEncryptionKey
exportDataManagement	importSupplementalData	setEssbaseQryGovExecTime
exportEssbaseData	importValidIntersections	setIdleSessionTimeout
encrypt	invalidLoginReport	setIPAllowlist
exportAppAudit	listBackups	setVirusScanOnFileUploads
exportAppSecurity	listFiles	setManualDataAccess
exportConsolidationJournals	login	setSubstVars
exportData	logout	simulateConcurrentUsage
exportEJournals	maskData	skipUpdate
exportJobConsole	provisionReport	snapshotCompareReport
exportLibraryDocument	recomputeOwnershipData	unassignRole
exportMapping	recreate	updateUsers
exportMetadata	refreshCube	upgrade
exportOwnershipData	removeUserFromGroups	uploadFile
exportSnapshot	removeUsers	userAuditReport
exportTaskManagerAccessControl	removeUsersFromGroup	userGroupReport
exportValidIntersections	removeUsersFromTeam	validateConsolidationMetadata
exportSnapshot		

Narrative Reporting-Befehle

EPM Automate-Befehle für Narrative Reporting

addUsers	getIdleSessionTimeout	restoreBackup
addUsersToGroup	getIPAllowlist	roleAssignmentAuditReport
addUserToGroups	getVirusScanOnFileUploads	roleAssignmentReport
assignRole	groupAssignmentAuditReport	runDailyMaintenance
cloneEnvironment	help	sendMail
copyFileFromInstance	importLibraryArtifact	setDailyMaintenanceStartTime
copyFromObjectStorage	invalidLoginReport	setEncryptionKey
copyToObjectStorage	listBackups	setIdleSessionTimeout
createGroups	listFiles	setIPAllowlist
createNRSnapshot	login	setManualDataAccess
deleteFile	logout	setVirusScanOnFileUploads
deleteGroups	provisionReport	skipUpdate
downloadFile	recreate	unassignRole
encrypt	removeUserFromGroups	updateUsers
executeBurstDefinition	removeUsers	upgrade
exportLibraryArtifact	removeUsersFromGroup	uploadFile
feedback	replay	userAuditReport
getDailyMaintenanceStartTime	resetService	userGroupReport

Oracle Enterprise Data Management Cloud-Befehle

EPM Automate-Befehle für Oracle Enterprise Data Management Cloud

addUsers	getDailyMaintenanceStartTime	replay
addUsersToGroup	getIdleSessionTimeout	resetService
addUserToGroups	getIPAllowlist	restoreBackup
assignRole	getVirusScanOnFileUploads	roleAssignmentAuditReport
cloneEnvironment	groupAssignmentAuditReport	roleAssignmentReport
copyFileFromInstance	help	runDailyMaintenance
copyFromObjectStorage	importDimension	sendMail
copySnapshotFromInstance	importSnapshot	setDailyMaintenanceStartTime
copyToObjectStorage	invalidLoginReport	setEncryptionKey
createGroups	listBackups	setIdleSessionTimeout
deleteFile	listFiles	setIPAllowlist
deleteGroups	loadViewpoint	setManualDataAccess
downloadFile	login	setVirusScanOnFileUploads
encrypt	logout	skipUpdate
exportDimension	provisionReport	unassignRole
exportDimensionMapping	recreate	updateUsers
exportSnapshot	removeUserFromGroups	upgrade
extractDimension	removeUsers	uploadFile
feedback	removeUsersFromGroup	userAuditReport
	renameSnapshot	userGroupReport

Befehle für Planning, Planning Modules, FreeForm, Strategische Personalplanung und Sales Planning

EPM Automate-Befehle für Planning, Planning Modules, FreeForm, Strategische Personalplanung und Sales Planning

addUsers	getApplicationAdminMode	replay
addUsersToGroup	getDailyMaintenanceStartTime	resetService
addUserToGroups	getEssbaseQryGovExecTime	restoreBackup
applicationAdminMode	getIdleSessionTimeout	restructureCube
assignRole	getIPAllowlist	roleAssignmentAuditReport
autoPredict * siehe Fußnote	getSubstVar	roleAssignmentReport
clearCube	getVirusScanOnFileUploads	runBatch
cloneEnvironment	groupAssignmentAuditReport	runBusinessRule
copyFileFromInstance	help	runDailyMaintenance
copyFromObjectStorage	importAppAudit	runDataRule
copySnapshotFromInstance	importAppSecurity	runDMReport
copyToObjectStorage	importCellLevelSecurity	runIntegration
createGroups	importData	runPlanTypeMap
deleteFile	importDataManagement	runRuleSet
deleteGroups	importJobConsole	sendMail
dismissIPMInsights **	importMapping	setApplicationAdminMode
downloadFile	importMetadata	setDailyMaintenanceStartTime
enableQueryTracking	importSnapshot	setEncryptionKey
encrypt	importValidIntersections	setEssbaseQryGovExecTime
executeAggregationProcess	invalidLoginReport	setIdleSessionTimeout
executeReportBurstingDefinition	listBackups	setIPAllowlist
exportAppAudit	listFiles	setManualDataAccess
exportAppSecurity	login	setSubstVars
exportCellLevelSecurity	logout	setVirusScanOnFileUploads
exportData	maskData	simulateConcurrentUsage
exportDataManagement	mergeDataSlices	skipUpdate
exportEssbaseData	provisionReport	snapshotCompareReport
exportJobConsole	recreate	sortMember
exportLibraryDocument	refreshCube	unassignRole
exportMapping	removeUserFromGroups	updateUsers
exportMetadata	removeUsers	upgrade
exportSnapshot	removeUsersFromGroup	uploadFile
exportValidIntersections	renameSnapshot	userAuditReport
feedback		userGroupReport

* Dieser Befehl wird nicht für FreeForm, Strategische Personalplanung und Sales Planning unterstützt.

** Dieser Befehl wird nicht für FreeForm unterstützt.

Profitability and Cost Management-Befehle

EPM Automate-Befehle für Profitability and Cost Management

addUsers	getEssbaseQryGovExecTime	renameSnapshot
addUsersToGroup	getIdleSessionTimeout	replay
addUserToGroups	getIPAllowlist	resetService
applyDataGrants	getVirusScanOnFileUploads	restoreBackup
assignRole	groupAssignmentAuditReport	roleAssignmentAuditReport
clearPOV	help	roleAssignmentReport
cloneEnvironment	importDataManagement	runBatch
copyFileFromInstance	importMapping	runCalc
copyFromObjectStorage	importSnapshot	runDailyMaintenance
copyPOV	importTemplate	runDataRule
copySnapshotFromInstance	invalidLoginReport	runDMReport
copyToObjectStorage	listBackups	runIntegration
createGroups	listFiles	sendMail
deleteFile	loadData	setDailyMaintenanceStartTime
deleteGroups	loadDimData	setEncryptionKey
deletePOV	login	setEssbaseQryGovExecTime
deployCube	logout	setIdleSessionTimeout
downloadFile	mergeSlices	setIPAllowlist
enableApp	optimizeASOCube	setManualDataAccess
encrypt	programDocumentationReport	skipUpdate
exportDataManagement	provisionReport	setVirusScanOnFileUploads
exportMapping	recreate	unassignRole
exportQueryResults	removeUserFromGroups	updateUsers
exportSnapshot	removeUsers	upgrade
exportTemplate	removeUsersFromGroup	uploadFile
feedback		userAuditReport
getDailyMaintenanceStartTime		userGroupReport

Enterprise Profitability and Cost Management-Befehle

EPM Automate-Befehle für Enterprise Profitability and Cost Management

addUsers	exportValidIntersections	removeUsers
addUsersToGroup	feedback	removeUsersFromGroup
addUserToGroups	getApplicationAdminMode	renameSnapshot
applicationAdminMode	getDailyMaintenanceStartTime	replay
assignRole	getEssbaseQryGovExecTime	resetService
calculateModel	getIdleSessionTimeout	restoreBackup
clearCube	getIPAllowlist	roleAssignmentAuditReport
copyDataByPointOfView	getSubstVar	roleAssignmentReport
cloneEnvironment	getVirusScanOnFileUploads	runBatch
copyFileFromInstance	groupAssignmentAuditReport	runDailyMaintenance
copyFromObjectStorage	help	runDataRule
clearDataByPointOfView	importAppAudit	runDMReport
copySnapshotFromInstance	importAppSecurity	runIntegration
copyToObjectStorage	importCellLevelSecurity	sendMail
createGroups	importData	setApplicationAdminMode
deleteFile	importDataManagement	setDailyMaintenanceStartTime
deleteGroups	importJobConsole	setEncryptionKey
downloadFile	importMapping	setEssbaseQryGovExecTime
deletePointOfView	importMetadata	getIdleSessionTimeout
enableQueryTracking	importSnapshot	setIPAllowlist
encrypt	importValidIntersections	setManualDataAccess
executeAggregationProcess	invalidLoginReport	setSubstVars
executeReportBurstingDefinition	listBackups	setVirusScanOnFileUploads
exportAppAudit	listFiles	skipUpdate
exportAppSecurity	login	snapshotCompareReport
exportCellLevelSecurity	logout	sortMember
exportData	maskData	unassignRole
exportDataManagement	mergeDataSlices	updateUsers
exportEssbaseData	provisionReport	upgrade
exportJobConsole	recreate	uploadFile
exportLibraryDocument	refreshCube	userAuditReport
exportMapping	removeUserFromGroups	userGroupReport
exportMetadata		validateModel
exportMetadata		
exportSnapshot		

Tax Reporting-Befehle

EPM Automate-Befehle für Tax Reporting

addUsers	getDailyMaintenanceStartTime	replay
addUsersToGroup	getEssbaseQryGovExecTime	resetService
addUsersToTeam	getIdleSessionTimeout	restoreBackup
addUserToGroups	getIPAllowlist	restructureCube
applicationAdminMode	getSubstVar	roleAssignmentAuditReport
assignRole	getVirusScanOnFileUploads	roleAssignmentReport
clearDataByProfile	groupAssignmentAuditReport	runBatch
copyDataByProfile	help	runBusinessRule
copyFileFromInstance	importAppSecurity	runDailyMaintenance
copyFromObjectStorage	importCellLevelSecurity	runDataRule
copyOwnershipDataToNextYear	importData	runDMReport
copySnapshotFromInstance	importDataManagement	runIntegration
copyToObjectStorage	importJobConsole	runRuleSet
createGroups	importMapping	runSupplementalDataReport
deleteFile	importMetadata	runTaskManagerReport
deleteGroups	importOwnershipData	sendMail
deployFormTemplates	importSnapshot	setApplicationAdminMode
downloadFile	importSupplementalCollectionData	setDailyMaintenanceStartTime
encrypt	importSupplementalData	setDemoDates
executeReportBurstingDefinition	importValidIntersections	setEncryptionKey
exportAppAudit	invalidLoginReport	setEssbaseQryGovExecTime
exportCellLevelSecurity	listBackups	setIdleSessionTimeout
exportData	listFiles	setIPAllowlist
exportDataManagement	login	setManualDataAccess
exportEssbaseData	logout	setSubstVars
exportJobConsole	maskData	setVirusScanOnFileUploads
exportLibraryDocument	provisionReport	simulateConcurrentUsage
exportMapping	recomputeOwnershipData	skipUpdate
exportMetadata	recreate	snapshotCompareReport
exportOwnershipData	refreshCube	unassignRole
exportSnapshot	removeUserFromGroups	upgrade
exportTaskManagerAccessControl	removeUsers	updateUsers
exportValidIntersections	removeUsersFromGroup	uploadFile
feedback	removeUsersFromTeam	userAuditReport
getApplicationAdminMode	renameSnapshot	userGroupReport