

Oracle® Fusion Cloud EPM

Utilizzo di Automazione EPM per Oracle Enterprise Performance Management Cloud



F28907-21



Oracle Fusion Cloud EPM Utilizzo di Automazione EPM per Oracle Enterprise Performance Management Cloud,

F28907-21

Copyright © 2016, 2024, , Oracle e/o relative consociate.

Autore principale: 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>.

Sommario

Accesso facilitato alla documentazione

Feedback relativi alla documentazione

1 Informazioni su EPM Automate

Installazione di EPM Automate	1-1
Requisiti di capacità e porte	1-2
Piattaforme supportate	1-2
Java Runtime Environment e EPM Automate	1-4
Utilizzo di OpenJDK	1-4
Istruzioni per Windows	1-5
Istruzioni per Linux/UNIX/macOS X	1-5
Esecuzione di comandi EPM Automate sul lato server	1-6
Informazioni sul livello di cifratura di EPM Automate	1-6
Utilizzo del protocollo di autorizzazione OAuth 2.0 con ambienti OCI (Gen2)	1-6

2 Guida comandi

Informazioni sull'esecuzione dei comandi di EPM Automate	2-1
Prerequisiti	2-1
Posizioni di file predefinite	2-3
Abilitazione del protocollo Transport Layer Security 1.2	2-4
Utilizzo dei comandi di EPM Automate	2-4
Definizione di valori multipli per un parametro	2-5
Comportamento durante la manutenzione giornaliera	2-5
Esecuzione di EPM Automate	2-6
Windows	2-6
Linux	2-7
Esecuzione di più istanze di EPM Automate	2-8
Panoramica dei comandi	2-10
Comandi di EPM Automate	2-16

addUsers	2-16
addUsersToGroup	2-17
addUsersToTeam	2-18
addUserToGroups	2-19
applicationAdminMode	2-20
applyDataGrants	2-21
archiveTmTransactions	2-21
assignRole	2-23
autoPredict	2-25
calculateModel	2-26
clearCube	2-28
clearDataByPointOfView	2-29
clearDataByProfile	2-30
clearPOV	2-31
cloneEnvironment	2-32
copyDataByPointOfView	2-37
copyDataByProfile	2-38
copyFileFromInstance	2-38
copyFromObjectStorage	2-39
copyOwnershipDataToNextYear	2-41
copyPOV	2-42
copySnapshotFromInstance	2-43
copyToObjectStorage	2-44
createGroups	2-46
createNRSnapshot	2-46
createReconciliations	2-47
deleteFile	2-47
deleteGroups	2-49
deletePointOfView	2-49
deletePOV	2-50
deployCube	2-51
deployEJTemplates	2-51
deployFormTemplates	2-52
deployTaskManagerTemplate	2-53
dismissIPMInsights	2-54
downloadFile	2-54
enableApp	2-55
enableQueryTracking	2-55
encrypt	2-56
executeAggregationProcess	2-57
executeBurstDefinition	2-58

executeReportBurstingDefinition	2-59
exportAccessControl	2-59
exportAppAudit	2-60
exportAppSecurity	2-61
exportARApplicationProperties	2-61
exportBackgroundImage	2-63
exportCellLevelSecurity	2-63
exportConsolidationJournals	2-64
exportData	2-64
exportDataManagement	2-65
exportDimension	2-65
exportDimensionMapping	2-66
exportEJournals	2-67
exportEssbaseData	2-68
exportJobConsole	2-69
exportLibraryArtifact	2-71
exportLibraryDocument	2-73
exportLogoImage	2-74
exportMapping	2-74
exportMetadata	2-75
exportOwnershipData	2-75
exportQueryResults	2-76
exportSnapshot	2-78
exportTemplate	2-79
exportTaskManagerAccessControl	2-80
exportValidIntersections	2-81
extractDimension	2-81
feedback	2-82
getApplicationAdminMode	2-83
getDailyMaintenanceStartTime	2-84
getEssbaseQryGovExecTime	2-85
getIdleSessionTimeout	2-85
getIPAllowlist	2-86
getSubstVar	2-87
getVirusScanOnFileUploads	2-88
groupAssignmentAuditReport	2-88
help	2-89
importAppAudit	2-89
importAppSecurity	2-90
importARApplicationProperties	2-91
importBackgroundImage	2-92

importBalances	2-92
importCellLevelSecurity	2-93
importConsolidationJournals	2-94
importData	2-94
importDataManagement	2-95
importDimension	2-96
importJobConsole	2-97
importLibraryArtifact	2-98
importLogolImage	2-99
importMapping	2-100
importMetadata	2-101
importOwnershipData	2-102
importPreMappedBalances	2-103
importPreMappedTransactions	2-104
importProfiles	2-104
importRates	2-105
importRCAttributeValues	2-105
importReconciliationAttributes	2-107
importSnapshot	2-108
importSupplementalCollectionData	2-111
importSupplementalData	2-112
importTemplate	2-113
importTMAttributeValues	2-113
importTmPremappedTransactions	2-115
importValidIntersections	2-116
invalidLoginReport	2-117
listBackups	2-118
listFiles	2-119
loadData	2-120
loadDimData	2-121
loadViewpoint	2-122
login	2-123
logout	2-126
maskData	2-126
mergeDataSlices	2-127
mergeSlices	2-127
optimizeASOCube	2-128
programDocumentationReport	2-129
provisionReport	2-130
purgeArchivedTmTransactions	2-132
purgeTmTransactions	2-132

recomputeOwnershipData	2-134
recreate	2-134
refreshCube	2-138
removeUserFromGroups	2-139
removeUsers	2-140
removeUsersFromGroup	2-141
removeUsersFromTeam	2-142
renameSnapshot	2-143
replay	2-143
resetService	2-145
restoreBackup	2-146
restructureCube	2-147
roleAssignmentAuditReport	2-147
roleAssignmentReport	2-149
runAutomatch	2-150
runBatch	2-151
runBusinessRule	2-151
runCalc	2-152
runComplianceReport	2-154
runDailyMaintenance	2-155
runDataRule	2-156
runDMReport	2-158
runIntegration	2-159
runMatchingReport	2-163
runPlanTypeMap	2-164
runRuleSet	2-165
runSupplementalDataReport	2-165
runTaskManagerReport	2-166
sendMail	2-167
setApplicationAdminMode	2-169
setDailyMaintenanceStartTime	2-169
setDemoDates	2-170
setEJJournalStatus	2-171
setEncryptionKey	2-172
setEssbaseQryGovExecTime	2-173
setIdleSessionTimeout	2-173
setIPAllowlist	2-174
setManualDataAccess	2-175
setPeriodStatus	2-176
setSubstVars	2-177
setVirusScanOnFileUploads	2-177

simulateConcurrentUsage	2-178
skipUpdate	2-181
snapshotCompareReport	2-184
sortMember	2-185
unassignRole	2-186
updateUsers	2-187
upgrade	2-188
uploadFile	2-189
userAuditReport	2-190
userGroupReport	2-192
validateConsolidationMetadata	2-192
validateModel	2-193
Codici di uscita	2-194

3 Scenari di esempio per l'esecuzione dei comandi

Informazioni sulla copia di script campione	3-1
Scenari di esempio per tutti i servizi	3-1
Creazione del backup dello snapshot dell'applicazione in un computer	3-3
Informazioni agli utenti sul completamento della manutenzione giornaliera	3-5
Copia di uno snapshot a oppure da Oracle Object Storage	3-13
Creazione di utenti e assegnazione a ruoli predefiniti	3-15
Conteggio del numero di utenti con licenza (utenti assegnati a ruoli)	3-18
Creazione di report di audit di utenti assegnati a ruoli	3-21
Creazione del report di audit sull'assegnazione e revoca dei ruoli	3-25
Mascheramento dei log di accesso e del report attività per conformità con le leggi sulla privacy	3-28
Automazione del download dei report attività in un computer locale	3-34
Scaricamento dei log di accesso da un ambiente	3-37
Automazione della clonazione degli ambienti	3-41
Rimozione dei file superflui da un ambiente	3-45
Ricerca e scaricamento di file da un ambiente	3-47
Come ricreare un vecchio ambiente EPM Cloud per gli audit	3-48
Automazione dell'audit e della conformità degli accessi al database	3-60
Replica degli utenti e delle assegnazioni dei ruoli predefiniti	3-71
Replica degli utenti da un dominio di Identity a un altro	3-72
Replica delle assegnazioni dei ruoli predefiniti da un ambiente a un altro	3-78
Creazione di una cadenza di aggiornamento trimestrale di EPM Cloud	3-85
Script per Windows e istruzioni	3-87
Script per UNIX/Linux e istruzioni	3-90
Script Groovy	3-93

Creazione di una cadenza di aggiornamento trimestrale di EPM Cloud con cicli di test di sei settimane	3-97
Scenari campione per Planning, Consolidation, Tax Reporting ed Enterprise Profitability and Cost Management	3-112
Automazione dell'esportazione di un numero elevato di celle da un cubo di memorizzazione di aggregazione	3-113
Importazione di metadati in un'applicazione	3-122
Importazione di dati, esecuzione di uno script di calcolo e copia dei dati da un database di memorizzazione a blocchi in un database di memorizzazione di aggregazione	3-124
Esportazione e download di metadati e dati	3-127
Esportazione e download dei dati dell'applicazione	3-130
Automatizzazione dell'archiviazione dei record di audit dell'applicazione	3-132
Script per Windows	3-133
Script per Linux	3-134
Caricamento di un file di dati in un ambiente ed esecuzione di una regola di caricamento dati	3-135
Automazione dell'integrazione dei dati giornalieri	3-138
Scenari di esempio per Account Reconciliation	3-140
Caricamento di saldi preformattati in un periodo	3-141
Caricamento e importazione di uno snapshot di backup	3-143
Archiviazione di transazioni con corrispondenza meno recenti e rimozione di transazioni archiviate	3-145
Scenari di esempio per Profitability and Cost Management	3-152
Importazione di metadati in un'applicazione	3-152
Importazione dei dati ed esecuzione delle regole del programma	3-154
Scenari di esempio per Oracle Enterprise Data Management Cloud	3-156
Sincronizzazione di dimensioni e mapping di Oracle Enterprise Data Management Cloud mediante applicazioni EPM Cloud	3-157
Sincronizzazione delle dimensioni di EPM Cloud mediante applicazioni Oracle Enterprise Data Management Cloud	3-159
Automazione dell'esecuzione degli script	3-160
Monitoraggio delle attività di EPM Automate	3-161

4 Esecuzione di comandi senza installare EPM Automate

Ambienti che supportano l'esecuzione di comandi sul lato server	4-1
Fonti di informazioni	4-2
Comandi supportati	4-2
Metodi da utilizzare per eseguire EPM Automate utilizzando Groovy su lato server	4-2
Clonazione di un ambiente tramite uno script Groovy su lato server	4-3
Invio del report attività via e-mail tramite uno script Groovy su lato server	4-5

5 Replica di un ambiente di EPM Cloud

Configurazione della replica giornaliera	5-1
Configurazione di repliche su richiesta	5-2
Configurazione dell'ambiente secondario	5-3

A Preparazione all'esecuzione del comando simulateConcurrentUsage

Creazione del file requirement.csv	A-1
Creazione dei file di input	A-3
File di input per l'apertura dei form	A-3
File di input per il salvataggio dei form	A-4
Esecuzione del file di input di una regola business	A-5
Esecuzione del file di input della regola dati	A-5
File di input della griglia ad hoc	A-5
Esecuzione del file di input del report	A-6
Esecuzione del file di input del registro	A-7
Creazione del file UserVarMemberMapping.csv	A-7
Creazione e caricamento del file ZIP di input nell'ambiente	A-8
Esempio di report di simulazione dell'utilizzo concorrente	A-8

B Preparazione per l'esecuzione del comando di ripetizione

Informazioni sul comando di ripetizione	B-1
Prerequisiti	B-1
Creazione di file HAR	B-2
Creazione di file di ripetizione	B-5
Generazione di file di tracce	B-6
Sessione di ripetizione di esempio	B-6

C Gestione dei caratteri speciali

D Comandi specifici di ciascun servizio EPM Cloud

Comandi di Account Reconciliation	D-2
Comandi di Financial Consolidation and Close	D-3
Comandi di Narrative Reporting	D-4
Comandi di Oracle Enterprise Data Management Cloud	D-5
Comandi di Planning, Planning Modules, FreeForm, Strategic Workforce Planning, e Sales Planning	D-6
Comandi di Profitability and Cost Management	D-7

Comandi di Enterprise Profitability and Cost Management

D-8

Comandi di Tax Reporting

D-9

Accesso facilitato alla documentazione

Per informazioni sull'impegno di Oracle riguardo l'accesso facilitato, visitare il sito Web Oracle Accessibility Program all'indirizzo <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Accesso al Supporto Oracle

I clienti Oracle che hanno acquistato il servizio di supporto tecnico hanno accesso al supporto elettronico attraverso My Oracle Support. Per informazioni, visitare <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> oppure <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> per clienti non udenti.

Feedback relativi alla documentazione

Per fornire un feedback su questa documentazione, fare clic sul pulsante Feedback in fondo alla pagina in un qualsiasi argomento di Oracle Help Center. È inoltre possibile inviare un messaggio e-mail a epmdoc_ww@oracle.com.

1

Informazioni su EPM Automate

EPM Automate consente agli utenti di eseguire da remoto i task all'interno degli ambienti Oracle Enterprise Performance Management Cloud.

Con EPM Cloud, gli amministratori di servizi possono automatizzare molti task ripetibili, compresi quelli riportati di seguito.

- Importare ed esportare metadati, dati, artifact, snapshot applicazione, template e mapping di Gestione dati
- Caricare i file negli ambienti, elencare i file ed eliminare i file dal servizio
- Eseguire il download di snapshot, report e file di dati e metadati dal servizio
- Eseguire regole business sui dati e aggiornare l'applicazione
- Copia di dati da un database ad un altro, solitamente da un database di memorizzazione a blocchi a un database di memorizzazione di aggregazione o da un database di memorizzazione a blocchi a un altro un database di memorizzazione a blocchi.
- Eseguire una regola batch di Gestione dati
- Generare i report di Gestione dati, report di assegnazione ruoli e report audit utente
- Importare i dati dei saldi premappati, i tassi valuta, le transazioni premappate, i dati dei saldi e i profili in Account Reconciliation
- Copiare profili in un periodo per avviare il processo di riconciliazione
- Distribuire il cubo di calcolo delle applicazioni Profitability and Cost Management
- Cancellare, copiare ed eliminare punti di vista nelle applicazioni Enterprise Profitability and Cost Management e Profitability and Cost Management
- Ripetere il caricamento di Oracle Smart View for Office o dell'API REST in un ambiente per consentire l'esecuzione del test delle performance in caso di carichi di particolare entità
- Importare dati supplementari da un file in Financial Consolidation and Close

Mediante un programma di pianificazione è possibile creare script che consentono di completare una vasta gamma di task automatizzandone l'esecuzione. Ad esempio, è possibile creare uno script per scaricare il backup della manutenzione giornaliera dagli ambienti per creare backup locali degli artifact e dei dati.



[Esercitazione su come eseguire i comandi di Planning mediante EPM Automate](#)

Installazione di EPM Automate

Installare EPM Automate per eseguire i comandi. Alcuni comandi possono anche essere eseguiti direttamente in Oracle Enterprise Performance Management Cloud utilizzando script Groovy, senza installare EPM Automate.

Il programma di installazione di EPM Automate per Windows e Linux/UNIX e per macOS X è disponibile nell'ambiente EPM Cloud.

Poiché Windows 10 e versioni successive consentono solo agli amministratori di Windows di installare EPM Automate, l'installazione e l'aggiornamento possono essere eseguiti solo da amministratori di Windows. EPM Automate può essere aggiornato dall'utente che l'ha installato o da un altro amministratore di Windows.

In questa sezione sono contenuti gli argomenti elencati di seguito.

- [Requisiti di capacità e porte](#)
- [Piattaforme supportate](#)
- [Java Runtime Environment e EPM Automate](#)
- [Utilizzo di OpenJDK](#)
- [Istruzioni per Windows](#)
- [Istruzioni per Linux/UNIX/macOS X](#)
- [Esecuzione dei comandi EPM Automate sul lato server](#)

Requisiti di capacità e porte

Poiché EPM Automate è un client ridotto, non richiede un footprint client esteso. Tutte le operazioni di elaborazione vengono eseguite in Oracle Enterprise Performance Management Cloud.

È possibile installare EPM Automate su computer client standard, su virtual machine e su computer Oracle Integration Cloud in grado di accedere a host esterni su una connessione HTTP sicura.

EPM Automate si connette a EPM Cloud utilizzando la porta TLS standard (porta 443). Non è necessario aprire ulteriori porte in uscita per EPM Automate.

Piattaforme supportate

È possibile installare EPM Automate su virtual machine e su computer Oracle Integration Cloud (OIC) in grado di accedere a host esterni su una connessione HTTP sicura.

 **Note:**

- È possibile utilizzare EPM Automate solo su sistemi operativi a 64 bit attualmente supportati dal fornitore del sistema operativo.
- EPM Automate non funziona con il proxy SOCKS; funziona solo con il proxy HTTP/HTTPS.
- EPM Automate supporta i meccanismi di autenticazione di base, del digest, Kerberos, con negoziazione e NTLM per la connessione al server proxy.
- EPM Automate può connettersi a Oracle Enterprise Performance Management Cloud tramite gateway API, ad esempio Google APIGEE, IBM Data Power e altri server proxy inversi.

Per il corretto funzionamento, configurare il gateway o il proxy inverso impostando il target come URL dell'ambiente EPM Cloud senza specificare alcun contesto come ad esempio /epmcloud. Esempio: `https://epm-idDomain.epm.dataCenterRegion.oraclecloud.com`. Utilizzare quindi l'URL del proxy inverso anziché l'URL di EPM Cloud nel comando [login](#). Per informazioni sulla configurazione, fare riferimento alla documentazione del gateway o del server proxy.

Mentre si configurano le impostazioni proxy, assicurarsi di passare il codice di risposta da EPM Cloud a EPM Automate senza modificarlo in alcun modo per consentire così a EPM Automate di elaborare correttamente codici di risposta quali ad esempio 200,206, 400, 404, 500, 501 e così via. Ad esempio, per IBM Datapower, impostare `proxy HTTP Response` su ON. Inoltre, il gateway API deve consentire i metodi HTTP (GET, POST, PUT, PATCH e DELETE).

Nei computer con Linux e UNIX, EPM Automate cerca le variabili di ambiente elencate di seguito per determinare le impostazioni proxy HTTP o HTTPS.

- `proxyHost`
- `proxyPort`

Esempi di impostazioni per proxy http:

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

Esempi di impostazioni per proxy https:

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

 **Note:**

EPM Automate può utilizzare il protocollo di autenticazione OAuth 2.0 per accedere agli ambienti EPM Cloud OCI (Gen 2) (se configurati per OAuth) per eseguire i comandi, soprattutto per automatizzarne l'esecuzione.

Negli ambienti Classic e negli ambienti OCI (Gen 2) che utilizzano l'autenticazione di base, EPM Automate non utilizza le credenziali SSO (provider di identità). Poiché gli utenti non possono eseguire l'accesso utilizzando le credenziali aziendali, gli account utente per l'accesso a EPM Automate devono essere gestiti nel dominio di Identity. Se la sottoscrizione è configurata per l'autenticazione SSO, è inoltre necessario abilitare l'accesso degli utenti EPM Automate con le credenziali personali del dominio di Identity. Vedere Abilitazione dell'accesso con credenziali del dominio di Identity in *Amministrazione di Oracle Cloud Identity Management*.

Istruzioni per il download: Scaricamento e installazione dei client nella *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori*.

Java Runtime Environment e EPM Automate

L'installazione di EPM Automate su Windows installa l'ambiente JRE (Java Runtime Environment) richiesto. Nei programmi di installazione per Linux, Unix e macOS X non è incluso alcun ambiente JRE. Per poter utilizzare EPM Automate è necessario disporre dell'accesso a un'installazione JRE (dalla versione 8 alla 11).

È possibile utilizzare Oracle Java Standard Edition (SE) con EPM Automate senza la necessità di acquistare separatamente una sottoscrizione Java SE. Per informazioni dettagliate sulle licenze di Oracle JDK con EPM Automate, fare riferimento al [Documento Supporto Oracle 1557737.1: "Support Entitlement for Java SE When Used As Part of Another Oracle Product"](#).

Utilizzo di OpenJDK

È possibile utilizzare OpenJDK versione 14 o successive anziché JRE su piattaforme Linux, Unix e macOS X.

OpenJDK, il JDK gratuito di Oracle, pronto per la produzione e con licenza GPL può essere scaricato all'indirizzo <https://openjdk.java.net>. In questo sito Web sono inoltre disponibili le istruzioni per installare OpenJDK.

Prima di avviare una sessione EPM Automate, impostare la variabile di ambiente `JAVA_HOME` in modo che punti all'installazione di OpenJDK:

Esempio per macOS X (presupponendo una shell Bash) per l'utilizzo di OpenJDK versione 14 installato nella directory home.

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

Esempio per Linux (con shell Bash) per utilizzare OpenJDK versione 14 installato nella directory home in uso.

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

Istruzioni per Windows

Per impostazione predefinita, EPM Automate viene installato in C:/Oracle/EPM Automate.

Per installare EPM Automate, procedere come segue.

1. Accedere a un ambiente dal computer Windows in cui si desidera installare EPM Automate.
2. Nella home page accedere a **Impostazione e azioni** facendo clic sul nome utente.
3. Fare clic su **Download**.
4. Nella pagina Download fare clic su **Download per Windows** nella sezione Automazione EPM.
5. Salvare il programma di installazione nel computer.
6. Fare clic con il pulsante destro del mouse sul programma di installazione (EPM Automate.exe) e selezionare **Esegui come amministratore**.
7. In **Controllo account utente**, fare clic su **Sì**.
8. Seguire le indicazioni a video per completare l'installazione.

Istruzioni per Linux/UNIX/macOS X

EPM Automate richiede l'accesso a una distribuzione di una versione supportata di JRE (dalla 8 alla 11). La variabile di ambiente `JAVA_HOME` deve essere impostata in modo da puntare all'installazione JRE in uso.

Per installare EPM Automate, procedere come segue.

1. Accedere a un ambiente.
2. Nella home page accedere a **Impostazione e azioni** facendo clic sul nome utente.
3. Fare clic su **Download**.
4. Nella pagina Scaricamenti, fare clic su **Scaricamento per Linux/macOS X** nella sezione EPM Automate.
5. Salvare il programma di installazione (`EPMAutomate.tar`) in una directory in cui si dispone di privilegi di scrittura, lettura ed esecuzione.
6. Estrarre il contenuto del programma di installazione, impostare le variabili di ambiente richieste ed eseguire `epmautomate.sh`:

Esempio per macOS X (presupponendo una shell Bash) per l'installazione e l'esecuzione dalla directory home.

```
cd ~/
tar xf path_to_downloaded_EPMAutomate.tar
export JAVA_HOME=$(/usr/libexec/java_home)
```

```
export PATH $HOME/epmautomate/bin:$PATH
epmautomate.sh
```

Esempio per Linux (presupponendo una shell Bash) da installare ed eseguire dalla directory home. Si presume l'utilizzo di JDK versione 1.8.0_191.

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

Esecuzione di comandi EPM Automate sul lato server

Alcuni comandi EPM Automate possono essere eseguiti direttamente in Oracle Enterprise Performance Management Cloud utilizzando Groovy. Non è necessario installare EPM Automate per eseguire i comandi con gli script Groovy.

Si noti che l'esecuzione di comandi sul lato server non equivale all'esecuzione di script Groovy in un computer client per l'esecuzione di comandi EPM Automate.

Per informazioni dettagliate, vedere [Running Commands without Installing EPM Automate \(in lingua inglese\)](#).

Informazioni sul livello di cifratura di EPM Automate

Oracle Enterprise Performance Management Cloud utilizza il protocollo Transport Layer Security (TLS) con l'algoritmo hash di crittografia SHA-2/SHA-256 per proteggere le comunicazioni con EPM Automate.

Utilizzo del protocollo di autorizzazione OAuth 2.0 con ambienti OCI (Gen2)

EPM Automate può utilizzare il protocollo di autenticazione OAuth 2.0 per accedere agli ambienti Oracle Enterprise Performance Management Cloud OCI (GEN 2) per eseguire i comandi, soprattutto per automatizzarne l'esecuzione.

Per abilitare l'accesso OAuth 2.0, un amministratore del dominio di Identity deve registrare l'applicazione come client pubblico in Servizi di Identity Oracle Cloud. Il protocollo OAuth viene utilizzato per l'applicazione, non per la sottoscrizione.

Per istruzioni dettagliate sull'impostazione di OAuth 2.0 per gli ambienti OCI (Gen 2), fare riferimento alla sezione Autenticazione con OAuth 2 - Solo per ambienti OCI (Gen 2) nel manuale *API REST per Oracle Enterprise Performance Management Cloud*.

Note:

L'autenticazione di base funziona anche quando è abilitato il protocollo OAuth per un ambiente. Assicurarsi di non sovrascrivere il password file cifrato esistente se si intende utilizzarlo in futuro.

Creazione di un password file cifrato contenente token di aggiornamento e ID client

Gli amministratori servizi che desiderano utilizzare OAuth 2.0 per l'accesso EPM Automate agli ambienti richiedono i dettagli elencati di seguito per creare il password file cifrato, che viene quindi utilizzato per l'accesso all'ambiente.

- **Token di aggiornamento**
Per istruzioni dettagliate su come ottenere il token di aggiornamento, fare riferimento ai passi "Task degli amministratori servizi di EPM Cloud per ottenere un token di aggiornamento" nella sezione Autenticazione con OAuth 2 - Solo per ambienti OCI (Gen 2) nella Guida *API REST per Oracle Enterprise Performance Management Cloud*.
- **ID client**
L'ID client viene generato quando l'amministratore del dominio di Identity configura l'applicazione per OAuth. È visibile nella scheda Configurazione dell'applicazione, in **Informazioni generali**.

Per creare il password file cifrato per l'autenticazione OAuth, procedere come segue.

1. Avviare una sessione di EPM Automate.

2. Eseguire un comando simile al seguente:

```
epmautomate encrypt REFRESH_TOKEN ENCRYPTION_KEY PASSWORD_FILE  
ClientID=CLIENT_ID, dove REFRESH_TOKEN è il token di aggiornamento decifrato della  
memorizzazione sicura, ENCRYPTION_KEY è qualsiasi chiave privata di cifratura della  
password e PASSWORD_FILE sono il nome e la posizione del file in cui è memorizzato il  
token di aggiornamento cifrato. Per il file delle password, usare l'estensione .epw.
```

Fare riferimento alla sezione [encrypt](#) per istruzioni dettagliate.

3. Utilizzare il password file appena generato per eseguire l'accesso utilizzando OAuth. Per le esecuzioni di script automatizzate, assicurarsi di aggiornare gli script in modo che puntino al password file appena generato.

2

Guida comandi

- [Informazioni sull'esecuzione dei comandi di EPM Automate](#)
- [Esecuzione di EPM Automate](#)
- [Panoramica dei comandi](#)
- [Comandi di EPM Automate](#)
- [Codici di uscita](#)

Alcuni comandi di EPM Automate si applicano a tutti i processi aziendali, mentre altri si applicano a un gruppo di processi aziendali. Se non specificato diversamente, un comando applicabile a un processo aziendale specifico (ad esempio Planning) non funziona con un altro processo aziendale (ad esempio Financial Consolidation and Close). Qualsiasi tentativo di eseguire un comando per un processo aziendale in cui non è supportato restituirà un errore. Per un elenco dei comandi applicabili a ciascun processo aziendale, fare riferimento alla sezione [Comandi specifici di ciascun servizio EPM Cloud](#).

Informazioni sull'esecuzione dei comandi di EPM Automate

Tutti i servizi di Oracle Enterprise Performance Management Cloud utilizzano i comandi di EPM Automate per l'amministrazione remota degli ambienti.

- [Prerequisiti](#)
- [Posizioni di file predefinite](#)
- [Abilitazione del protocollo Transport Layer Security 1.2](#)
- [Utilizzo dei comandi di EPM Automate](#)
- [Definizione di valori multipli per un parametro](#)
- [Comportamento durante la manutenzione giornaliera](#)

Prerequisiti

In questa sezione sono elencati i prerequisiti per l'utilizzo di EPM Automate, ad esempio l'uso delle credenziali di Oracle Enterprise Performance Management Cloud e le posizioni predefinite dei file nei vari ambienti.

Generale

Tutti gli utenti di EPM Cloud possono utilizzare le credenziali del proprio dominio di Identity per connettersi a un ambiente mediante EPM Automate. I comandi che un utente può eseguire dipendono dai ruoli predefiniti e dai ruoli dell'applicazione che gli sono stati assegnati.

- Inoltre, è necessario disporre del ruolo amministratore del dominio di Identity per eseguire i comandi che consentono di aggiungere o eliminare gli utenti nel dominio di Identity.

- Qualsiasi file necessario per eseguire un comando deve esistere all'interno dell'ambiente. Utilizzare il comando `uploadFile` per caricare i file.
Vedere [Posizioni di file predefinite](#) per informazioni sulla posizione di file predefinita utilizzata da ciascun servizio.
- Utilizzo delle estensioni file nei comandi:
 - Specificare il nome completo del file, inclusa l'estensione (ad esempio, `data.csv`), per l'esecuzione di comandi che eseguono operazioni sui file. `deletefile`, `listfiles` e `uploadfile` sono esempi di tali comandi.
 - Non utilizzare le estensioni file per l'esecuzione di comandi che eseguono operazioni di migrazione. Per tali operazioni è necessario specificare il nome di uno snapshot.
- I valori dei parametri contenenti uno spazio; ad esempio commenti, nomi di posizione e percorsi di cartelle, devono essere racchiusi tra punti esclamativi.

Planning

- Job
Molti comandi descritti nella sezione seguente richiedono l'utilizzo dei job. I job sono azioni, quali l'importazione o l'esportazione di dati, che possono essere avviati immediatamente o programmati per un momento futuro, come nel caso dell'importazione o esportazione dei dati e dell'aggiornamento del database.
Utilizzare la console dei job per creare job appropriati al fine di eseguire le operazioni riportate di seguito. Per istruzioni dettagliate sulla creazione dei job in Planning, fare riferimento alla sezione "Gestione dei job" in *Amministrazione di Planning*.
 - Importazione di dati nell'applicazione
 - Esportazione di dati da un'applicazione
 - Importazione di metadati in un'applicazione
 - Esportazione di metadati da un'applicazione
 - Copia di dati da un database di memorizzazione a blocchi a un database di memorizzazione di aggregazione o da un database di memorizzazione a blocchi a un altro database di memorizzazione a blocchi.
- Regole business
Le regole business che si desidera eseguire devono essere presenti nell'applicazione.
Le regole business vengono create con Calculation Manager e quindi distribuite nell'applicazione. Vedere *Progettazione con Calculation Manager per Oracle Enterprise Performance Management Cloud*.

Gestione dati

- Regole dati
Le regole di caricamento dati definiscono come Gestione dati carica i dati da un file. È necessario aver predefinito regole di caricamento dati per caricare i dati mediante EPM Automate.
- Batch
È possibile caricare i dati utilizzando i batch definiti in Gestione dati. Utilizzando un batch, gli utenti possono combinare più regole di caricamento in un batch ed eseguirle in modalità seriale o parallela.

Posizioni di file predefinite

Posizione di caricamento predefinita

Per impostazione predefinita, tutti i file caricati in Oracle Enterprise Performance Management Cloud vengono memorizzati in una posizione predefinita accessibile a Migrazione.

È necessario caricare nella posizione predefinita i file che devono essere elaborati da Migrazione, ad esempio gli snapshot che si desidera importare nel servizio.

Cartelle inbox e outbox

Le posizioni delle cartelle inbox e outbox possono differire a seconda dei processi aziendali EPM Cloud. La cartella inbox viene utilizzata per caricare i file che si desidera importare o comunque elaborare utilizzando un processo aziendale diverso da Profitability and Cost Management. Data Management può elaborare i file nella cartella inbox o in una directory in essa contenuta.

EPM Cloud in genere memorizza nella cartella outbox i file che vengono generati mediante i processi aziendali, ad esempio file di esportazione di dati o metadati.

- La cartella inbox in cui EPM Automate carica i file e la cartella outbox in cui vengono memorizzati i file per lo scaricamento sono accessibili a queste applicazioni. È necessario caricare i file in questa posizione se si prevede di elaborare tali file utilizzando un processo nativo di queste applicazioni. È anche possibile caricare i file nella cartella outbox.
 - Planning
 - Moduli di Planning
 - Riconciliazione conti
 - Consolidamento finanziario e chiusura
 - Tax Reporting
 - Narrative Reporting
 - Enterprise Profitability and Cost Management

È possibile utilizzare la funzionalità Explorer file in entrata/in uscita per sfogliare i file memorizzati nella posizione predefinita. Gli snapshot dell'applicazione creati mediante EPM Automate non compaiono in Explorer file in entrata/in uscita. È possibile visualizzarli nella scheda Snapshot di Migrazione.

- I file che devono essere elaborati utilizzando un processo di Profitability and Cost Management devono essere caricati nella cartella `profitinbox`. È anche possibile caricare file nella cartella `profitoutbox`. I file esportati dai processi di Profitability and Cost Management vengono memorizzati nella cartella `profitinbox`. Utilizzare la funzionalità Explorer file per sfogliare questi file.
- I file elaborati utilizzando Gestione dati devono essere disponibili nella cartella dei file in entrata o in una cartella al suo interno. Per impostazione predefinita, i file esportati utilizzando Gestione dati vengono memorizzati nella cartella Posta in uscita, mentre gli output dei report di Gestione dati vengono memorizzati nella cartella `outbox/report di Gestione dati`. Utilizzare la funzionalità Browser file di Gestione dati per sfogliare questi file.

- Oracle Enterprise Data Management Cloud utilizza la posizione predefinita per importare ed esportare i dati caricati, copiati o scaricati. Per visualizzare i file nella posizione predefinita, utilizzare il comando `ListFiles`.

File di log

Ogni esecuzione di comandi di EPM Automate genera un file di debug, che viene eliminato automaticamente se il comando ha esito positivo. Se durante l'esecuzione di un comando si verifica un errore, il file di debug per il comando non riuscito viene conservato nella directory di esecuzione di EPM Automate. Per impostazione predefinita, si tratta della directory `Oracle/epm_automate/bin` (Windows) o `home/user/epmautomate/bin` (Linux/UNIX).

I file di debug di EPM Automate utilizzano la seguente convenzione di denominazione:

`nomecomando_data_indicatore_orario.log`. Ad esempio, in caso di esecuzione di un comando `listfiles` non riuscito alle 09.28.02 del 23 novembre 2020, verrà generato il seguente nome di file di debug: `listfiles_23_11_2020_09_28_02.log`.

Non è possibile disattivare la creazione del file di debug per un comando non riuscito. È tuttavia possibile scrivere le informazioni di debug e l'output dei comandi in un file in una directory diversa aggiungendo `-d` assieme al nome del file di debug e ai flussi di errori e output (`-d >> c:\logs\LOG_FILE_NAME.log 2>&1`) alla fine del comando come illustrato nel seguente esempio per Windows:

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

Abilitazione del protocollo Transport Layer Security 1.2

È necessario installare EPM Automate in un sistema operativo che supporti il protocollo Transport Layer Security (TLS) 1.2 o versioni successive.

Per garantire il livello massimo di sicurezza per l'autenticazione e per la cifratura dei dati, EPM Automate supporta solo TLS 1.2. Se sul computer in cui viene eseguito EPM Automate non è abilitato il protocollo TLS 1.2, viene visualizzato il messaggio di errore `EPMAT-7: Unable to connect. Unsupported Protocol: HTTPS`. Per risolvere questo problema, collaborare con l'amministratore IT per abilitare TLS 1.2.

La procedura per l'abilitazione di TLS 1.2 dipende dal sistema operativo. Utilizzare le fonti di informazioni elencate di seguito. Potrebbero essere disponibili risorse Web simili per altri sistemi operativi supportati.

- [Aggiornamento per abilitare TLS 1.1 e 1.2 TLS come protocolli di protezione predefiniti in WinHTTP in Windows](#) per informazioni sull'abilitazione di TLS 1.2 per computer Windows.
- [Hardening TLS Configuration](#) (in lingua inglese) per informazioni sull'abilitazione di TLS 1.2 in OpenSSL per Red Hat Enterprise Linux.

Utilizzo dei comandi di EPM Automate

Sequenza dei parametri di comando

Tutti i parametri obbligatori di un comando devono essere passati nella sequenza identificata nell'uso del comando. I parametri obbligatori e i relativi valori precedono i parametri facoltativi, che possono essere passati in qualsiasi sequenza. I parametri facoltativi non sono posizionali.

Si consideri ad esempio il seguente uso del comando `login`:

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

Questo comando prevede tre parametri obbligatori: `USERNAME`, `PASSWORD` e `URL`, che devono comparire secondo la sequenza indicata nell'uso. Il comando restituirà un errore se non viene mantenuta questa sequenza. I parametri facoltativi `ProxyServerUserName`, `ProxyServerPassword` e `ProxyServerDomain` e i relativi valori possono essere specificati in qualsiasi sequenza.

Distinzione tra maiuscole e minuscole nei comandi di EPM Automate

I comandi di EPM Automate non fanno distinzione tra maiuscole e minuscole. Il modo in cui viene scritto un comando non ha effetto sull'esecuzione. È ad esempio possibile digitare il comando `addUsers` come `addusers`, `ADDUSERS` o `AdDuSeRs`.

Distinzione tra maiuscole e minuscole nei parametri dei comandi di EPM Automate

I parametri dei comandi di EPM Automate non fanno distinzione tra maiuscole e minuscole. Il modo in cui viene scritto un parametro di un comando non ha effetto sull'esecuzione del comando stesso. Ad esempio, è possibile immettere il parametro `FileName` come `filename`, `fileName` o `filEnAmE` senza alcuna ripercussione sull'esecuzione del comando.

Definizione di valori multipli per un parametro

Alcuni comandi di EPM Automate accettano più valori di parametro separati da virgole, ad esempio un prompt runtime del tipo Membri in regole business, set di regole e template in un'applicazione Planning.

Per impostare più di un membro per un tipo Membri del prompt runtime denominato `Entities` in un comando di EPM Automate, utilizzare una virgola (,) come illustrato nell'esempio seguente per l'esecuzione del comando `runbusinessrule`.

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

I nomi dei membri contenenti caratteri speciali come lo spazio e la virgola devono essere racchiusi tra virgolette aggiungendo il carattere di escape `\` (barra rovesciata) come mostrato nell'esempio seguente:

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

Comportamento durante la manutenzione giornaliera

Non eseguire i comandi EPM Automate mentre è in corso la manutenzione giornaliera di un ambiente.

Attività utente non consentita durante la manutenzione giornaliera. Se si tenta di eseguire i comandi di EPM Automate, direttamente o mediante gli script, mentre è in corso la manutenzione giornaliera verrà visualizzato l'errore seguente:

EPMAT-11:Internal server error. A causa della manutenzione giornaliera, l'ambiente Oracle EPM Cloud Service non è attualmente disponibile.

Esecuzione di EPM Automate

Utilizzare le credenziali di Oracle Enterprise Performance Management Cloud per effettuare l'accesso mediante EPM Automate. Non è possibile effettuare l'accesso utilizzando le credenziali SSO.

Tutti gli utenti di EPM Cloud possono utilizzare le credenziali del proprio dominio di Identity per connettersi a un ambiente mediante EPM Automate. I comandi che un utente può eseguire dipendono dai ruoli predefiniti e dell'applicazione che gli sono stati assegnati.

Inoltre, solo gli amministratori servizi possono eseguire alcuni comandi, mentre potrebbe essere necessario disporre anche dei ruoli amministratore del dominio di Identity per eseguire altri comandi.

Generazione del file di log di debug

Per risolvere i problemi rilevati durante l'esecuzione di EPM Automate, è necessario fornire al Supporto Oracle il file di log di debug della sessione. EPM Automate supporta l'opzione `-d` per la generazione dei messaggi di debug, che possono quindi essere reindirizzati a un file mediante la direttiva `>`. È possibile creare un file di debug per un comando oppure un file o uno script di esecuzione batch contenente diversi comandi.

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

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

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

Windows

Prima di eseguire EPM Automate, assicurarsi di essere in grado di accedere all'ambiente dal computer da cui si sta eseguendo EPM Automate.

EPM Automate crea un file `.prefs`, contenente le informazioni utente, e file di log nella directory corrente. Nei computer Windows, il contenuto del file `.prefs` è visibile solo all'utente che lo ha creato e agli amministratori di Windows. Negli ambienti Linux, UNIX e macOSX, il file `.prefs` viene generato con l'autorizzazione 600, che consente solo al proprietario di leggere e scrivere nel file.

EPM Automate visualizza l'errore `FileNotFoundException: .prefs` (Accesso negato) negli ambienti Windows se non si dispone dell'autorizzazione di scrittura nella directory di Windows da cui si esegue EPM Automate. Per risolvere questo errore, verificare che l'account Windows dell'utente corrente disponga dell'accesso di lettura/scrittura alla directory dalla quale viene eseguito EPM Automate. Inoltre questo utente deve essere dotato dell'accesso opportuno a qualsiasi altra directory dalla quale si accede a un file, ad esempio mentre è in esecuzione il comando `uploadFile`, oppure si scrive un file, ad esempio mentre è in esecuzione il comando `downloadFile`.



Nota:

Non è possibile eseguire EPM Automate da una cartella contenente & nel relativo nome, ad esempio C:\Oracle\A&B.

Per eseguire EPM Automate in un client Windows, procedere come segue.

1. Fare clic su **Start**, quindi **Tutti i programmi, Automazione EPM** e infine **Avvia Automazione EPM**. Viene visualizzato il prompt dei comandi di Automazione EPM.
2. **Facoltativo:** passare alla directory dalla quale si desidera eseguire le operazioni mediante EPM Automate.
3. **Facoltativo:** generare un file di cifratura delle password. Utilizzare il file di cifratura delle password per passare la password cifrata per avviare una sessione.

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

4. Avviare una sessione come amministratore del servizio. Utilizzare un comando come il seguente:

- Utilizzando una password non cifrata:

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

- Utilizzando una password cifrata:

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

5. Immettere i comandi per eseguire i task da completare.

Fare riferimento alla sezione [Codici di uscita](#) per informazioni sullo stato di esecuzione dei comandi.

6. Scollegarsi dall'ambiente. Utilizzare il seguente comando:

```
epmautomate logout
```

Linux



Nota:

Assicurarsi che sia impostato `JAVA_HOME` nella variabile `PATH` del file `.profile` o come variabile dell'ambiente shell. È richiesto JRE in una versione supportata (dalla 8 alla 11).

Per eseguire EPM Automate in un client Linux, procedere come segue.

1. Aprire una finestra terminale e passare alla directory di installazione di EPM Automate.

2. **Facoltativo:** generare un file di cifratura delle password. Utilizzare il file di cifratura delle password per passare la password cifrata invece do una password non cifrata per avviare una sessione.

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

3. Avviare una sessione come amministratore del servizio. Utilizzare un comando come il seguente:

- Utilizzando una password non cifrata:

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

- Utilizzando una password cifrata:

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

4. Immettere i comandi per eseguire i task da completare.

Fare riferimento alla sezione [Codici di uscita](#) per informazioni sullo stato di esecuzione dei comandi.

5. Scollegarsi dall'ambiente. Utilizzare il seguente comando:

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

Esecuzione di più istanze di EPM Automate

È possibile eseguire più istanze di EPM Automate in un unico ambiente dalla stessa directory. Allo stesso modo, è possibile eseguire più istanze di EPM Automate in ambienti diversi dalla stessa directory o da directory diverse.

Si supponga, ad esempio, di dover aggiornare contemporaneamente il cubo dell'applicazione Planning in <https://cloudpln.pbcs.us1.oraclecloud.com> e <https://testcloudpln.pbcs.us1.oraclecloud.com>. In questo scenario, è possibile procedere in due modi:

- Eseguire due istanze di EPM Automate dalla stessa directory per aggiornare i cubi dell'applicazione in ambienti diversi.
- Eseguire EPM Automate da directory separate per connettersi agli ambienti e quindi aggiornare i cubi dell'applicazione.

In entrambi gli scenari, ciascuna istanza di EPM Automate funziona in modo indipendente, pertanto la disconnessione da un'istanza non comporta la disconnessione dalle altre. Le attività avviate utilizzando EPM Automate continuano fino al completamento nell'ambiente anche se si esegue la disconnessione dall'altra istanza.

In questa sezione sono contenuti script di esempio di Windows e di Unix/Linux (`caller` e `multisession`) che possono essere utilizzati per creare due sessioni di EPM Automate per eseguire i task. Per eseguire più sessioni simultanee, è necessario aggiungere le seguenti informazioni di connessione nello script `caller`, che richiama lo script `multisession` per eseguire i comandi `login`, `uploadfile`, `listfiles` e

logout. È possibile modificare lo script `multisession` per eseguire task diversi da quelli indicati. Assicurarsi che entrambi gli script siano memorizzati nella stessa directory.

- EPM Automate utilizza la variabile di ambiente `EPM_SID` per distinguere tra più sessioni. La variabile deve essere impostata nello script chiamante su un valore univoco per ogni sessione. Negli script di esempio è impostato su valori univoci nel modo seguente:
 - In `caller.BAT`, `EPM_SID` è impostato su `!RANDOM!`, che assegna un numero univoco generato dal sistema. Questo numero viene inoltre utilizzato per generare i file di log per ogni sessione. Se si desidera tenere traccia del file di log per ogni sessione, invece di `!RANDOM!` è possibile specificare un numero univoco.
 - In `caller.sh`, `EPM_SID` è impostato sull'ID processo, che è un valore univoco. Se si desidera tenere traccia del file di log per ogni sessione, è possibile specificare un valore univoco per `EPM_SID` modificando l'istruzione `export EPM_SID=$$` nello script `multisession` in modo da utilizzare il valore passato, quindi passare un valore univoco per questo parametro nello script `caller` per ogni sessione, ad esempio specificando il valore di `EPM_SID` in `caller.sh` nel modo seguente:

```

$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: ID di accesso di un utente con ruolo Amministratore servizi
- PASSWORD: password dell'utente con ruolo Amministratore servizi
- URL: URL di connessione dell'ambiente

Script di esempio di Windows

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
```

Script di esempio della shell Bourne

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
```

Panoramica dei comandi

Di seguito è riportato l'elenco in ordine alfabetico di tutti i comandi di EPM Automate.

Tabella 2-1 Tutti i comandi di EPM Automate

Nome comando	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
addUsers	✓	✓	✓	✓	✓	✓	✓	✓
addUsersToGroup	✓	✓	✓	✓	✓	✓	✓	✓
addUsersToTeam		✓	✓			✓		
addUserToGroups	✓	✓	✓	✓	✓	✓	✓	✓
applicationAdminMode	✓	✓	✓		✓			
applyDataGrants				✓				
archiveTmTransactions						✓		
assignRole	✓	✓	✓	✓	✓	✓	✓	✓
autoPredict* Fare riferimento alla nota a piè di pagina	✓							
calculateModel					✓			
clearCube	✓				✓			
clearDataByPointOfView					✓			
clearDataByProfile		✓	✓					
clearPOV				✓				
cloneEnvironment	✓	✓	✓	✓	✓	✓	✓	✓
copyDataByPointOfView					✓			
copyDataByProfile		✓	✓					
copyFileFromInstance	✓	✓	✓	✓	✓	✓	✓	✓
copyFromObjectStorage	✓	✓	✓	✓	✓	✓	✓	✓
copyOwnershipDataToNextYear		✓	✓					
copyPOV				✓				
copySnapshotFromInstance	✓	✓	✓	✓	✓	✓	✓	
copyToObjectStorage	✓	✓	✓	✓	✓	✓	✓	✓
createGroups	✓	✓	✓	✓	✓	✓	✓	✓
createNRSnapshot								✓
createReconciliations						✓		
deleteFile	✓	✓	✓	✓	✓	✓	✓	✓
deleteGroups	✓	✓	✓	✓	✓	✓	✓	✓
deletePointOfView					✓			
deletePOV				✓				
deployCube				✓				
deployEJTemplates		✓						
deployFormTemplates		✓	✓					
deployTaskManagerTemplate		✓						
dismissIPMInsights**	✓							
downloadFile	✓	✓	✓	✓	✓	✓	✓	✓
enableApp				✓				
enableQueryTracking	✓				✓			

Tabella 2-1 (Cont.) Tutti i comandi di EPM Automate

Nome comando	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
encrypt	✓	✓	✓	✓	✓	✓	✓	✓
executeAggregationProcess	✓				✓			
executeBurstDefinition								✓
executeReportBurstingDefinition	✓	✓	✓		✓			
exportAccessControl						✓		
exportAppAudit	✓	✓	✓		✓			
exportAppSecurity	✓	✓	✓		✓			
exportARApplicationProperties						✓		
exportBackgroundImage						✓		
exportCellLevelSecurity	✓		✓		✓			
exportData	✓	✓	✓		✓			
exportConsolidationJournals		✓						
exportDataManagement	✓	✓	✓	✓	✓			
exportDimension							✓	
exportDimensionMapping							✓	
exportEJournals		✓						
exportEssbaseData	✓	✓	✓		✓			
exportJobConsole	✓	✓	✓		✓			
exportLibraryArtifact								✓
exportLibraryDocument	✓	✓	✓		✓			
exportLogoImage						✓		
exportMapping	✓	✓	✓	✓	✓	✓		
exportMetadata	✓	✓	✓		✓			
exportOwnershipData		✓	✓					
exportQueryResults				✓				
exportSnapshot	✓	✓	✓	✓	✓	✓	✓	
exportTemplate				✓				
exportTaskManagerAccessControl		✓	✓					
exportValidIntersections	✓	✓	✓		✓			
extractDimension							✓	
feedback	✓	✓	✓	✓	✓	✓	✓	✓
getApplicationAdminMode	✓	✓	✓		✓	✓		
getDailyMaintenanceStartTime	✓	✓	✓	✓	✓	✓	✓	✓
getEssbaseQryGovExecTime	✓	✓	✓	✓	✓			
getIdleSessionTimeout	✓	✓	✓	✓	✓	✓	✓	✓
getIPAllowlist	✓	✓	✓	✓	✓	✓	✓	✓
getSubstVar	✓	✓	✓		✓			
getVirusScanOnFileUploads	✓	✓	✓	✓	✓	✓	✓	✓
groupAssignmentAuditReport	✓	✓	✓	✓	✓	✓	✓	✓

Tabella 2-1 (Cont.) Tutti i comandi di EPM Automate

Nome comando	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
help	✓	✓	✓	✓	✓	✓	✓	✓
importAppAudit	✓				✓			
importAppSecurity	✓	✓	✓		✓			
importARApplicationProperties						✓		
importBackgroundImage						✓		
importBalances						✓		
importCellLevelSecurity	✓		✓		✓			
importConsolidationJournals		✓						
importData	✓	✓	✓	✓	✓			
importDataManagement	✓	✓	✓	✓	✓			
importDimension							✓	
importJobConsole	✓	✓	✓		✓			
importLibraryArtifact								✓
importLogoImage						✓		
importMapping	✓	✓	✓	✓	✓	✓		
importMetadata	✓	✓	✓		✓			
importOwnershipData		✓	✓					
importPreMappedBalances						✓		
importPreMappedTransactions						✓		
importProfiles						✓		
importRates						✓		
importRCAAttributeValues						✓		
importReconciliationAttributes						✓		
importSnapshot	✓	✓	✓	✓	✓	✓	✓	
importSupplementalCollectionData		✓	✓					
importSupplementalData		✓	✓					
importTemplate				✓				
importTMAAttributeValues						✓		
importValidIntersections	✓	✓	✓		✓			
invalidLoginReport	✓	✓	✓	✓	✓	✓	✓	✓
listBackups	✓	✓	✓	✓	✓	✓	✓	✓
listFiles	✓	✓	✓	✓	✓	✓	✓	✓
loadData				✓				
loadDimData				✓				
loadViewpoint							✓	
login	✓	✓	✓	✓	✓	✓	✓	✓
logout	✓	✓	✓	✓	✓	✓	✓	✓
maskData	✓	✓	✓		✓			
mergeDataSlices	✓				✓			

Tabella 2-1 (Cont.) Tutti i comandi di EPM Automate

Nome comando	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
mergeSlices				✓				
optimizeASOCube				✓				
programDocumentationReport				✓				
provisionReport	✓	✓	✓	✓	✓	✓	✓	✓
purgeArchivedTmTransactions						✓		
purgeTmTransactions						✓		
recomputeOwnershipData		✓	✓					
recreate	✓	✓	✓	✓	✓	✓	✓	✓
refreshCube	✓	✓	✓		✓			
removeUserFromGroups	✓	✓	✓	✓	✓	✓	✓	✓
removeUsers	✓	✓	✓	✓	✓	✓	✓	✓
removeUsersFromGroup	✓	✓	✓	✓	✓	✓	✓	✓
removeUsersFromTeam		✓	✓			✓		
renameSnapshot	✓	✓	✓	✓	✓	✓	✓	
replay	✓	✓	✓	✓	✓	✓	✓	✓
resetService	✓	✓	✓	✓	✓	✓	✓	✓
restoreBackup	✓	✓	✓	✓	✓	✓	✓	✓
restructureCube	✓	✓	✓					
roleAssignmentAuditReport	✓	✓	✓	✓	✓	✓	✓	✓
roleAssignmentReport	✓	✓	✓	✓	✓	✓	✓	✓
runAutomatch						✓		
runBatch	✓	✓	✓	✓	✓	✓		
runBusinessRule	✓	✓	✓					
runCalc				✓				
runComplianceReport						✓		
runDailyMaintenance	✓	✓	✓	✓	✓	✓	✓	✓
runDataRule	✓	✓	✓	✓	✓	✓		
runDMReport	✓	✓	✓	✓	✓	✓		
runIntegration	✓	✓	✓	✓	✓	✓		
runMatchingReport						✓		
runPlanTypeMap	✓							
runRuleSet	✓	✓	✓					
runSupplementalDataReport		✓	✓					
runTaskManagerReport		✓	✓					
sendMail	✓	✓	✓	✓	✓	✓	✓	✓
setApplicationAdminMode	✓	✓	✓		✓	✓		
setDailyMaintenanceStartTime	✓	✓	✓	✓	✓	✓	✓	✓
setDemoDates		✓	✓			✓		
setEJJournalStatus		✓						

Tabella 2-1 (Cont.) Tutti i comandi di EPM Automate

Nome comando	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
setEncryptionKey	✓	✓	✓	✓	✓	✓	✓	✓
setEssbaseQryGovExecTime	✓	✓	✓	✓	✓			
setIdleSessionTimeout	✓	✓	✓	✓	✓	✓	✓	✓
setIPAllowlist	✓	✓	✓	✓	✓	✓	✓	✓
setManualDataAccess	✓	✓	✓	✓	✓	✓	✓	✓
setPeriodStatus						✓		
setSubstVars	✓	✓	✓		✓			
setVirusScanOnFileUploads	✓	✓	✓	✓	✓	✓	✓	✓
simulateConcurrentUsage	✓	✓	✓					
skipUpdate	✓	✓	✓	✓	✓	✓	✓	✓
snapshotCompareReport	✓	✓	✓		✓			
sortMember	✓				✓			
unassignRole	✓	✓	✓	✓	✓	✓	✓	✓
updateUsers	✓	✓	✓	✓	✓	✓	✓	✓
upgrade	✓	✓	✓	✓	✓	✓	✓	✓
uploadFile	✓	✓	✓	✓	✓	✓	✓	✓
userAuditReport	✓	✓	✓	✓	✓	✓	✓	✓
userGroupReport	✓	✓	✓	✓	✓	✓	✓	✓
validateConsolidationMetadata		✓						
validateModel					✓			

- * Questo comando è supportato solo se nell'applicazione sono abilitati i cubi di Oracle Essbase ibridi. Strategic Workforce Planning e Sales Planning non supportano Essbase ibrido. Questo comando non è supportato per FreeForm.
- ** Questo comando non è supportato per FreeForm.

Abbreviazioni

- PLN: Planning (inclusi i moduli di Planning)
- FF: FreeForm
- SWP: Strategic Workforce Planning
- 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

Comandi di EPM Automate

In questa sezione sono descritti in dettagli tutti i comandi di EPM Automate. Le informazioni disponibili per ciascun comando includono i servizi che possono utilizzare il comando, la sintassi del comando e alcuni esempi.

addUsers

Crea un batch di utenti in un dominio di Identity utilizzando un file con valori separati da virgole (CSV) con codifica ANSI o UTF-8 caricato nell'ambiente. Inoltre fornisce ai nuovi utenti il loro nome utente e una password temporanea.

Utilizzare il comando [uploadFile](#) per caricare file in un ambiente. Tutte le colonne nel file CSV sono obbligatorie. Questo comando convalida il valore di ogni colonna di una definizione e visualizza messaggi di errore che identificano ogni valore mancante o non valido. Il formato del file CSV è il seguente:

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

Per una descrizione dettagliata del formato file CSV, fare riferimento alla sezione *Importazione di un batch di account utente* in *Introduzione a Oracle Cloud*.

Per il valore di `User Login` specificato nel file di importazione non è prevista la distinzione tra maiuscole e minuscole. Ad esempio, il valore `John.doe@example.com` viene considerato uguale a `John.Doe@example.com` o a qualsiasi variante di maiuscole e minuscole.

Se una definizione utente nel file CSV corrisponde a un account utente esistente nel dominio di Identity, non verrà apportata alcuna modifica all'account utente esistente. Questo comando crea account solo per i nuovi utenti con informazioni sugli account incluse nel file. Poiché gli account utente sono comuni a tutti gli ambienti supportati dal dominio di Identity, i nuovi utenti sono disponibili per tutti gli ambienti che condividono tale dominio.

Quando termina l'esecuzione del comando, EPM Automate invia in stampa alla console le informazioni su ogni singola voce non riuscita. Esaminare queste informazioni per comprendere il motivo per cui l'esecuzione del comando non è riuscita per alcune voci nel file CSV.

Si applica a

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, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore del dominio di Identity e qualsiasi ruolo predefinito (Amministratore servizi, Utente avanzato, Utente o Visualizzatore)

Utilizzo

`epmautomate addUsers FILE_NAME [userPassword=PASSWORD] [resetPassword=true|false]` dove:

- `FILE_NAME` è il nome di un file CSV contenente le informazioni sugli utenti. Il file di input contenente caratteri multibyte deve utilizzare la codifica di caratteri UTF-8. L'uso della codifica ANSI causa problemi nel modo in cui le informazioni dell'utente vengono visualizzate nelle schermate di Servizi personali.
- `userPassword` (facoltativo) indica la password predefinita di tutti i nuovi utenti creati nel dominio di Identity. Se viene specificata, questa password deve soddisfare i requisiti minimi per le password del dominio di Identity. Se il parametro non è specificato, a ogni utente viene assegnata una password temporanea univoca.
Se specificato, il valore del parametro `userPassword` viene utilizzato come password per tutti gli utenti specificati nel file CSV. L'assegnazione della stessa password a tutti gli utenti potrebbe essere una soluzione consigliata in caso di creazione di utenti solo a scopo di test. Se si stanno creando utenti reali per Oracle Enterprise Performance Management Cloud e si desidera assegnare a ognuno di essi una specifica password, utilizzare questo comando senza specificare un valore per il parametro facoltativo `userPassword`.
- `resetPassword` (facoltativo) indica se i nuovi utenti devono modificare la password al primo accesso. Il valore predefinito è `true`. A meno che questo parametro non venga impostato su `false`, i nuovi utenti dovranno modificare la password al primo accesso. Questo comando invia a ogni nuovo utente un messaggio e-mail con i dettagli relativi agli account (nome utente e password) se l'impostazione di `resetPassword` è `true`. Se l'impostazione di `resetPassword` è `false`, il messaggio e-mail non viene inviato. Se si imposta `resetPassword` su `false`, è necessario specificare `userPassword`. In caso contrario, verrà assegnata una password temporanea univoca a ogni nuovo utente, ma poiché non viene inviato alcun messaggio e-mail, gli utenti non conosceranno le password e non potranno eseguire l'accesso.

Esempi

- Aggiungere utenti di test nel dominio di Identity con la stessa password senza l'obbligo di cambiare la password:

```
epmautomate addUsers user_file.CSV userPassword=Example@PwD12
resetPassword=false
```
- Aggiungere utenti al dominio di Identity utilizzando una password temporanea e chiedendo loro di cambiarla:

```
epmautomate addUsers user_file.CSV
```

addUsersToGroup

Aggiunge un batch di utenti a un gruppo esistente in Controllo accesso utilizzando un file CSV con codifica ANSI o UTF-8 caricato nell'ambiente.

Utilizzare il comando [uploadFile](#) per caricare file in un ambiente. Per il valore di User Login non è prevista la distinzione tra maiuscole e minuscole. Il formato del file è il seguente:

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

 **Nota:**

L'utente viene aggiunto al gruppo solo se vengono soddisfatte entrambe le seguenti condizioni:

- Il valore di User Login incluso nel file esiste nel dominio di Identity che gestisce l'ambiente. Per i valori di User Login non è prevista la distinzione tra maiuscole e minuscole.
- L'utente è assegnato a un ruolo predefinito nel dominio di Identity

Quando termina l'esecuzione del comando, EPM Automate invia in stampa alla console le informazioni su ogni singola voce non riuscita. Esaminare queste informazioni per comprendere il motivo per cui l'esecuzione del comando non è riuscita per alcune voci nel file CSV.

Si applica a

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, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi o Gestione controllo accesso

Utilizzo

`epmautomate addUsersToGroup FILE_NAME GROUP_NAME` dove:

- *FILE_NAME* rappresenta il nome di un file CSV contenente i nomi di accesso degli utenti che si desidera assegnare a un gruppo in Controllo accesso.
- *GROUP_NAME* rappresenta il nome di un gruppo esistente in Controllo accesso. Per questo valore non è prevista la distinzione tra maiuscole e minuscole.

Esempio

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

addUsersToTeam

Aggiunge a un team esistente gli utenti di Oracle Enterprise Performance Management Cloud elencati in un file CSV.

Se un utente incluso nel file CSV è già membro del team, questo comando lo ignora. Per i valori in questo file non è prevista la distinzione tra maiuscole e minuscole. Il formato del file CSV è il seguente:

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



Nota:

Per impostazione predefinita, viene designato un utente principale per eseguire i task assegnati al team.

Si applica a

Financial Consolidation and Close, Tax Reporting e Account Reconciliation.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

`epmautomate addUsersToTeam FILE TEAM_NAME dove:`

- `FILE` identifica un file CSV in formato UTF-8 in cui sono elencati gli ID di accesso degli utenti da aggiungere al team. Prima di eseguire questo comando, utilizzare il comando [uploadFile](#) per caricare il file in un ambiente.
- `TEAM_NAME` identifica un nome di team definito in Controllo accesso. Per questo valore non è prevista la distinzione tra maiuscole e minuscole.

Esempio

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

addUserToGroups

Aggiunge un utente come membro dei gruppi di controllo dell'accesso definiti in un file CSV con codifica ANSI o UTF-8.

Utilizzare il comando [uploadFile](#) per caricare file in un ambiente. Il formato del file è il seguente:

```
Group Name
Group1
Group2
```

Per i valori di Group Name non è prevista la distinzione tra maiuscole e minuscole.

Quando termina l'esecuzione del comando, EPM Automate invia in stampa alla console le informazioni su ogni singola voce non riuscita. Esaminare queste informazioni per comprendere il motivo per cui l'esecuzione del comando non è riuscita per alcune voci nel file CSV.

Si applica a

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, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi o Gestione controllo accesso

Utilizzo

```
epmautomate addUserToGroups FILE_NAME User_Login, dove:
```

- *FILE_NAME* è il nome di un file CSV contenente i nomi dei gruppi di controllo dell'accesso a cui si desidera assegnare l'utente
- *User_Login* è l'ID di accesso di un utente di Oracle Enterprise Performance Management Cloud che deve essere assegnato ai gruppi Controllo accesso. Questo ID di accesso dell'utente, per cui non è prevista la distinzione tra maiuscole e minuscole, deve esistere nel dominio di Identity che gestisce l'ambiente e deve disporre di un ruolo predefinito assegnato.

Esempio

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

applicationAdminMode

Imposta l'applicazione in modalità amministrazione in modo che l'accesso all'applicazione sia limitato solo agli amministratori del servizio.

Questo comando è utile per impedire agli utenti di utilizzare l'applicazione quando gli amministratori del servizio stanno eseguendo operazioni amministrative. L'applicazione rimane in modalità amministrazione fino a quando non la si modifica in modo che tutti gli utenti possano accedervi.



Nota:

Questo comando non è più valido, ma non è stato rimosso da EPM Automate. Oracle consiglia di utilizzare invece il comando [setApplicationAdminMode](#).

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate applicationAdminMode VALUE dove VALUE specifica se impostare l'applicazione in modalità amministrazione. Di seguito sono riportati i valori consentiti.
```

- `true` per impostare l'applicazione in modalità amministrazione

- `false` per riportare l'applicazione alla modalità normale in modo che tutti gli utenti possano accedervi

Esempi

- Impostare l'applicazione in modalità amministrazione: `epmautomate applicationAdminMode true`
- Riportare l'applicazione alla modalità normale: `epmautomate applicationAdminMode false`

applyDataGrants

Aggiorna le autorizzazioni dati, che controllano l'accesso alle sezioni dati di Oracle Essbase in modo che corrispondano alle autorizzazioni dati definite in un'applicazione Profitability and Cost Management.

Le autorizzazioni dati a livello di utente e gruppo definite nell'applicazione Profitability and Cost Management vengono automaticamente sincronizzate in Essbase. Utilizzare questo comando per sincronizzare l'accesso ai dati di Essbase se si presume l'esistenza di una discordanza tra l'autorizzazione dati nell'applicazione e i filtri in Essbase.

Il tempo richiesto per completare questa operazione dipende dalle dimensioni dell'applicazione. Assicurarsi che l'operazione di aggiornamento dell'autorizzazione dati venga completata prima che venga creato il backup dell'applicazione durante la successiva finestra di manutenzione. Dal momento che l'applicazione non deve essere utilizzata mentre questa operazione è in corso, Oracle consiglia di programmare questa operazione quando gli utenti non stanno utilizzando l'applicazione.

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate applyDataGrants APPLICATION_NAME` dove `APPLICATION_NAME` è il nome dell'applicazione Profitability and Cost Management per la quale è necessario ricreare le autorizzazioni dati.

Esempio

`epmautomate applyDataGrants BksML12`

archiveTmTransactions

Archivia le transazioni con corrispondenza risalenti a un numero di giorni uguale o superiore a un valore specificato, inclusi i relativi dettagli di supporto e adeguamento. Le transazioni con corrispondenza vengono registrate in un file ZIP.

Utilizzare questo comando per mantenere la dimensione dell'applicazione Account Reconciliation a livelli ottimali archiviando e quindi rimuovendo le transazioni con corrispondenza meno recenti secondo i criteri di conservazione delle transazioni applicati dall'organizzazione.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

`epmautomate archiveTmTransactions matchType age [filterOperator=VALUE] [filterValue=VALUE] [logFilename=FILE_NAME] [filename=FILE_NAME]`, I parametri sono descritti di seguito.

- `matchType` indica l'identificativo (TextID) del tipo di corrispondenza da cui devono essere archiviate le transazioni con corrispondenza.
- `age` indica da quanti giorni è stata eseguita la corrispondenza della transazione. Una transazione con corrispondenza che risale a un numero di giorni superiore o uguale a questo valore verrà archiviata.
- `filterOperator` (facoltativo) è una delle condizioni di filtro elencate di seguito per l'identificazione dei conti contenenti le transazioni con corrispondenza da archiviare. Questo valore viene utilizzato insieme al valore `filterValue` per identificare i conti a partire da cui devono essere archiviate le transazioni con corrispondenza:
 - `equals`
 - `not_equals`
 - `starts_with`
 - `ends_with`
 - `contains`
 - `not_contains`
- `filterValue` (facoltativo) indica un valore di filtro che identifica le transazioni da archiviare. Se il valore di `filterOperator` è `equals` o `not_equals`, è possibile utilizzare un elenco separato da spazi per specificare più valori, ad esempio `filterValue=101-120 filterValue=102-202`. Se vengono specificati più valori, vengono selezionate per l'archiviazione le transazioni contenute in conti che soddisfano la combinazione operatore e valore di filtro.

Note:

Se non vengono specificati i valori `filterOperator` e `filterValue`, vengono archiviate tutte le transazioni con corrispondenza risalenti a un numero di giorni superiore o uguale al valore specificato per `age` contenute in tutti i conti corrispondenti al valore di `matchType` specificato.

- `logFilename` (facoltativo) indica il nome di un file di log per la registrazione di informazioni sull'attività dei comandi. Se non viene specificato un nome di file,

viene generato automaticamente un file di log denominato
Archive_Transactions_matchType_JOBID.log.

- `filename` (facoltativo) indica il nome di un file .ZIP che deve contenere le transazioni archiviate. Se non viene specificato, per impostazione predefinita il comando crea `Archived_Transactions_matchType_JOBID.zip`. Utilizzare il comando [downloadFile](#) per scaricare questo file in un computer locale.

Note:

Questo comando esegue il job di archiviazione delle transazioni di Corrispondenza transazione utilizzando i parametri specificati. L'ID del job viene restituito nell'output del comando per facilitarne l'utilizzo con il comando [purgeArchivedTmTransactions](#). È possibile monitorare il job dalla console job.

Esempi

- Archiviazione di transazioni con corrispondenza meno recenti senza l'utilizzo di filtri, bensì tramite un nome di file di log e .ZIP customizzato:

```
epmautomate archiveTmTransactions cashrecon 180 logFile=tmlogs.log  
filename=trans.zip
```
- Archiviazione di transazioni con corrispondenza meno recenti mediante filtri:
 - ```
epmautomate archiveTmTransactions cashrecon 180 filterOperator>equals
filterValue=101-120 FilterValue=102-202
```
  - ```
epmautomate archiveTmTransactions cashrecon 180 filterOperator=contains  
filterValue=11
```

assignRole

Assegna un ruolo agli utenti (incluso quello che esegue il comando) i cui ID di accesso sono inclusi nel file CSV con codifica ANSI o UTF-8. Utilizzare questo comando per assegnare gli utenti a un ruolo predefinito o a un ruolo applicazione appartenente a un'applicazione Planning, Financial Consolidation and Close, Tax Reporting, Account Reconciliation o Oracle Enterprise Data Management Cloud.

Nota:

Non è possibile utilizzare questo comando per assegnare ruoli applicazione ad applicazioni Profitability and Cost Management e Narrative Reporting, che non supportano l'assegnazione di ruoli applicazione agli utenti.

Prima di eseguire questo comando, utilizzare il comando [uploadFile](#) per caricare il file in un ambiente. Il formato del file è il seguente:

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

Fare riferimento alla sezione Assegnazione di un ruolo a più utenti della *Guida introduttiva a Oracle Cloud*.

 **Nota:**

- Per i valori di User Login inclusi nel file non è prevista la distinzione tra maiuscole e minuscole.
- Utilizzare le virgolette doppie per racchiudere i nomi dei ruoli contenenti uno spazio.

Quando termina l'esecuzione del comando, EPM Automate invia in stampa alla console le informazioni su ogni singola voce non riuscita. Esaminare queste informazioni per comprendere il motivo per cui l'esecuzione del comando non è riuscita per alcune voci nel file CSV.

Si applica a

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, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Per assegnare ruoli predefiniti, fare riferimento a quanto segue.

- Ambienti Classic: amministratore del dominio di Identity e qualsiasi ruolo predefinito (amministratore servizi, utente avanzato, utente o visualizzatore)
- Ambienti OCI: amministratore servizi o amministratore del dominio di Identity e qualsiasi ruolo predefinito (amministratore servizi, utente avanzato, utente o visualizzatore)

Per assegnare ruoli applicazione: Amministratore servizi o Gestione controllo accesso

Utilizzo

`epmautomate assignRole FILE_NAME ROLE dove:`

- `FILE_NAME` indica il nome di un file CSV contenente gli ID di accesso degli utenti. Specificare l'estensione CSV in caratteri minuscoli.
- `ROLE` indica uno degli elementi descritti di seguito. Per questo valore non è prevista la distinzione tra maiuscole e minuscole.
 - Se agli utenti vengono assegnati ruoli predefiniti del dominio di Identity, `ROLE` deve identificare un ruolo predefinito applicabile al servizio. Fare riferimento alla sezione Introduzione ai ruoli predefiniti nella *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori*. Per una descrizione di questi ruoli, fare riferimento alla sezione Gestione delle assegnazioni di ruoli a livello di applicazione in *Amministrazione del controllo dell'accesso per Oracle Enterprise Performance Management Cloud*
 - Se si sta effettuando l'assegnazione di utenti a ruoli applicazione, `ROLE` deve identificare un ruolo appartenente all'applicazione nell'ambiente corrente. I

ruoli applicazione sono elencati nella scheda **Ruoli** di Controllo accesso. Per una descrizione dei ruoli applicazione per ciascun processo aziendale, fare riferimento agli argomenti della guida *Amministrazione del controllo dell'accesso per Oracle Enterprise Performance Management Cloud* elencati di seguito.

- * Planning, Financial Consolidation and Close e Tax Reporting
- * Enterprise Profitability and Cost Management
- * Oracle Enterprise Data Management
- * Profitability and Cost Management
- * Account Reconciliation

Esempi

- Assegnare gli utenti a un ruolo di dominio di Identity predefinito:
`epmautomate assignRole admin_role_file.csv "Service Administrator"`
- Assegnare gli utenti a un ruolo applicazione:
`epmautomate assignRole example_file.csv "Task List Access Manager"`

autoPredict

Genera previsioni di prestazioni future in base a una definizione di previsione automatica esistente in Planning o Planning Modules.

Questo comando avvia un job che utilizza i dati cronologici di ciascun membro incluso nella definizione di previsione automatica specificata nell'applicazione. Per informazioni dettagliate sulle applicazioni che utilizzano la funzione Previsione automatica e sull'impostazione delle previsioni, fare riferimento alla sezione Impostazione dell'esecuzione automatica delle previsioni con Previsione automatica nella Guida *Amministrazione di Planning*.

Si applica a

Planning, Planning Modules, se nell'applicazione sono abilitati i cubi Oracle Essbase ibridi.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate autoPredict PREDICTION_DEFINITION [forceRun=true|false]
[paginatedDim=DIMENSION_NAME] dove:
```

- `PREDICTION_DEFINITION` è il nome di una definizione di previsione automatica disponibile nell'applicazione.
- `forceRun` (facoltativo) specifica se eseguire la previsione se la definizione sottostante non è cambiata dopo l'esecuzione iniziale. Il valore predefinito è `false`. Impostare il valore di questo parametro su `true` per eseguire il job di previsione automatica se non sono presenti modifiche nella definizione del job. Utilizzare il valore predefinito (`false`) per eseguire la previsione una volta alla prima esecuzione del job.
- `paginatedDim` (facoltativo) specifica una dimensione utilizzata per velocizzare un job di previsione automatica eseguendo previsioni in parallelo all'interno di thread separati. Per ottimizzare l'efficienza di questi thread paralleli, specificare una dimensione che restituisca dati distribuiti uniformemente per ciascun thread di previsione.

Esempio

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

calculateModel

Esegue il processo di calcolo nelle applicazioni Enterprise Profitability and Cost Management.

Si applica a

Enterprise Profitability and Cost Management

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate calculateModel POV_NAME MODEL_NAME EXECUTION_TYPE
[povDelimiter=DELIMITER] [optimizeForReporting=true|false]
[captureDebugScripts=true|false] [comment=COMMENT] [PARAMETER=VALUE]. I
parametri sono descritti di seguito.
```

- *POV_NAME* indica il nome del POV dati da calcolare. Per calcolare più POV, elencare i relativi nomi separandoli utilizzando una virgola come delimitatore. Non utilizzare altri delimitatori per separare i nomi dei POV. Se sono presenti spazi nei nomi dei membri, racchiudere l'elenco dei nomi di POV tra virgolette.
- *MODEL_NAME* indica il nome del modello da calcolare. Se il nome del modello contiene spazi, racchiuderlo tra virgolette.
- *EXECUTION_TYPE* può utilizzare uno dei valori elencati di seguito, che identifica il tipo di esecuzione delle regole.
 - *ALL_RULES* per utilizzare tutte le regole per calcolare il POV.
Se si specifica questo valore, non indicare i parametri runtime correlati a un subset di regole o a una singola regola, ad esempio *rulesetSeqNumStart*, *rulesetSeqNumEnd* e *ruleName*.
 - *RULESET_SUBSET* per utilizzare un subset di un set di regole per calcolare il POV.
Se si utilizza questo valore, è necessario specificare i valori *rulesetSeqNumStart* e *rulesetSeqNumEnd* come parametri runtime.
 - *SINGLE_RULE* per eseguire una specifica regola per il calcolo del POV.
Se si utilizza questo valore, è necessario specificare solo un valore *ruleName* come parametro runtime.
 - *RUN_FROM_RULE* per eseguire i calcoli su un POV a partire da una regola specifica.
Se si utilizza questo valore, è necessario specificare solo un valore *ruleName* come parametro runtime.
 - *STOP_AFTER_RULE* per interrompere il calcolo del POV dopo che una regola specifica ha terminato i calcoli.
Se si utilizza questo valore, è necessario specificare solo un valore *ruleName* come parametro runtime.

- `povDelimiter` (facoltativo) indica il delimitatore utilizzato nei valori POV. Il delimitatore predefinito è `_` (carattere di sottolineatura). Il delimitatore deve essere racchiuso tra virgolette. Sono supportati solo i delimitatori elencati di seguito.
 - `_` (carattere di sottolineatura)
 - `#` (cancelletto)
 - `&` (E commerciale)
 - `~` (tilde)
 - `%` (percentuale)
 - `;` (punto e virgola)
 - `:` (due punti)
 - `-` (trattino)
- `optimizeForReporting=true|false` (facoltativo) specifica se i calcoli devono essere eseguiti con o senza ottimizzazione per il reporting. Il valore predefinito è `false`. Impostare questo valore su `false` per risparmiare tempo di elaborazione saltando il passo di creazione dell'aggregazione, ad esempio durante l'esecuzione di una singola regola o di una serie sequenziale di POV. Durante l'esecuzione di più job di calcolo concorrenti, impostare `optimizeForReporting=true` per tutti i job. In questo modo, solo l'ultimo job da completare eseguirà l'aggregazione, evitando un'elaborazione ridondante e impedendo il rallentamento dei job in esecuzione.
- `captureDebugScripts=true|false` (facoltativo) indica se generare script di debug nella cartella inbox. È possibile che Oracle necessiti di tali script per la risoluzione dei problemi di calcolo. Il valore predefinito è `false`.
- `comment="COMMENT"` (facoltativo) specifica un commento sul processo indicato tra virgolette.
- `PARAMETER=VALUE` (facoltativo) indica i parametri runtime e i relativi valori per l'esecuzione del calcolo. Specificare il numero di coppie parametro/valore richiesto dal processo. I parametri validi e i relativi valori sono indicati di seguito.
 - `rulesetSeqNumStart` specifica il numero di sequenza della prima regola nel set di regole da eseguire. Valido solo se viene utilizzato `EXECUTION_TYPE=RULESET_SUBSET`.
 - `rulesetSeqNumEnd` specifica il numero di sequenza dell'ultima regola nel set di regole da eseguire. Valido solo se viene utilizzato `EXECUTION_TYPE=RULESET_SUBSET`.
 - `ruleName` specifica il nome della regola da eseguire. Se il valore contiene spazi, racchiuderlo tra virgolette. Valido solo se il valore di `EXECUTION_TYPE` è impostato su `SINGLE_RULE`, `RUN_FROM_RULE` o `STOP_AFTER_RULE`.
 - `clearCalculatedData=true|false` specifica se cancellare i calcoli esistenti. Il valore predefinito è `false`.
 - `executeCalculations=true|false` specifica se eseguire i calcoli. Il valore predefinito è `false`.

 **Note:**

I valori dei parametri (`true` e `false`) devono essere specificati in caratteri minuscoli.

Esempi

- **Esecuzione di tutte le regole per il calcolo di un singolo POV:**

```
epmautomate calculateModel FY21_Jan_Actual_Working ForecastingModel
ALL_RULES clearCalculatedData=true executeCalculations=true
optimizeForReporting=true comment="Running all rules to calculate a
POV"
```
- **Esecuzione di tutte le regole per il calcolo di più POV:**

```
epmautomate calculateModel
"FY21:Jan:Actual:Working,FY21:Feb:Actual:Working,FY21:Mar:Actual:Worki
ng" "10 Actuals Allocation Process" ALL_RULES clearCalculatedData=true
executeCalculations=true optimizeForReporting=true
captureDebugScripts=true comment="Test calculation of many POVs"
povDelimiter=":"
```
- **Esecuzione di un subset di un set di regole per il calcolo del POV:**

```
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"
```
- **Esecuzione di una regola specifica per il calcolo del POV:**

```
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"
```
- **Esecuzione di tutte le regole per il calcolo di un singolo POV tramite un delimitatore di POV customizzato:**

```
epmautomate calculateModel FY21:Jan:Actual_Working ForecastingModel
ALL_RULES clearCalculatedData=true executeCalculations=true
optimizeForReporting=true comment="Running all rules to calculate a
POV" povDelimiter=":"
```
- **Esecuzione di tutte le regole per il calcolo dei POV e del modello con spazi nei nomi:**

```
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

Elimina dati specifici dai cubi di input e reporting usando le impostazioni specificate in un job di tipo `clear cube`.

Questo comando non elimina la definizione dell'applicazione nelle tabelle relazionali dell'applicazione. Fare riferimento alla sezione Cancellazione di cubi nella Guida *Amministrazione di Planning*.

Si applica a

Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Strategic Workforce Planning Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate clearCube JOB_NAME` dove `JOB_NAME` è il nome di un job definito nell'applicazione.

Esempio

```
epmautomate clearCube ClearPlan1
```

clearDataByPointOfView

Cancella i dati per uno specifico POV per un cubo Enterprise Profitability and Cost Management.

Si applica a

Enterprise Profitability and Cost Management

Ruoli necessari

Amministratore servizi

Utilizzo

`epmAutomate clearDataByPointOfView POV_NAME [cubeName=CUBE_NAME] [PARAMETER=VALUE]`. I parametri sono descritti di seguito.

- `POV_NAME` indica il nome di un POV nell'applicazione.
- `cubeName` (facoltativo) è il nome del cubo in cui devono essere cancellati i dati. Il valore predefinito è `PCM_CLC`.
- `PARAMETER=VALUE` indica i parametri runtime facoltativi e i relativi valori. Specificare il numero di parametri e coppie di valori richiesto dal processo. I parametri validi e i relativi valori sono indicati di seguito.
 - `povDelimiter` indica il delimitatore utilizzato nei valori POV. Il valore predefinito è `::` (doppio carattere di due punti). Questo valore deve essere racchiuso tra virgolette. Esempio: `povDelimiter="_"`.
Oltre al delimitatore predefinito, sono supportati solo questi delimitatori: `_` (carattere di sottolineatura), `#` (cancelletto), `&` (E commerciale), `~` (tilde), `%` (percentuale), `;` (punto e virgola), `:` (due punti), `-` (trattino).
 - `clearInput=true|false` specifica se cancellare i dati di input. Il valore predefinito è `false`.
 - `clearAllocatedValues=true|false` specifica se cancellare i valori allocati. Il valore predefinito è `false`.
 - `clearAdjustmentValues=true|false` specifica se cancellare i valori di adeguamento. Il valore predefinito è `false`.

 **Note:**

- * I valori dei parametri (`true` o `false`) devono essere in minuscolo.
- * Almeno uno dei parametri `clearInput`, `clearAllocatedValues` o `clearAdjustmentValues` deve essere impostato su `true`.

Esempi

- Cancellazione dei dati da un POV nel cubo PCM_CLC predefinito con il delimitatore di POV predefinito:

```
epmAutomate clearDataByPointOfView FY21::Jan::Actual::Working
clearInput=true clearAllocatedValues=true clearAdjustmentValues=true
```
- Cancellazione dei dati di input e dei valori allocati da un POV in un cubo specifico con un delimitatore di POV customizzato:

```
epmAutomate clearDataByPointOfView FY21_Jan_Actual_Working
cubeName=PCM_REP povDelimiter="_" clearInput=true
clearAllocatedValues=true
```
- Cancellazione dei dati di input da un POV in un cubo specifico con un delimitatore di POV customizzato:

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

clearDataByProfile

Cancella i dati dagli elementi (ad esempio le aree) identificati in un profilo di cancellazione dati definito in Financial Consolidation and Close e Tax Reporting.

Si applica a

Financial Consolidation and Close, Tax Reporting

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate clearDataByProfile PROFILE_NAME` dove `PROFILE_NAME` è il nome di un profilo di cancellazione dati.

Esempio

```
epmautomate clearDataByProfile clearDataProfile_01
```

clearPOV

Cancella gli artifact e i dati del modello da una combinazione di punti di vista (POV) o da un'area dati all'interno del POV in un'applicazione Profitability and Cost Management.

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate clearPOV APPLICATION_NAME POV_NAME [QUERY_NAME] PARAMETER=VALUE stringDelimiter="DELIMITER" dove:`

- `APPLICATION_NAME` è il nome di un'applicazione Profitability and Cost Management
- `POV_NAME` è un POV nell'applicazione. Questo valore è obbligatorio.
- `QUERY_NAME`, facoltativamente, è il nome di una query esattamente come definito in Profitability and Cost Management. Se specificato, la query corrispondente verrà utilizzata per cancellare l'area dati all'interno del POV.

Nota:

Se si specifica il nome di una query, è necessario impostare su false il valore di tutti i parametri runtime (vedere sotto).

- `PARAMETER=VALUE` indica i parametri runtime e i relativi valori per la cancellazione del POV. Specificare il numero di parametri e coppie di valori richiesto dal processo. Di seguito sono riportati i parametri validi con i relativi valori. Almeno un parametro è obbligatorio.
 - `isManageRule=true|false` specifica se cancellare le regole
 - `isInputData=true|false` specifica se cancellare i dati di input
 - `isAllocatedValues=true|false` specifica se cancellare i valori di allocazione
 - `isAdjustmentValues=true|false` specifica se cancellare i valori di adeguamento

Nota:

I valori dei parametri (`true` o `false`) devono essere in minuscolo.

Per cancellare le aree dati in un POV (se viene specificato un `QUERY_NAME`), è necessario impostare il valore dei parametri runtime (`isManageRule`, `isInputData`, `isAllocatedValues` e `isAdjustmentValues`) **SU** `false`.

- `stringDelimiter="DELIMITER"` specifica il delimitatore utilizzato nei valori POV. Il delimitatore deve essere racchiuso tra virgolette.

Esempi

- **Cancellare tutti gli artifact e i dati del modello da un POV:** `epmautomate clearPOV BksML12 2012_Jan_Actual isManageRule=true isInputData=true isAllocatedValues=true isAdjustmentValues=true stringDelimiter="_"`
- **Svuotare l'area dati all'interno di un POV:** `epmautomate clearPOV BksML12 2012_Jan_Actual queryName=BksML12_2012_Jan_clear_query isManageRule=false isInputData=false isAllocatedValues=false isAdjustmentValues=false stringDelimiter=" "`

cloneEnvironment

Clona l'ambiente corrente e, facoltativamente, gli artifact del dominio di Identity (utenti e assegnazioni di ruoli predefiniti), i record di Data Management, i record di audit, i record della console job, il contenuto delle cartelle inbox e outbox e gli snapshot memorizzati. Questo comando rappresenta un'alternativa all'utilizzo della funzione Clona ambiente in Migrazione.

 **Note:**

- **Account Reconciliation:** dopo la clonazione, le impostazioni dell'applicazione Account Reconciliation target verranno ripristinate ai valori predefiniti. Se desideri conservare le impostazioni dell'applicazione target, esportale dall'ambiente di origine utilizzando il comando [exportARApplicationProperties](#) . Quindi, al termine della clonazione, importare le proprietà dell'applicazione nell'ambiente target utilizzando il comando [importARApplicationProperties](#) .
- **Data Management:** se le tabelle intermedie includono un numero elevato di record, la clonazione dei record di Data Management può richiedere tempo. Analogamente, anche la clonazione del contenuto delle cartelle inbox e outbox e degli snapshot memorizzati può richiedere molto tempo, soprattutto se contengono una quantità elevata di dati.
- **Ambienti precedenti:** la clonazione mantiene la versione Oracle Essbase corrente come illustrato negli scenari descritti di seguito.
 - Scenario 1: si sta clonando un ambiente precedente di origine in cui viene utilizzata una versione di Essbase che non supporta i cubi ibridi verso un ambiente precedente target in cui viene utilizzata una versione di Essbase che li supporta. In questo scenario, viene eseguito il downgrade della versione di Essbase nell'ambiente target in modo che corrisponda alla versione nell'ambiente di origine.
 - Scenario 2: si sta clonando un ambiente precedente di origine in cui viene utilizzata una versione di Essbase che supporta i cubi ibridi verso un ambiente precedente target in cui viene utilizzata una versione di Essbase che non li supporta. In questo scenario, viene eseguito l'upgrade della versione di Essbase nell'ambiente target in modo che corrisponda alla versione nell'ambiente di origine.
 - Scenario 3: si sta clonando un ambiente precedente di origine in cui viene utilizzata una versione di Essbase che non supporta i cubi ibridi verso un ambiente EPM Standard Cloud Service o EPM Enterprise Cloud Service target in cui, per impostazione predefinita, viene utilizzata una versione di Essbase che li supporta. In questo scenario, non viene eseguito il downgrade della versione di Essbase nell'ambiente target in modo che corrisponda alla versione nell'ambiente di origine.
- **Planning:** la clonazione potrebbe non riuscire se il processo aziendale di Planning contiene un membro periodo generato ridenominato che è stato sostituito da un membro periodo customizzato. Ad esempio, si è rinominato il membro periodo *YearTotal* generato in *unused_YearTotal* e quindi è stato aggiunto un membro periodo di tipo alternativo con il nome del membro originale (*YearTotal* in questo esempio). In questo scenario, la clonazione dell'ambiente potrebbe non riuscire.
- **Quando eseguire questo comando:** dopo la manutenzione giornaliera pianificata degli ambienti di origine e target. L'avvio della manutenzione giornaliera degli ambienti di origine e target non può avvenire durante il processo di clonazione. Viene eseguita all'ora programmata successiva dopo che il processo di clonazione termina o viene interrotto dopo 36 ore.

Se questo comando viene eseguito mentre viene generato lo snapshot dell'ambiente di origine, durante la manutenzione giornaliera ad esempio verrà visualizzato l'errore `File not found`.

Per informazioni dettagliate su questi argomenti, fare riferimento alla sezione Clonazione degli ambienti di EPM Cloud nella Guida *Administering Migration for Oracle Enterprise Performance Management Cloud (in lingua inglese)*.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Il ruolo amministratore del dominio di Identity è necessario per clonare gli utenti e i ruoli predefiniti.

Utilizzo

```
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]. I parametri sono descritti di seguito.
```

Note:

- Il parametro `dataManagement` non si applica agli ambienti Oracle Enterprise Data Management Cloud e Narrative Reporting. Clonare record di Data Management solo se gli ambienti di origine e target sono entrambi inclusi nello stesso aggiornamento mensile o se l'ambiente target ha un solo livello di aggiornamento in più rispetto all'ambiente di origine. I record di Data Management di un ambiente 22.01 ad esempio possono essere clonati solo in un altro ambiente 22.01 o in un ambiente 22.02.
- Il parametro `jobConsole` si applica solo a Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.
- Il parametro `appAudit` si applica solo a Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning. Le informazioni di audit per Financial Consolidation and Close e Tax Reporting sono incluse nello snapshot per impostazione predefinita.
- Se il parametro `dataManagement`, `jobConsole` o `appAudit` non è applicabile a un ambiente, EPM Automate ignora il valore specificato.

- *TARGET_USERNAME* indica l'ID di un amministratore servizi nell'ambiente target. È necessario utilizzare il nome utente del dominio di Identity target, non il nome utente SSO. Se si prevede di clonare le assegnazioni di utenti e ruoli nell'ambiente target, è necessario che questo utente disponga anche del ruolo Amministratore del dominio di Identity.
- *TARGET_PASSWORD* indica la posizione del password file cifrato dell'utente identificato da *TARGET_USERNAME*.
- *TARGET_URL* indica l'URL dell'ambiente che diventerà l'ambiente clonato.
- *SnapshotName* (facoltativo) indica il nome di uno snapshot da utilizzare per la clonazione. Lo snapshot deve essere presente nell'ambiente di origine. Il valore predefinito è *Artifact Snapshot*, che utilizza lo snapshot della manutenzione più recente per clonare l'ambiente.
- *UsersAndPreDefinedRoles* (facoltativo) indica se clonare gli utenti e le relative assegnazioni di ruoli predefiniti (i gruppi Controllo accesso vengono sempre clonati). Il valore predefinito è *false*.
Per il funzionamento di questa opzione, è necessario che l'utente identificato da *TARGET_USER_NAME* disponga del ruolo di amministratore del dominio di Identity nell'ambiente target.

Se un utente che non è amministratore del dominio di Identity clona un ambiente dopo aver selezionato questa casella di controllo, l'importazione degli utenti e dei loro ruoli predefiniti non riuscirà. Nel report sullo stato della migrazione viene registrato il seguente errore: Impossibile importare l'artifact di directory esterna <artifact_name>. L'utente <user_name> non è autorizzato a eseguire questa operazione. L'utente deve disporre del ruolo Amministratore del dominio di Identity per poter eseguire questa operazione.

- Se non si stanno importando utenti e un utente nello snapshot di origine non è assegnato a un ruolo predefinito nell'ambiente target, viene visualizzato un errore (EPMIE-00070: Impossibile trovare il membro utente durante l'importazione dei ruoli assegnati).
- L'assegnazione del ruolo amministratore del dominio di Identity non viene clonata. Gli utenti che dispongono solo dell'assegnazione del ruolo amministratore del dominio di Identity non vengono clonati nell'ambiente target. Gli utenti assegnati al ruolo amministratore del dominio di Identity combinato con ruoli predefiniti nell'ambiente di origine vengono clonati, ma vengono assegnati solo ai rispettivi ruoli predefiniti nell'ambiente target. Questi utenti non disporranno del ruolo amministratore del dominio di Identity nell'ambiente target.
- Le modifiche apportate ai ruoli predefiniti dell'utente verranno aggiornate in base ai ruoli assegnati nello snapshot di origine. Le assegnazioni dei ruoli nel target tuttavia non verranno rimosse per garantire la corrispondenza con quelle nello snapshot di origine. Si supponga ad esempio che l'utente *jdoe* sia assegnato al ruolo predefinito Utente avanzato nell'ambiente target, ma che disponga solo del ruolo Utente nello snapshot di origine. In questo caso, questo comando assegna *jdoe* al ruolo Utente e non rimuove l'assegnazione del ruolo Utente avanzato nell'ambiente target.
- Questo comando non elimina gli utenti esistenti dall'ambiente target se non sono presenti nello snapshot di origine. Si supponga ad esempio che *jdoe* disponga di un account nell'ambiente target, ma che questo account non sia presente nello snapshot di origine. In questo caso, l'account di *jdoe* nell'ambiente target non viene eliminato.

- Questo comando aggiunge utenti non presenti nell'ambiente target. Non aggiorna le proprietà degli utenti correnti nell'ambiente target anche se sono diverse nello snapshot di origine. Se ad esempio il cognome di `jdoe` nello snapshot di origine è scritto in modo diverso rispetto all'ambiente target, non verranno apportate modifiche nell'ambiente target. Ai nuovi utenti nell'ambiente target viene assegnata una password casuale. I nuovi utenti riceveranno e-mail di attivazione degli account in cui verrà richiesto di cambiare le password.
- Questo comando non cambia le password degli utenti esistenti nell'ambiente target anche se sono diverse nello snapshot di origine.
- `dataManagement=true|false` (facoltativo) clona i record di Data Management dall'ambiente di origine all'ambiente target. Il valore predefinito è `true`, che clona i record di Data Management. Impostare questo valore su `false` se non si desidera clonare i record di Data Management.
- `appAudit=true|false`(facoltativo) clona i record di audit dall'ambiente di origine all'ambiente target. Il valore predefinito è `true`, che clona i valori di audit dell'applicazione. Impostare questo valore su `false` se non si desidera clonare i dati di audit dell'applicazione nell'ambiente target.
- `jobConsole=true|false`(facoltativo) clona i record della console job dall'ambiente di origine all'ambiente target. Il valore predefinito è `true`. Impostare questo valore su `false` se non si desidera clonare i record della console job.
- `storedSnapshotsAndFiles`, (facoltativo) indica se il comando deve clonare il contenuto delle cartelle inbox e outbox e gli snapshot memorizzati. Il valore predefinito è `false`.

 **Note:**

Vengono clonate solo le cartelle di livello superiore della posta in entrata e in uscita, non le sottocartelle. Se è necessario conservare il contenuto delle sottocartelle, eseguirne il backup su un computer locale e quindi caricarle nell'ambiente target.

- `DailyMaintenanceStartTime`(facoltativo) reimposta l'ora di inizio della manutenzione dell'ambiente target clonato su quella dell'ambiente di origine. Il valore predefinito è `true`. Per mantenere l'ora di inizio corrente della manutenzione dell'ambiente target, impostare questo valore su `false`.

Esempi

- Clonazione dell'ambiente, degli utenti con relative assegnazioni di ruoli predefiniti, dei dati di audit, dei record della console job e dei record di Data Management. Impostazione inoltre dell'ora di inizio della manutenzione dell'ambiente target su quella dell'ambiente di origine:

```
epmAutomate cloneEnvironment serviceAdmin Password.epw https://test-cloudpln.pbcs.us1.oraclecloud.com UsersAndPreDefinedRoles=true
```
- Clonazione dell'ambiente, inclusi il contenuto delle cartelle inbox e outbox e gli snapshot memorizzati, ma non gli utenti e le assegnazioni di ruoli predefiniti, i record di Data Management, i dati di audit e i record della console job, senza modificare l'ora di inizio della manutenzione dell'ambiente target:

```
epmAutomate cloneEnvironment serviceAdmin Password.epw https://test-
cloudpln.pbcs.us1.oraclecloud.com DataManagement=false appAudit=false
jobConsole=false storedSnapshotsAndFiles=true
DailyMaintenanceStartTime=false
```

- Clonazione dell'intero ambiente (utenti e assegnazioni di ruoli predefiniti, dati di audit, record della console job, contenuto delle cartelle inbox e outbox, snapshot memorizzati e record di Data Management) utilizzando uno snapshot customizzato. Impostazione inoltre dell'ora di inizio della manutenzione dell'ambiente target su quella dell'ambiente di origine:

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

copyDataByPointOfView

Copia i dati da un POV di origine in un cubo a un POV di destinazione nello stesso o in un altro cubo Enterprise Profitability and Cost Management.

Si applica a

Enterprise Profitability and Cost Management

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmAutomate copyDataByPointOfView SOURCE_POV_NAME TARGET_POV_NAME
copyType=ALL_DATA|INPUT SOURCE_CUBE_NAME TARGET_CUBE_NAME [PARAMETER=VALUE]. I
parametri sono descritti di seguito.
```

- *SOURCE_POV_NAME* indica il nome del POV di origine da cui devono essere copiati i dati.
- *TARGET_POV_NAME* indica il nome di un POV target valido in cui devono essere copiati i dati dell'origine.
- *copyType* identifica i dati da copiare dal POV di origine. I valori validi sono:
 - *ALL_DATA* per copiare nel POV di destinazione tutti i dati di input e tutti i dati calcolati.
 - *INPUT* per copiare nel POV di destinazione tutti i dati di input, inclusi i dati driver.
- *SOURCE_CUBE_NAME* indica il nome del cubo che contiene il POV di origine.
- *TARGET_CUBE_NAME* indica il nome del cubo che contiene il POV target.
- *PARAMETER=VALUE* indica i parametri runtime facoltativi e i relativi valori. Specificare il numero di parametri e coppie di valori richiesto dal processo. I parametri validi e i relativi valori sono indicati di seguito.
 - *povDelimiter* (facoltativo) indica il delimitatore utilizzato nei valori POV. Il valore predefinito è `::` (doppio carattere di due punti). Questo valore deve essere racchiuso tra virgolette. Esempio: `povDelimiter="_"`.
Oltre al delimitatore predefinito, sono supportati solo questi delimitatori: `_` (carattere di sottolineatura), `#` (cancellotto), `&` (E commerciale), `~` (tilde), `%` (percentuale), `;` (punto e virgola), `:` (due punti), `-` (trattino).

- `createDestPOV=true|false` specifica se creare il POV target se non esiste. Il valore predefinito è `false`. È necessario impostare questo valore di parametro su `true` se il POV di destinazione non esiste.

Esempi

- **Copia di tutti i dati in un POV diverso nello stesso cubo:**

```
epmAutomate copyDataByPointOfView FY21_Jan_Actual_Working
FY22_Jan_Actual_Working ALL_DATA PCM_CLC PCM_CLC povDelimiter="_"
createDestPOV=true
```
- **Copia di tutti i dati in un POV diverso in un cubo diverso:**

```
epmAutomate copyDataByPointOfView FY21_Jan_Actual_Working
FY22_Jan_Actual_Working ALL_DATA PCM_CLC PCM_REP povDelimiter="_"
createDestPOV=true
```
- **Copia dei dati di input in un POV diverso nello stesso cubo:**

```
epmAutomate copyDataByPointOfView FY21_Jan_Actual_Working
FY22_Jan_Actual_Working INPUT PCM_CLC PCM_CLC povDelimiter="_"
createDestPOV=true
```
- **Copia dei dati di input in un POV diverso in un cubo diverso:**

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

copyDataByProfile

Copia di dati per gli elementi, ad esempio le aree, definiti in un profilo di cancellazione dati.

Si applica a

Financial Consolidation and Close, Tax Reporting

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate copyDataByProfile PROFILE_NAME` dove `PROFILE_NAME` è il nome di un profilo di copia dati definito in Financial Consolidation and Close e Tax Reporting.

Esempio

```
epmautomate copyDataByProfile copyDataProfile_01
```

copyFileFromInstance

Copia un file da un ambiente di origine all'ambiente dal quale si esegue questo comando.

Prima di eseguire il comando, accedere all'ambiente nel quale si desidera copiare il file utilizzando EPM Automate.

Si applica a

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, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato assegnati al ruolo applicazione Amministratore migrazione

Utilizzo

`epmautomate copyFileFromInstance SOURCE_FILE_NAME USERNAME PASSWORD_FILE URL TARGET_FILE_NAME` dove:

- `SOURCE_FILE_NAME` è il nome del file (inclusa l'estensione) che si desidera copiare dall'ambiente di origine.
- `USERNAME` è il nome utente di un Amministratore servizi dell'ambiente di origine.
- `PASSWORD_FILE` è il nome e la posizione del file in cui è memorizzata la password cifrata dell'amministratore dei servizi dell'ambiente di origine.
- `URL` è l'URL dell'ambiente di origine.
- `TARGET_FILE_NAME` è un nome univoco per il file (inclusa l'estensione) nell'ambiente in cui si esegue questo comando.

Esempio

```
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

Copia nell'ambiente corrente un file o uno snapshot di backup da un gruppo di Oracle Object Storage.

In caso di copia di uno snapshot di backup, questo comando copia lo snapshot dal gruppo di Object Storage ed estrae il relativo contenuto in Oracle Enterprise Performance Management Cloud.

Si applica a

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, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate copyFromObjectStorage USERNAME PASSWORD URL TARGET_FILE_NAME`
dove:

- *USERNAME* indica l'ID di un utente che dispone dei diritti di accesso richiesti in Oracle Object Storage Cloud.
Per gli utenti creati in un provider di identità federata, specificare il nome completamente qualificato dell'utente, ad esempio `exampleIdP/jdoe` o `exampleIdP/john.doe@example.com`, dove `exampleIdP` indica il nome del provider di identità federata. Per gli altri utenti, specificare l'ID utente.
- *PASSWORD* indica la password Swift o il token di autenticazione associato all'utente. Questa password non corrisponde a quella utilizzata per accedere alla console Object Storage. Il token di autenticazione è un token generato da Oracle utilizzato per l'autenticazione con API di terze parti, ad esempio per l'autenticazione con un client Swift. Per istruzioni per la creazione di questo token, fare riferimento alla sezione [Come creare un token di autenticazione](#) nella *Documentazione dell'infrastruttura Oracle Cloud*.
- *URL* è l'URL del gruppo di Oracle Object Storage Cloud, contenente il nome del gruppo e dell'oggetto da copiare.
Formato dell'URL:

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

Componenti dell'URL:

- *region_identifier* è una regione di hosting dell'infrastruttura Oracle Cloud.
- *namespace* è il contenitore di livello superiore di tutti i gruppi e oggetti. Ad ogni tenant dell'infrastruttura Oracle Cloud viene assegnato il nome di uno spazio dei nomi di memorizzazione degli oggetti immutabile e generato dal sistema in fase di creazione dell'account. Il nome dello spazio dei nomi del tenant, ad esempio `axaxnpcrorw5`, è valido in tutte le regioni.
- *bucket_name* è il nome di un contenitore logico dove memorizzare dati e file. I gruppi sono organizzati e gestiti in sezioni. Il nome del gruppo generato dal sistema, ad esempio, `bucket-20210301-1359` indica l'anno, il mese, il giorno e l'ora correnti.
- *object_name* è il nome dello snapshot o del file che si desidera copiare da Oracle Object Storage Cloud. Questo valore deve corrispondere esattamente al nome completo dell'oggetto in Object Storage Cloud. Non utilizzare un'estensione, ad esempio `.zip`, a meno che non sia contenuta nel nome dell'oggetto.

Per ulteriori informazioni, fare riferimento ai seguenti argomenti nella *documentazione dell'infrastruttura Oracle Cloud*

- [Regions and Availability Domains](#)
- [Understanding Object Storage Namespaces](#)
- [Managing Buckets](#)
- *TARGET_FILE_NAME* è un nome univoco del file o dello snapshot nell'ambiente EPM Cloud. Quando si copiano gli snapshot, non specificare l'estensione ZIP in modo da poter utilizzare questo nome file con il comando `importSnapshot`.

I file di dimensioni maggiori di 100 MB vengono memorizzati in Oracle Object Storage in una directory logica insieme al file manifest che ne identifica i segmenti. Specificare il nome della directory logica come *TARGET_FILE_NAME*.

Esempi

In questi esempi, sostituire *URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET* con un URL funzionante in questo formato: `https://`

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

- **Copia di uno snapshot denominato backup_Snapshot_12_05_20.zip dal gruppo di Oracle Object Storage in EPM Cloud e ridenominazione:**

```
epmautomate copyFromObjectStorage oracleidentitycloudservice/jDoe
example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/
backup_Snapshot_12_05_20.zip snapshot_from_osc
```
- **Copia di uno snapshot denominato backup_Snapshot_12_05_20 dal gruppo di Oracle Object Storage in EPM Cloud e ridenominazione:**

```
epmautomate copyFromObjectStorage oracleidentitycloudservice/jDoe
example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/backup_Snapshot_12_05_20
snapshot_from_osc
```
- **Copia di uno snapshot denominato backup_Snapshot_12_05_20 dal gruppo di Oracle Object Storage in EPM Cloud senza ridenominazione:**

```
epmautomate copyFromObjectStorage oracleidentitycloudservice/jDoe
example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/backup_snapshot_12_05_20
backup_snapshot_12_05_20
```
- **Copia di un file in EPM Cloud da un gruppo di Oracle Object Storage:**

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

copyOwnershipDataToNextYear

Copia i dati di proprietà dall'ultimo periodo di un anno al primo periodo dell'anno successivo.

Le impostazioni predefinite e di sostituzione iniziali relative alla proprietà vengono automaticamente riportate da periodo a periodo nell'ambito dello stesso anno, ma non per gli anni successivi. Per riportare le impostazioni più recenti dall'ultimo periodo di un anno al primo periodo dell'anno successivo, copiare le impostazioni di proprietà dall'ultimo periodo dell'anno nel POV al primo periodo dell'anno successivo.

Si applica a

Financial Consolidation and Close e Tax Reporting.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente

Utilizzo

`epmautomate copyOwnershipDataToNextYear Scenario Year dove:`

- Scenario è il nome dello scenario dal quale copiare i dati proprietà.
- Year è l'anno dal quale copiare i dati proprietà nel primo periodo dell'anno successivo.

Esempio

```
epmautomate copyOwnershipDataToNextYear FCCS_total_Actual FY18
```

copyPOV

Copia gli artifact di modello e i dati dei cubi di Oracle Essbase da un POV di origine a un POV di destinazione.

Si applica a

Profitability and Cost Management.

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

```
epmautomate copyPOV APPLICATION_NAME SOURCE_POV_NAME TARGET_POV_NAME  
PARAMETER=VALUE stringDelimiter="DELIMITER" [isInputData=true|false  
isAllInputData=true|false]. I parametri sono descritti di seguito.
```

- *APPLICATION_NAME* è il nome dell'applicazione Profitability and Cost Management contenente il POV di origine.
- *SOURCE_POV_NAME* è il nome del POV di origine nell'applicazione specificata.
- *TARGET_POV_NAME* è il nome di un POV target valido con stato *Draft*
- *PARAMETER=VALUE* indica i parametri runtime e i relativi valori per la copia del POV. Specificare il numero di parametri e coppie di valori richiesto dal processo. I parametri validi e i relativi valori sono indicati di seguito.
 - *isManageRule=true|false* specifica se copiare le regole.
 - *isInputData=true | isAllData=true | isAllInputData=true* (facoltativo) specifica come copiare i dati. Il valore predefinito per questi parametri è *false*. Specificare *true* per uno solo dei parametri elencati di seguito.
 - * Specificare *isInputData=true* per copiare nel POV di destinazione i dati di input.
 - * Specificare *isAllData=true* per copiare nel POV di destinazione tutti i dati di input e tutti i dati calcolati.
 - * Specificare *AllInputData=true* per copiare nel POV di destinazione tutti i dati di input, inclusi i dati driver.
 - *modelViewName=NAME* specifica il nome della sezione dati che deve essere copiata dal POV di origine al POV target.
 - *createDestPOV=true|false* specifica se creare il POV target se non esiste.
 - *nonEmptyTupleEnabled=true|false* specifica se abilitare Tupla non vuota (NET) in modo che il comando tenga esclusivamente conto delle intersezioni che contengono dei dati. L'impostazione predefinita è *true*, che, in alcuni rari casi, può creare problemi nell'esecuzione del comando quando si copiano i dati Essbase. In quei casi, sostituire l'impostazione predefinita utilizzando *nonEmptyTupleEnabled=false* per migliorare le performance.

 **Nota:**

I valori dei parametri (`true` o `false`) devono essere in minuscolo.

- `stringDelimiter="DELIMITER"` specifica il delimitatore utilizzato nei valori POV. Il delimitatore deve essere racchiuso tra virgolette.

Esempi

- `epmautomate copyPOV BksML12 2012_Jan_Actual 2012_Feb_Actual isManageRule=true isInputData=true modelName="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

Copia lo snapshot corrente da un ambiente di origine all'ambiente (target) da cui si esegue questo comando.

Questo comando viene utilizzato principalmente come primo passo per la migrazione di un ambiente mediante la copia dello snapshot corrente da un altro ambiente, ad esempio da un ambiente di test a un ambiente di produzione. Utilizzare il comando [importSnapshot](#) per completare il processo di migrazione.

Prima di eseguire il comando, avviare una sessione di EPM Automate ed eseguire l'accesso all'ambiente target.

Se questo comando viene eseguito per copiare lo snapshot corrente mentre viene generato lo snapshot dell'ambiente di origine, durante la manutenzione giornaliera ad esempio verrà visualizzato l'errore `File not found`.

Si applica a

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, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato assegnati al ruolo applicazione Amministratore migrazione

Utilizzo

`epmautomate copySnapshotFromInstance SNAPSHOT_NAME USERNAME PASSWORD_FILE URL`
dove:

- `SNAPSHOT_NAME` è il nome di uno snapshot esistente nell'ambiente di origine.
- `USERNAME` è il nome utente di un amministratore dei servizi dell'ambiente di origine.

- *PASSWORD_FILE* è il nome e la posizione del file in cui è memorizzata la password cifrata dell'amministratore dei servizi dell'ambiente di origine.
- *URL* è l'URL dell'ambiente di origine.

Esempio

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

copyToObjectStorage

Copia un file o uno snapshot dall'ambiente corrente in un gruppo di Oracle Object Storage Cloud.

Se si copia uno snapshot, questo comando ne comprime il contenuto prima della copia in Oracle Object Storage.

Per facilitare la copia rapida dei file, questo comando suddivide i file di grandi dimensioni (superiori a 100 MB) in segmenti da 10 MB (con nome da *FILE_NAME/FILE_NAME_object_store_bytes_seg_0* a *FILE_NAME/FILE_NAME_object_store_bytes_seg_n*) e crea un file manifest (denominato *FILE_NAME/FILE_NAME.manifest*). I segmenti di file vengono memorizzati in Oracle Object Storage insieme al file manifest. Nella console Object Storage, il file viene visualizzato come directory logica contenente i segmenti di file e il file manifest.

I file di dimensioni inferiori a 100 MB non vengono segmentati e vengono memorizzati con il nome file originale.

Si applica a

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, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate copyToObjectStorage SOURCE_FILE_NAME USERNAME PASSWORD URL
dove:
```

- *SOURCE_FILE_NAME* è il nome del file o dello snapshot in Oracle Enterprise Performance Management Cloud. In caso di copia di uno snapshot, non specificare l'estensione ZIP.
- *USERNAME* indica l'ID di un utente che dispone dei diritti di accesso necessari per scrivere in Oracle Object Storage Cloud. Per gli utenti creati in un provider di identità federata, specificare il nome completamente qualificato dell'utente, ad esempio `exampleIdP/jdoe` o `exampleIdP/john.doe@example.com`, dove `exampleIdP` indica il nome del provider di identità federata. Per gli altri utenti, specificare l'ID utente.

- *PASSWORD* indica la password Swift o il token di autenticazione associato all'utente. Questa password non corrisponde a quella utilizzata per accedere alla console Object Storage. Il token di autenticazione è un token generato da Oracle utilizzato per l'autenticazione con API di terze parti, ad esempio per l'autenticazione con un client Swift. Per istruzioni per la creazione di questo token, fare riferimento alla sezione [Come creare un token di autenticazione](#) nella *Documentazione dell'infrastruttura Oracle Cloud*.
- *URL* è l'URL di Oracle Object Storage Cloud contenente l'ID del gruppo e un nome di oggetto facoltativo.

Formato dell'URL senza il nome oggetto:

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

Formato dell'URL con il nome oggetto:

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

Componenti dell'URL:

- *region_identifier* è una regione di hosting dell'infrastruttura Oracle Cloud.
- *namespace* è il contenitore di livello superiore di tutti i gruppi e oggetti. Ad ogni tenant dell'infrastruttura Oracle Cloud viene assegnato il nome di uno spazio dei nomi di memorizzazione degli oggetti immutabile e generato dal sistema in fase di creazione dell'account. Il nome dello spazio dei nomi del tenant, ad esempio *axaxnpcrow5*, è valido in tutte le regioni.
- *bucket_name* è il nome di un contenitore logico dove memorizzare dati e file. I gruppi sono organizzati e gestiti in sezioni. Il nome del gruppo generato dal sistema, ad esempio, *bucket-20210301-1359* indica l'anno, il mese, il giorno e l'ora correnti.
- *object_name* (facoltativo) è il nome che si desidera utilizzare per il file in Oracle Object Storage Cloud. Se non viene specificato alcun nome di oggetto, il file verrà copiato con il relativo nome originale.

Per ulteriori informazioni, fare riferimento ai seguenti argomenti nella *documentazione dell'infrastruttura Oracle Cloud*

- [Regions and Availability Domains](#)
- [Understanding Object Storage Namespaces](#)
- [Managing Buckets](#)

Esempi

In questi esempi, sostituire *URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET* con un URL funzionante in questo formato: `https://`

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

- Copia e ridenominazione di uno snapshot in un gruppo di Object Storage:

```
epmautomate copyToObjectStorage "Artifact Snapshot"
oracleidentitycloudservice/jDoe example_pwd
URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/Snapshot_04_30_21
```
- Copia di un file in un gruppo di Object Storage:

```
epmautomate copyToObjectStorage example_file.txt oracleidentitycloudservice/
jDoe example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET
```
- Copia e ridenominazione di un file in un gruppo di Object Storage:

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

createGroups

Aggiunge un batch di utenti a un gruppo esistente in Controllo accesso utilizzando un file CSV con codifica ANSI o UTF-8 caricato nell'ambiente.

Utilizzare il comando [uploadFile](#) per caricare file in un ambiente. Il formato del file è il seguente:

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

Per i nomi gruppo non è prevista la distinzione tra maiuscole e minuscole. Quando termina l'esecuzione del comando, EPM Automate invia in stampa alla console le informazioni su ogni singola voce non riuscita. Esaminare queste informazioni per comprendere il motivo per cui l'esecuzione del comando non è riuscita per alcune voci nel file CSV.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi o Gestione controllo accesso

Utilizzo

epmautomate createGroups *FILE_NAME* dove *FILE_NAME* è il nome di un file CSV contenente i nomi e le descrizioni dei gruppi.

Esempio

```
epmautomate createGroups group_file.CSV
```

createNRSnapshot

Creare uno snapshot su richiesta, denominato `EPRCS_Backup.tar.gz`, di un ambiente Narrative Reporting.

È possibile scaricare `EPRCS_Backup.tar.gz` e il file degli errori in un computer locale utilizzando il comando [downloadFile](#) oppure copiare il file in un altro ambiente con il comando [copyFileFromInstance](#).

I dati dell'applicazione in `EPRCS_Backup.tar.gz` sono aggiornati all'ultima manutenzione giornaliera. Per eseguire il backup di dati più recenti, utilizzare la funzione di Narrative Reporting relativa all'esportazione dei dati.

Si applica a

Narrative Reporting

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate createNRSnapshot [errorFile=Error_File.txt] dove errorFile (facoltativo)` indica il nome di un file di testo univoco per la registrazione degli eventuali errori rilevati dal comando.

Esempio

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

createReconciliations

Copia i profili in un periodo specifico.

Si applica a

Account Reconciliation.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

```
epmautomate createreconciliations PERIOD SAVED_FILTER dove:
```

- *PERIOD* è il nome di un periodo
- *SAVED_FILTER* è il nome del filtro pubblico salvato. Se non si specifica un filtro salvato, EPM Automate copia tutti i profili validi

Esempi

- Copiare tutti i profili per il periodo: `epmautomate createReconciliations "January 2015"`
- Copiare i profili di un filtro specifico: `epmautomate createReconciliations "January 2015" "Corporate Recs"`

deleteFile

Elimina un file o uno snapshot dalla posizione di caricamento predefinita, dalla cartella inbox o outbox, da una cartella di Data Management o da profitinbox/profitoutbox.

Per eliminare un file da una posizione diversa dalla posizione di caricamento predefinita, specificare la posizione del file.

Se questo comando viene eseguito per eliminare uno snapshot in fase di generazione o archiviazione, verrà visualizzato uno degli errori descritti di seguito.

- `File not found` se lo snapshot è in fase di generazione
- `Archive process is in progress. Unable to Rename or Delete` se lo snapshot è in fase di archiviazione

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato assegnati al ruolo applicazione
Amministratore migrazione

Utilizzo

```
epmautomate deleteFile FILE_NAME
```

Nota:

È necessario specificare il nome del file compresa l'estensione, ad esempio `data.csv` o `data.zip`, se applicabile. È possibile eliminare uno snapshot senza specificare la relativa estensione file (.ZIP). Questo utilizzo, tuttavia, è obsoleto. Specificare la posizione del file, se diversa da quella predefinita. Per informazioni dettagliate, vedere [Posizioni di file predefinite](#). Le posizioni supportate includono `inbox`, `profitinbox`, `outbox`, `profitoutbox`, `to_be_imported` e `inbox/directory_name`.

Esempi

- Eliminare un file dalla posizione di caricamento predefinita:
`epmautomate deleteFile data.csv`
- Eliminare un file dalla cartella inbox:
`epmautomate deleteFile inbox/data.csv`
- Eliminare dalla cartella outbox:
`epmautomate deleteFile outbox/data.csv`
- Eliminare uno snapshot creato mediante la migrazione:
 - `epmautomate deleteFile "Backup 18-06-12.zip"` oppure
 - `epmautomate deleteFile "Backup 18-06-12"` (obsoleto)
- Eliminare da `profitinbox` (Profitability and Cost Management):
`epmautomate deleteFile profitinbox/data.csv`
- Eliminare da `profitoutbox` (Profitability and Cost Management):
`epmautomate deleteFile profitoutbox/data.csv`
- Eliminare da una cartella di caricamento di Data Management:

```
epmautomate deleteFile inbox/dm_data/data.csv
```

- Eliminare da una cartella di Data Management:
epmautomate deleteFile outbox/dm_data/data.csv

deleteGroups

Rimuove i gruppi da Controllo accesso sulla base delle informazioni disponibili in un file CSV con codifica ANSI o UTF-8 caricato nell'ambiente.

Utilizzare il comando [uploadFile](#) per caricare file in un ambiente. Il formato del file è il seguente:

```
Group Name
Example_grp1
Example_grp2
```

Per i valori di Group Name nel file non è prevista la distinzione tra maiuscole e minuscole. Quando termina l'esecuzione del comando, EPM Automate invia in stampa alla console le informazioni su ogni singola voce non riuscita. Esaminare queste informazioni per comprendere il motivo per cui l'esecuzione del comando non è riuscita per alcune voci nel file CSV.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi o Gestione controllo accesso

Utilizzo

epmautomate deleteGroups *FILE_NAME* dove *FILE_NAME* è il nome di un file CSV contenente i nomi dei gruppi da rimuovere da Controllo accesso.

Esempio

```
epmautomate deleteGroups group_file.CSV
```

deletePointOfView

Elimina gli artifact e i dati dei cubi di Oracle Essbase da un POV in un'applicazione Enterprise Profitability and Cost Management.

Si applica a

Enterprise Profitability and Cost Management

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate deletePointOfView POV_NAME [povDelimiter="DELIMITER"].` I parametri sono descritti di seguito.

- `POV_NAME` identifica il nome del POV da eliminare.
- `povDelimiter` indica il delimitatore utilizzato nei valori POV. Il valore predefinito è `::` (doppio carattere di due punti). Questo valore deve essere racchiuso tra virgolette. Esempio: `povDelimiter="_"`.
Oltre al delimitatore predefinito, sono supportati solo questi delimitatori: `_` (carattere di sottolineatura), `#` (cancellotto), `&` (E commerciale), `~` (tilde), `%` (percentuale), `;` (punto e virgola), `:` (due punti), `-` (trattino).

Esempio

- Eliminazione di un POV che utilizza un delimitatore di POV customizzato
`epmAutomate deletePointOfView FY21_Jan_Actual_Working povDelimiter="_"`
- Eliminazione di un POV che utilizza il delimitatore di POV predefinito
`epmAutomate deletePointOfView FY21::Jan::Actual::Working`

deletePOV

Elimina gli artifact di modello e i dati dei cubi di Oracle Essbase da un POV in Profitability and Cost Management.

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate deletePOV APPLICATION_NAME POV_NAME`
`stringDelimiter="DELIMITER" dove:`

- `APPLICATION_NAME` è il nome dell'applicazione Profitability and Cost Management contenente il POV da eliminare.
- `POV_NAME` è il nome del POV da eliminare. Questo valore è obbligatorio.
- `stringDelimiter="DELIMITER"` specifica il delimitatore utilizzato nei valori POV. Il delimitatore deve essere racchiuso tra virgolette.

Esempio

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

deployCube

Distribuisce o ridistribuisce il cubo di calcolo di un'applicazione Profitability and Cost Management.

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate deployCube APPLICATION_NAME PARAMETER=VALUE comment="comment"`, dove:

- `APPLICATION_NAME` è il nome di un'applicazione Profitability and Cost Management
- `PARAMETER=VALUE` indica i parametri runtime e i relativi valori per la distribuzione del cubo. Specificare il numero di parametri e coppie di valori richiesto dal processo. I parametri validi e i relativi valori sono indicati di seguito.

Nota:

I valori dei parametri (`true` o `false`) devono essere in minuscolo.

- `isKeepData=true|false`
specifica se conservare i dati esistenti, se presenti
- `isReplaceCube=true|false` specifica se sostituire il cubo esistente

Nota:

I valori di `isKeepData` e `isReplaceCube` non possono essere entrambi impostati su `true`.

- `isRunNow=true|false` specifica se eseguire subito il processo
- `comment` è un commento facoltativo racchiuso tra virgolette

Esempio

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

deployEJTemplates

Distribuisce template di giornale Enterprise finalizzati in periodi aperti in Financial Consolidation and Close. La distribuzione di un template di giornale Enterprise comporta la

creazione di giornali ricorrenti associati al template per il periodo selezionato. Consente inoltre di creare giornali ad hoc utilizzando uno o più template distribuiti.

Questo comando è un'alternativa all'utilizzo delle schermate di Financial Consolidation and Close per distribuire un nuovo template di giornale Enterprise all'inizio del mese.

Si applica a

Consolidamento finanziario e chiusura

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate deployEJTemplates YEAR PERIOD [Template=TEMPLATE_NAME] [ResetJournals=true|false]`. I parametri sono descritti di seguito.

- `Year` indica l'anno del giornale.
- `Period` indica il periodo del giornale. Questo valore può essere specificato solo se è specificato l'anno.
- `Template=TEMPLATE_NAME` identifica i nomi dei giornali da distribuire. Per distribuire più giornali, fornire ogni nome di template univoco nel formato `Template=TEMPLATE_NAME`, ad esempio `Template="Loan Details" Template="Housing Details" Template="Repayment Details"`. Se il valore di questo parametro non viene specificato, il comando distribuisce tutti i template per la combinazione di anno e periodo specificata.
- `ResetJournals` (facoltativo) indica se tutti i giornali devono essere reimpostati sul primo stadio dopo la ridistribuzione dei template. Il valore predefinito è `false`. Financial Consolidation and Close convalida questo valore internamente in base alle modifiche apportate ai template e può sostituire il valore specificato, se necessario.

Esempio

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

deployFormTemplates

Distribuisce template form finalizzati in nuovi periodi di raccolta dati per creare form dati supplementari, garantendo un processo di raccolta dati coerente e ripetibile.

Si applica a

Financial Consolidation and Close, Tax Reporting.

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate deployFormTemplates COLLECTION_INTERVAL [DIMENSION] [Template] [resetWorkFlows=true|false] dove:`

- `COLLECTION_INTERVAL` è il nome dell'intervallo di raccolta in cui il template deve essere distribuito.
- `DIMENSION` (facoltativo) specifica le dimensioni della frequenza del processo di raccolta dati nel formato `DIMENSION=MEMBER_NAME`. Specificare il numero di dimensioni definite nell'intervallo di raccolta, per un massimo di quattro comprese le dimensioni anno e periodo, ad esempio "Year=2020" "Period=July" "Product=Oracle EPM" "Consolidation=entity Input". Non viene utilizzato alcun valore predefinito se non viene specificato questo valore di parametro.
- `Template` (facoltativo) identifica i nomi univoci dei template form da distribuire nel formato `Template=TEMPLATE_NAME`. È possibile specificare un numero qualsiasi di nomi univoci, in base alle necessità, utilizzando questo formato. Ad esempio, `Template="Loan Details Template" Template="Housing Details Template" Template="Repayment Details Template"`.
Se il valore di questa proprietà non viene specificato, il comando distribuisce tutti i template per l'intervallo specificato.
- `resetWorkFlows` (facoltativo) indica se tutti i form devono essere reimpostati sul primo stadio dopo la loro distribuzione. Il valore predefinito è `false`.

Esempio

```
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

Distribuisce i task da un template Task Manager in una programmazione task garantendo l'esecuzione coerente di processi aziendali ripetitivi.

Si applica a

Financial Consolidation and Close, Tax Reporting

Ruoli necessari

Amministratore servizi

Utilizzo

`epmAutomate deployTaskManagerTemplate TEMPLATE_NAME SCHEDULE_NAME YEAR PERIOD DAY_ZERO_DATE [dateFormat=DATE_FORMAT] [orgUnit=ORGANIZATION UNIT]`. I parametri sono descritti di seguito.

- `TEMPLATE_NAME` indica il nome del template Task Manager da distribuire.
- `SCHEDULE_NAME` indica il nome programmazione da creare a partire dal template.
- `YEAR` indica il membro dimensione Anno in cui deve essere distribuito il template.
- `PERIOD` indica il membro dimensione Periodo in cui deve essere distribuito il template.
- `DAY_ZERO_DATE` indica la data del giorno zero, in formato valido, da utilizzare per creare la programmazione.
- `dateFormat` (facoltativo) indica il formato della data per il giorno zero. Il formato predefinito è YYYY-MM-DD.

- `orgUnit` (facoltativo) indica il nome dell'unità organizzativa. Se non viene specificato un valore, la programmazione verrà creata utilizzando il mapping delle date standard. Le regole festività non verranno utilizzate.

Esempio

- Distribuzione del template Task Manager per l'unità organizzativa *Ind* utilizzando il formato della data predefinito (YYYY-MM-DD) per la data zero:

```
epmautomate deployTaskManagerTemplate "Vision Monthly Close" "Qtr 2 Close" 2021 July 2021-07-10 orgUnit=Ind
```
- Distribuzione del template Task Manager per l'unità organizzativa *Ind* utilizzando `dd/mm/yyyy` come formato della data zero:

```
epmautomate deployTaskManagerTemplate "Vision Monthly Close" "Qtr 2 Close" 2021 July 02/07/2021 dateFormat=dd/MM/yyyy orgUnit=Ind
```

dismissIPMInsights

Automatizza la chiusura dei dati di Intelligent Performance Management (IPM) Insights prima di eseguire nuovi job IPM Insights. Con la chiusura dei dati vengono chiusi tutti gli approfondimenti aperti su cui non si prevede di eseguire alcuna azione. Questo comando è un'alternativa alla chiusura manuale dei dati mediante il dashboard di IPM Insights.

Si applica a

Planning, Planning Modules, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate dismissIPMInsights [comment="comment"]`, dove `comment` (facoltativo) consente di giustificare la chiusura di approfondimenti aperti.

Esempio

```
epmautomate dismissIPMInsights comment="dismissing unusable insights"
```

downloadFile

Esegue il download di un file di un ambiente nel computer locale.

Utilizzare questo comando per eseguire il download di dati, metadati e snapshot di backup per la memorizzazione locale. Il file viene scaricato nella cartella dalla quale si esegue EPM Automate.

Se questo comando viene eseguito per scaricare lo snapshot corrente mentre viene generato lo snapshot dell'ambiente, durante la manutenzione giornaliera ad esempio verrà visualizzato l'errore `File not found`.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato assegnati al ruolo applicazione Amministratore migrazione

Utilizzo

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

Esempi

- Scaricare uno snapshot di manutenzione: `epmautomate downloadFile "Artifact Snapshot"`
- Scaricare uno snapshot customizzato: `epmautomate downloadFile "mySnapshot.zip"`
- Scaricare uno snapshot di manutenzione di Narrative Reporting: `epmautomate downloadFile "EPRCS_Backup.tar.gz"`
- Scarica un file da una posizione di scaricamento predefinita: `epmautomate downloadFile data.csv`
- Eseguire il download da una cartella di Gestione dati: `epmautomate downloadfile outbox/dm_data/data.csv`
- Scaricare da profitoutbox: `epmautomate downloadFile profitOutbox/data.csv`

enableApp

Abilita un'applicazione.

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

```
epmautomate enableapp APPLICATION_NAME dove APPLICATION_NAME è il nome dell'applicazione Profitability and Cost Management che si desidera abilitare.
```

Esempio

```
epmautomate enableApp BksML12
```

enableQueryTracking

Abilita il tracciamento query nei cubi ASO per avviare l'acquisizione dei pattern di recupero dei dati utente (query).

I pattern di recupero dei dati acquisiti consentono di ottimizzare l'aggregazione dei cubi ASO, che viene avviata utilizzando il comando [executeAggregationProcess](#).

Si applica a

Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Strategic Workforce Planninge Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate enableQueryTracking ASO_CUBE_NAME`, dove `ASO_CUBE_NAME` indica il nome del cubo ASO in cui deve essere attivato il tracciamento query.

Esempio

```
epmautomate enableQueryTracking VISION_ASO
```

encrypt

Utilizza l'algoritmo Advanced Encryption Standard (AES/CBC/PKCS5Padding(128)) per cifrare la password di Oracle Enterprise Performance Management Cloud (o il token di aggiornamento e l'ID client OAuth2.0 per l'accesso agli ambienti OCI - Gen 2) e facoltativamente la password del server proxy Internet utilizzata per l'accesso agli ambienti Oracle Fusion Cloud EPM e ne esegue quindi la memorizzazione in un password file.

La cifratura dei segreti consente agli amministratori servizi di condividere il proprio password file cifrato con gli sviluppatori che scrivono script EPM Automate in modo da consentire loro di eseguire gli script. Ciò evita la necessità di condividere la password dell'amministratore dei servizi o creare un account EPM Cloud generico condiviso in modo specifico per l'esecuzione degli script.

Il processo di cifratura della password viene eseguito una sola volta.



Nota:

Per informazioni sulla cifratura delle password contenenti caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#).

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

Utilizzo

`epmautomate encrypt PASSWORD|REFRESH_TOKEN KEY PASSWORD_FILE [ClientID=CLIENT_ID] [ProxyServerPassword=PROXY_PASSWORD]`. I parametri sono descritti di seguito.

- `PASSWORD|REFRESH_TOKEN PASSWORD` indica la password o il token di aggiornamento OAuth che si desidera cifrare. Non è possibile utilizzare credenziali aziendali con EPM Automate.
- `KEY` è la chiave privata da utilizzare per cifrare la password.
- `PASSWORD_FILE` indica il nome e la posizione del file in cui è memorizzata la password o il token di aggiornamento cifrato. Il file delle password deve utilizzare l'estensione `.epw`.
- `ClientID` (facoltativo) indica l'identificativo client creato durante l'installazione di OAuth 2.0. Questo valore deve essere specificato durante la cifratura di un token di aggiornamento OAuth 2.0. Non specificare questo valore durante la cifratura di una password.
- `ProxyServerPassword` indica la password per l'autenticazione dell'utente con il server proxy HTTP. Obbligatorio solo se l'autenticazione a livello di server proxy è abilitata per la rete in uso.

Esempi

- **Cifrare solo la password di EPM Cloud:** `epmautomate encrypt P@ssword1 myKey C:\mySecuredir\password.epw`
- **Cifrare le password di EPM Cloud e dei server proxy Internet:** `epmautomate encrypt E@xample1 myKey C:\mySecuredir\password.epw ProxyServerPassword=Proxy_Pwd1`
- **Encrypt esegue l'aggiornamento del token e dell'ID del client:** `epmautomate encrypt AAyyilyBAWD4...FVxkxefd8kjoJr6HJPA= myEncyprtion42Key C:\mySecuredir\oauthfile1.epw ClientID=6fdf2e72fd343430ABR22394C`

executeAggregationProcess

Avvia il processo di aggregazione, utilizzando facoltativamente le statistiche di tracciamento query, per migliorare le performance dei cubi ASO. Questo è un passo importante per l'ottimizzazione dei cubi ASO.

Prima di eseguire questo comando, procedere come segue.

- Utilizzare il comando [enableQueryTracking](#) per acquisire le statistiche di recupero dati per l'ottimizzazione dell'aggregazione ASO.
- Lasciare al processo aziendale tempo sufficiente per acquisire i pattern di recupero dei dati utente (query) che possono essere utilizzati per creare viste di aggregazione.

Si applica a

Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Strategic Workforce Planning Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate executeAggregationProcess ASO_CUBE_NAME [useQueryData=true|false] [includeAlternateRollups=disable|enable] [growthSizeRatio=VALUE].` I parametri sono descritti di seguito.

- `useQueryData` consente di utilizzare i dati di query registrati, raccolti utilizzando il tracciamento query, per selezionare il set di viste di aggregazione più appropriato. Il valore predefinito è `false`.
- `includeAlternateRollups` consente di includere gerarchie secondarie (con utilizzo del livello predefinito) nel processo di selezione delle viste. L'impostazione predefinita è `disable`.
- `growthSizeRatio` (facoltativo) indica il rapporto per la crescita massima del cubo per l'aggregazione delle viste selezionate dal server. La crescita del cubo viene arrestata quando viene raggiunto il rapporto specificato per la crescita massima. L'impostazione predefinita consente la crescita del cubo senza alcun limite di rapporto di crescita.



Note:

Per creare viste di aggregazione predefinite, eseguire questo comando senza specificare i parametri facoltativi.

Esempi

- Creazione di un vista di aggregazione basata su dati di query acquisiti utilizzando il comando [enableQueryTracking](#):
`epmautomate executeAggregationProcess VISION_ASO useQueryData=true includeAlternateRollups=enable`
- Creazione di una vista di aggregazione predefinita:
`epmautomate executeAggregationProcess Vis1ASO`

executeBurstDefinition

Esegue una definizione separazione che specifica gli artifact, i POV e altre impostazioni necessarie per eseguire report o registri per più membri di una singola dimensione per un'origine dati.

Si applica a

Narrative Reporting

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore
Agli utenti che dispongono dei ruoli Utente avanzato, Utente e Visualizzatore deve essere assegnata sicurezza supplementare mediante ACL.

Utilizzo

`epmAutomate executeBurstDefinition ARTIFACT_NAME` dove `ARTIFACT_NAME` indica il percorso e il nome della definizione separazione.

Esempio

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

executeReportBurstingDefinition

Utilizzando una definizione di separazione è possibile eseguire la separazione per un singolo report o registro per più di un membro di una singola dimensione, nonché pubblicare un output PDF o di Excel statico (non aggiornabile in Oracle Smart View for Office) per ogni membro.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmAutomate executeReportBurstingDefinition BURST_DEFINITION_NAME  
[jobName=JOB_NAME]. I parametri sono descritti di seguito.
```

- *BURST_DEFINITION_NAME* indica il percorso e il nome di una definizione di separazione.
- *JOB_NAME* (facoltativo) indica il nome del job che deve essere utilizzato per l'esecuzione della definizione di separazione. Il valore predefinito è *Execute Bursting Definition*.

Esempio

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

exportAccessControl

Esporta il report relativo ai dettagli utente, che contiene informazioni sugli utenti che dispongono di ruoli predefiniti nell'ambiente ed elenca in un file di Excel o CSV gli attributi di ciascun utente, ad esempio nome ed e-mail, nonché informazioni sull'accesso, ad esempio l'assegnazione a Gruppi, Team e Organizzazioni.

Esempio di report:

	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 AR Preparers	john.doe@example.com	Administrator		Yes	Yes			
2	Jane Doe	jane.doe@example.com	Available	AP Reviewers AR Reviewers	jane.doe@example.com	Power User		No	No			
4	ats_power_user4	ats_power_user4	Available		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@example.com	Viewer		No	No			

È possibile scaricare il report utilizzando il comando [downloadFile](#).

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi

Utilizzo

`epmAutomate exportAccessControl REPORT_NAME [reportFormat=XLS|CSV].` I parametri sono descritti di seguito.

- `REPORT_NAME` indica il nome del file di esportazione in cui sarà contenuto il report.
- `reportFormat` (facoltativo) indica il formato file. I valori validi sono XLS e CSV (valore predefinito).

Esempio

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

exportAppAudit

Esporta i record di audit dei dati in un file ZIP che può essere scaricato e archiviato in un computer locale. Le informazioni di audit relative a un massimo di 365 giorni sono disponibili nell'ambiente.

Il primo carattere nel file CSV di output è il carattere Byte Order Mark (BOM) `\u00ff` seguito da un identificativo applicazione cifrato racchiuso tra virgolette doppie. L'identificativo applicazione è seguito dall'intestazione del file CSV.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate exportAppAudit EXPORT_FILE_NAME [userNames=USER_NAMES] [nDays=Number_of_Days] [excludeApplicationId=true|false].` I parametri sono descritti di seguito.

- `EXPORT_FILE_NAME` è il nome del file ZIP in cui verranno memorizzati i dati di audit esportati. Utilizzare il comando [downloadFile](#) per scaricare i file da un ambiente.
- `userNames`, facoltativamente, è un elenco di nomi di accesso utente separati da virgole. Se specificato, verranno esportati solo i dati di audit creati da questi utenti. Non specificare questo valore se si desidera esportare i dati di audit per tutti gli utenti.

- `nDays`, facoltativamente, identifica il numero di giorni per cui esportare i record di audit. Il valore predefinito è sette giorni. I valori possibili sono: `all` per esportare i dati di audit disponibili relativi agli ultimi 365 giorni, `1`, `2`, `7`, `30`, `60` e `180`.
- `excludeApplicationId` (facoltativo) indica se l'identificativo applicazione deve essere scritto nel file di esportazione. Il valore predefinito è `false`.

 **Nota:**

I dati di file esportati che non contengono l'identificativo applicazione non possono essere importati negli ambienti Oracle Enterprise Performance Management Cloud.

Esempi

- Esportazione di dati di audit con l'identificativo applicazione:
`epmautomate exportAppAudit auditData userNames=johnDoe,jane.doe@example.com ndays=30`
- Esportazione di dati di audit senza l'identificativo applicazione:
`epmautomate exportAppAudit auditData userNames=johnDoe,jane.doe@example.com ndays=30 excludeApplicationId=true`

exportAppSecurity

Esporta le assegnazioni dell'accesso a livello di artifact (ACL) in un file CSV che può essere scaricato e memorizzato in locale.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate exportAppSecurity EXPORT_FILE_NAME.CSV` dove `EXPORT_FILE_NAME` è il nome del file in cui verranno memorizzati i dati di sicurezza esportati. Il file verrà creato nella cartella in uscita, da cui è possibile scaricarlo nel computer in uso.

Esempio

`epmautomate exportAppSecurity app_security.CSV`

exportARApplicationProperties

Esporta in un file JSON le impostazioni dell'applicazione Account Reconciliation (correlate all'esperienza utente Redwood, al tema, alla notifica e-mail e al nome del processo aziendale), l'immagine di sfondo e l'immagine del logo in modo che sia possibile importarle nello stesso ambiente o in un ambiente diverso.

Questo comando è utile quando si importa un'applicazione da ambienti di produzione ad ambienti di test. Se le impostazioni dell'applicazione sono diverse tra gli ambienti di produzione e di test, è possibile esportarle dall'ambiente di test prima di importare l'applicazione dall'ambiente di produzione e quindi importarle nell'ambiente di test per mantenere le impostazioni originali.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi

Utilizzo

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

- *FILE_NAME* indica il nome del file JSON in cui verranno memorizzati i valori delle proprietà esportate. È possibile scaricare il file di esportazione utilizzando il comando [downloadFile](#). Utilizzare il comando [uploadFile](#) per caricarlo nell'ambiente target e quindi eseguire il comando [importARApplicationProperties](#) per ripristinare queste impostazioni nell'ambiente target.
- *Properties* (facoltativo) indica un elenco separato da virgole contenente le proprietà da esportare. È possibile esportare tutte o alcune delle proprietà elencate di seguito. Se questa proprietà viene omessa, vengono esportate tutte le proprietà elencate di seguito.
 - *Theme*: esporta il tema di visualizzazione utilizzato nell'ambiente.
 - *EmailNotification*: esporta le impostazioni relative alla notifica e-mail definite nell'ambiente.
 - *DisplayBusinessProcessName*: esporta l'impostazione che indica se visualizzare il nome del processo aziendale sulla pagina nell'ambiente.
 - *RedwoodExperience*: esporta l'impostazione dell'ambiente relativa all'esperienza utente Redwood.
 - *BackgroundImage*: esporta l'immagine di sfondo utilizzata nell'ambiente.
 - *LogoImage*: esporta l'immagine del logo utilizzata nell'ambiente.

Esempio

Esportazione solo delle impostazioni relative alla notifica e-mail e all'esperienza utente Redwood, nonché dell'immagine del logo da un ambiente:

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

exportBackgroundImage

Esporta in un file JPG l'immagine di sfondo utilizzata in un ambiente Account Reconciliation in modo che sia possibile importarla in un altro ambiente.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate exportBackgroundImage IMAGE_NAME.jpg`, dove `IMAGE_NAME` indica il nome del file dell'immagine di sfondo.

È possibile scaricare il file di immagine utilizzando il comando [downloadFile](#). Utilizzare il comando [uploadFile](#) per caricarlo nell'ambiente target e quindi eseguire il comando [importBackgroundImage](#) per importarlo.

Esempio

```
epmautomate exportBackgroundImage corpImage.jpg
```

exportCellLevelSecurity

Esporta le impostazioni di sicurezza a livello di cella dal processo aziendale in un file ZIP che può essere scaricato in locale su un computer utilizzando il comando [downloadFile](#).

Si applica a

Planning, Planning Modules, FreeForm, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate exportCellLevelSecurity FILE_NAME.ZIP [names=SECURITY_RECORD_NAMES]
dove:
```

- `FILE_NAME` è il nome del file ZIP che verrà creato per contenere le informazioni di sicurezza a livello di cella.
- `names` (facoltativo) fornisce un elenco, con voci separate da virgole, di definizioni di sicurezza a livello di cella nell'applicazione. Se questa opzione non è disponibile, tutte le definizioni di sicurezza a livello di cella vengono esportate nell'applicazione.

Esempi

- **Esportazione di specifiche definizioni di sicurezza a livello di cella**

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

- **Esportazione di tutte le definizioni di sicurezza a livello di cella**
`epmautomate exportCellLevelSecurity ExportCLSDRecordsFile.zip`

exportConsolidationJournals

Esporta i giornali di consolidamento utilizzando un job definito in Financial Consolidation and Close.

Si applica a

Consolidamento finanziario e chiusura

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate exportConsolidationJournals jobName [fileName=FILE_NAME].|`
parametri sono descritti di seguito.

- `jobName` indica il nome di un job Esportazione giornale creato in Financial Consolidation and Close.
- `fileName` (facoltativo) indica il nome di un file .JLF in cui devono essere esportate le scritture contabili. Utilizzare il comando [downloadFile](#) per scaricare questo file in un computer locale.

Esempio

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

exportData

Esporta i dati dell'applicazione in un file ZIP utilizzando le impostazioni di esportazione dati, ad esempio il nome di file, specificate in un job di tipo `export data`.

Il file di dati esportati viene memorizzato nella posizione di download predefinita da cui è possibile scaricarlo nel computer in uso. Utilizzare la funzionalità Explorer file in entrata/in uscita per visualizzare i dettagli del file esportato.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate exportData JOB_NAME [FILE_NAME] dove JOB_NAME` è il nome di un job definito nell'applicazione e `FILE_NAME` è il nome del file ZIP (facoltativo) nel quale devono essere esportati i dati.

Esempio

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

exportDataManagement

Esporta i record di Data Management da un ambiente a un file ZIP.

Questo comando esporta un set completo di dati di tabelle dati e intermedie, incluse le colonne ID, in un file ZIP in modo che i dati possano essere importati senza perdere l'integrità referenziale.

Il file esportato, ad esempio `dataFile.zip`, viene memorizzato nella cartella `outbox`. Il file esportato può essere scaricato utilizzando il comando [downloadFile](#), ad esempio `epmAutomate downloadFile outbox/dataFile.zip`. È possibile utilizzare questo file ZIP per importare i dati utilizzando il comando [importDataManagement](#).

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate exportDataManagement FILE_NAME.zip` dove `FILE_NAME` indica il nome del file ZIP in cui devono essere esportati i dati.

Esempio

```
epmautomate exportDataManagement dataFile.zip
```

exportDimension

Esporta una dimensione da un'applicazione Oracle Enterprise Data Management Cloud in un file nell'area temporanea oppure, facoltativamente, in un ambiente target definito in una connessione.

Si applica a

Oracle Enterprise Data Management Cloud

Ruoli necessari

Amministratore servizi, Utente (con autorizzazione Gestione dati)

Utilizzo

```
epmautomate exportDimension APPLICATION DIMENSION FILE_NAME [connection=NAME]  
dove:
```

- `APPLICATION` è il nome di un'applicazione Oracle Enterprise Data Management Cloud

- `DIMENSION` è il nome di una dimensione dell'applicazione
- `FILE_NAME` è il nome del file (CSV per l'esportazione in un file oppure ZIP per l'esportazione in Oracle Financials Cloud) per l'archiviazione dei dati esportati. Se il parametro `connection` non è impostato, questo file viene creato nell'area temporanea. È possibile scaricare il file in un computer locale utilizzando il comando [downloadFile](#) oppure copiarlo in un altro ambiente Oracle Enterprise Performance Management Cloud mediante il comando [copyFileFromInstance](#).
- `connection=NAME` (facoltativo) identifica un nome di connessione (posizione dell'istanza) definito in Oracle Enterprise Data Management Cloud. Se specificato, il file di esportazione viene caricato nell'ambiente target (directory inbox per EPM Cloud e posizione di caricamento predefinita per Oracle Financials Cloud).

 **Nota:**

Le credenziali specificate nella definizione della connessione devono disporre dei diritti di accesso in scrittura all'ambiente target.

Esempi

- Esportare nell'area temporanea di Oracle Enterprise Data Management Cloud:
`epmautomate exportDimension USOperations Entity EntityData.CSV`
- Esportare e caricare in Oracle Financials Cloud: `epmautomate exportDimension USOperations Entity EntityData.zip Connection=ora_fusion_gl`
- Esportare e caricare nella directory inbox target di EPM Cloud: `epmautomate exportDimension USOperations Entity EntityData.CSV Connection=EPM_cloud_pln`

exportDimensionMapping

Esporta le regole di mapping di una dimensione Oracle Enterprise Data Management Cloud specifica per una posizione per creare un file di regole di mapping e, facoltativamente, aggiorna il file esportato nella directory inbox di Data Management di un altro ambiente Oracle Enterprise Performance Management Cloud.

Si applica a

Oracle Enterprise Data Management Cloud

Ruoli necessari

Amministratore servizi, Utente (con autorizzazione Gestione dati)

Utilizzo

`epmautomate exportDimensionMapping APPLICATION DIMENSION LOCATION FILE_NAME [connection=NAME] dove:`

- `APPLICATION` è il nome di un'applicazione Oracle Enterprise Data Management Cloud
- `DIMENSION` è il nome di una dimensione dell'applicazione

- `LOCATION` è la posizione specifica per la quale è necessario esportare le regole di mapping.
- `FILE_NAME` è il nome del file CSV per la memorizzazione dei mapping esportati. Questo file viene creato nell'area temporanea, se il parametro `connection` non è impostato. È possibile scaricare il file in locale su un computer mediante il comando [downloadFile](#) oppure utilizzando il comando [copyFileFromInstance](#) per copiare il file in un altro ambiente EPM Cloud.
- `connection=NAME` (facoltativo) identifica un nome di connessione (posizione dell'istanza) definito in Oracle Enterprise Data Management Cloud. Se il parametro viene specificato, EPM Automate carica il file esportato nella posizione di caricamento predefinita dell'ambiente target.

 **Nota:**

Le credenziali specificate nella connessione devono disporre dei diritti di accesso in scrittura all'ambiente target.

Esempi

- **Esportazione in un'area temporanea:** `epmautomate exportDimensionMapping USOperations Entity Loc1 Loc1Mappings.CSV`
- **Esportare e caricare nella directory inbox dell'ambiente EPM Cloud target:** `epmautomate exportDimensionMapping USOperations Entity Loc1 Loc1Mappings.CSV Connection=EPM_cloud_pln`

exportEJournals

Esporta da Financial Consolidation and Close in un file ZIP i dati di Giornali Enterprise pronti per la contabilizzazione. Questo file può quindi essere utilizzato per importare dati di giornale in un sistema ERP.

Dopo l'esportazione di giornali in un file di esportazione, questo comando aggiorna per ogni giornale esportato lo stato di contabilizzazione da `Ready To Post` a `Post In Progress`.

Si applica a

Consolidamento finanziario e chiusura

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate exportEJournals FILE_NAME.zip [year=YEAR [period=PERIOD]].I`
parametri sono descritti di seguito.

- `FILE_NAME` identifica un file ZIP in cui devono essere archiviati i file CSV di esportazione dei giornali. Il comando genera un file CSV (il formato del nome è `YEAR_PERIOD_JOURNALID_YYYYDDMMHHMMSS.csv`) per ogni giornale e quindi comprime tali file per creare questo file ZIP.

- *YEAR* (facoltativo) indica l'anno di raccolta dati per cui devono essere esportati i dati dei giornali. Se non viene specificato, vengono esportati i dati di tutti gli anni.
- *PERIOD* (facoltativo) indica il periodo di raccolta dati a partire dal quale devono essere esportati i dati dei giornali. Può essere impostato solo se è specificato un anno di raccolta dati. Se non viene specificato un valore, vengono esportati i dati di tutti i periodi.

 **Note:**

Se i parametri *YEAR* e *PERIOD* non vengono specificati, questo comando esporta tutti i giornali con stato di contabilizzazione Pronto per contabilizzazione per tutti gli anni e tutti i periodi.

Esempi

- Esportazione dei dati dei giornali per tutti gli anni e tutti i periodi:
`epmautomate exportEJournals Journal_Export.zip`
- Esportazione dei dati dei giornali per un anno specifico:
`epmautomate exportEJournals Journal_Export.zip year=2020`
- Esportazione dei dati dei giornali per una specifica combinazione di anno e periodo:
`epmautomate exportEJournals Journal_Export.zip year=2021 period=March`

exportEssbaseData

Esporta i dati da un cubo dell'applicazione (cubo di Oracle Essbase) in un archivio. È possibile esportare solo i dati di livello 0 (cubi ASO e BSO) o tutti i dati del cubo (cubi BSO).

Utilizzare l'archivio esportato per analizzare i dati di Essbase per individuare i pattern, ad esempio per consentire di migliorare le performance.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

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

- *CUBE_NAME* identifica il cubo da cui devono essere esportati i dati.
- *FILE_NAME* indica il nome di un file ZIP in cui saranno contenuti i dati esportati. È possibile scaricare questo archivio eseguendo il comando [downloadFile](#).
- *level* (facoltativo) identifica il livello di dati da esportare. Il valore predefinito è 0.

- **Cubi ASO:** specificare 0 per esportare i dati di livello 0. Non è possibile utilizzare l'opzione All.
- **Cubi BSO:** specificare 0 per esportare i dati di livello 0 oppure All per esportare tutti i dati.

Esempi

- **Esportazione di tutti i dati da un cubo BSO:**
epmautomate exportEssbaseData Report1 Report1_all_data.zip level=All
- **Esportazione dei dati di livello 0 da un cubo:**
epmautomate exportEssbaseData Plan1 Plan1_lvl0_data.zip

exportJobConsole

Esporta i record della console job in un file CSV e crea un file ZIP di esportazione.

Il primo carattere nel file CSV di output è il carattere Byte Order Mark (BOM) \u00ff seguito da un identificativo applicazione cifrato racchiuso tra virgolette doppie. L'identificativo applicazione è seguito dall'intestazione del file CSV.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

epmautomate exportJobConsole *FILE_NAME*.zip [nDays=*NUMBER_OF_DAYS*] [jobTypes=*JOB_TYPE*] [jobStatusCodes=*STATUS_CODE*] [exportErrorDetails=true|false] [excludeApplicationId=true|false]. I parametri sono descritti di seguito.

- *FILE_NAME* indica il nome del file ZIP in cui verranno memorizzati i record della console job esportati. Utilizzare il comando [downloadFile](#) per scaricare il file da un ambiente.
- nDays (facoltativo) indica il numero di giorni per cui devono essere esportati i record della console job. I valori possibili sono: all (in caratteri tutti minuscoli) per esportare tutti i record della console job disponibili, 1, 2, 7, 30 e 60. L'impostazione predefinita è 7.
- jobTypes (facoltativo) indica un elenco di codici job separati da virgole per cui devono essere esportati i record della console. L'impostazione predefinita è Rules. I valori validi sono:
 - all (in caratteri tutti minuscoli)
 - 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
- JOB_CONSOLE_EXPORT
- SORT_MEMBERS
- SMART_PUSH
- IMPORT_EXCHANGE_RATES
- *jobStatusCodes* (facoltativo) indica un elenco di codici stato job separati da virgole per cui devono essere esportati i record. L'impostazione predefinita è 2 (Completato correttamente). Di seguito sono riportati i valori possibili.
 - all (in caratteri tutti minuscoli) per tutti i job con qualsiasi stato
 - 1 - Elaborazione in corso
 - 2 - Completato correttamente
 - 3 - Non riuscito con errori
 - 4 - Completato con stato sconosciuto
 - 5 - Completato con violazioni di soglia
 - 6 - Annullamento in sospeso
 - 7 - Annullato
 - 8 - Completato con errori
 - 9 - Completato con avvisi
- *exportErrorDetails* (facoltativo) esporta nei file di log i dettagli dei job non riusciti o per i quali è stato segnalato un errore se questo valore è impostato su `true`. Questo file di log degli errori è incluso nel file ZIP di output. L'impostazione

predefinita è `false`. Se questo valore è impostato su `true`, vengono esportati i dettagli di stato dei job con lo stato indicato di seguito.

- Non riuscito con errori
 - Completato con stato sconosciuto
 - Completato con violazioni di soglia
 - Completato con errori
 - Completato con avvisi
- `excludeApplicationId` (facoltativo) indica se l'identificativo applicazione deve essere scritto nel file di esportazione. Il valore predefinito è `false`.

 **Note:**

I dati di file esportati che non contengono l'identificativo applicazione non possono essere importati negli ambienti Oracle Enterprise Performance Management Cloud.

Esempi

- Esportazione dei record della console job disponibili:
`epmautomate exportJobConsole jobs.zip nDays=all jobTypes=all jobStatusCodes=all`
- Esportazione di tutti i record della console job Rules disponibili:
`epmautomate exportJobConsole jobs.zip nDays=all jobStatusCodes=all`
- Esportazione di tutti i record della console job Rules disponibili senza identificativo applicazione:
`epmautomate exportJobConsole jobs.zip nDays=all jobStatusCodes=all excludeApplicationId=true`
- Esportazione solo dei record relativi al job Rules completati correttamente negli ultimi 14 giorni:
`epmautomate exportJobConsole jobs.zip nDays=14`
- Esportazione dei record e degli errori della console relativi a job di importazione metadati e cancellazione cubo non riusciti con errori o completati con errori eseguiti negli ultimi sette giorni:
`epmautomate exportJobConsole jobs.zip jobtypes=IMPORT_METADATA,CLEAR_CUBE jobStatusCodes=3,8 exportErrorDetails=true`

exportLibraryArtifact

Esporta artifact della libreria di Narrative Reporting. Facoltativamente, solo per gli artifact di report, questo comando può convertire l'esportazione in un file LCM che può essere importato in Financial Consolidation and Close, Planning, Planning Modules o Tax Reporting.

Al completamento dell'esportazione utilizzare il comando [downloadFile](#) per scaricare il file di esportazione e il file degli errori in un computer locale.

Si applica a

Narrative Reporting

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore
Agli utenti che dispongono dei ruoli Utente avanzato, Utente e Visualizzatore deve essere concessa sicurezza supplementare mediante ACL.

Utilizzo

`epmautomate exportLibraryArtifact ARTIFACT_PATH EXPORT_FILE`
[exportFormat=Native|File|LCM] [applicationName=APP_NAME]
[errorFile=ERROR_FILE.txt]. I parametri sono descritti di seguito.

- `ARTIFACT_PATH` è la posizione dell'artifact nella libreria di Narrative Reporting.
- `EXPORT_FILE` indica un nome univoco per il file in cui devono essere esportati gli artifact.
- `exportFormat` (facoltativo) può essere impostato su uno dei valori elencati di seguito.
 - `Native` esporta gli artifact come file ZIP che può essere utilizzato con altri ambienti Narrative Reporting. Questo è il valore predefinito.
 - `File` esporta i file nel formato binario originale (PDF, DOCX, ZIP, JPEG e così via) in cui sono disponibili in Narrative Reporting. Questo parametro può essere utilizzato solo per esportare file binari. Non deve essere utilizzato con artifact di report.
 - `LCM` converte i report nel formato utilizzato da Migrazione e li esporta in un file ZIP che può essere importato in ambienti Financial Consolidation and Close, Planning, Planning Modules o Tax Reporting.
- `applicationName` (facoltativo) indica il nome dell'applicazione target in cui si prevede di importare i report. Questo valore è obbligatorio solo se si utilizza `LCM` come valore del parametro `exportFormat`.
- `errorFile` (facoltativo) è il nome univoco del file di testo in cui verranno memorizzati gli errori relativi all'esportazione.

Esempi

- Esportazione di un report nel formato nativo in modo che possa essere importato in un altro ambiente Narrative Reporting:
`epmautomate exportLibraryArtifact "Library/Samples/Sample Report 1" exp_SampleReport1.doc errorFile=export_errors.txt`
- Esportazione di un foglio di calcolo nel formato binario originale:
`epmautomate exportLibraryArtifact "Library/Spreadsheets/Sheet1.xlsx" exp_Sheet1.xlsx exportFormat=File errorFile=export_errors.txt`
- Esportazione e formattazione di report per l'importazione in Financial Consolidation and Close, Planning, Planning Modules o Tax Reporting:
`epmautomate exportLibraryArtifact "Library/Samples/Sample Report 1" exp_SampleReport1.zip exportFormat=LCM applicationName=Vision errorFile=report_exp_errors.txt`

exportLibraryDocument

Esegue l'esportazione in un file di qualsiasi documento disponibile nella libreria dei report.

È possibile scaricare il file esportato in un computer locale utilizzando il comando [downloadFile](#).

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

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

- *ARTIFACT_PATH* è la posizione del documento nella libreria dei report.
- *jobName*, facoltativo, è il nome del job di esportazione artifact della libreria da usare per esportare il documento. Il nome predefinito del job è Copy Artifact From Library.
- *EXPORT_FILE* è un nome univoco per il file nel quale deve essere esportato il documento. Se non si specifica questo valore, il file di esportazione verrà creato utilizzando il nome del documento nella libreria.
- *exportFormat* (facoltativo) può essere impostato su uno dei valori elencati di seguito.
 - *file* esegue l'esportazione del documento nel formato binario originale (PDF, DOCX, Zip, JPEG e via di seguito) in cui è disponibili all'interno della libreria. Questo è il valore predefinito.

Note:

Nell'aggiornamento 24.02 quest'opzione non funziona.

- *zip* esegue l'esportazione di un file ZIP contenente il documento nel formato binario originale. È l'unica opzione che funziona in questo aggiornamento.
- *errorFile* (facoltativo) è un nome univoco del file in cui verranno memorizzati gli errori relativi all'esportazione. Se non si specifica questo valore, non verranno creati file di errore.
- *overwrite*, facoltativo, controlla se nella posizione di download predefinita è presente un file con lo stesso nome da sovrascrivere. Il valore predefinito è *false*, che impedisce il completamento del comando se nella cartella outbox è presente un file con lo stesso nome.

Esempio

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

exportLogoImage

Esporta in un file JPG il logo aziendale utilizzato in un ambiente Account Reconciliation in modo che sia possibile importarlo in un altro ambiente.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi

Utilizzo

epmautomate exportLogoImage *IMAGE_NAME*.jpg, dove *IMAGE_NAME* indica il nome del file dell'immagine del logo.

È possibile utilizzare il comando [downloadFile](#) per scaricare il file del logo esportato. Utilizzare il comando [uploadFile](#) per caricarlo nell'ambiente target e quindi eseguire il comando [importLogoImage](#).

Esempio

```
epmautomate exportLogoImage corpLogo.jpg
```

exportMapping

Esporta le regole di mapping di una dimensione o posizione specifica per creare un file di regole di mapping. È necessario specificare il nome del file e una posizione all'interno della cartella della posta in entrata (ad esempio, `inbox/exportedAccountMap.txt` o `inbox/france sales/exportedAccountMap.txt`) per esportare i mapping.

Utilizzare il comando [downloadFile](#) per eseguire il download del file di mapping esportato in un computer locale.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

```
epmautomate exportMapping DIMENSION_NAME|ALL FILE_NAME LOCATION dove:
```

- *DIMENSION_NAME*|*ALL* è la dimensione di origine da cui devono essere esportati i mapping. Specificare il nome della dimensione da cui i mapping devono essere esportati oppure *ALL* per esportare i mapping da tutte le dimensioni di una posizione.
- *FILE_NAME* è un nome univoco per il file di mapping e una posizione all'interno della cartella outbox.
- *LOCATION* è la posizione di Data Management per la quale è necessario esportare le regole di mapping.

Esempi

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

exportMetadata

Esporta i metadati in un file utilizzando le impostazioni specificate in un job di tipo `export metadata`. Il file contenente i dati esportati viene memorizzato nella posizione di download predefinita da cui è possibile scaricarlo in un computer locale.

Se lo si desidera, è possibile specificare il nome di un file per i dati esportati, che sostituisce il nome file predefinito (nome del job utilizzato per esportare i metadati). I metadati vengono esportati unicamente come file ZIP.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate exportMetadata JOB_NAME [FILE_NAME]` dove *JOB_NAME* è il nome di un job definito nell'applicazione e *FILE_NAME* è il nome del file ZIP nel quale devono essere esportati i metadati.

Utilizzare il comando [downloadFile](#) per scaricare questo file in un server locale.

Esempio

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

exportOwnershipData

Esporta i dati di proprietà da un'entità in un file con valori separati da virgole (CSV).

I dati proprietà predefiniti popolati da Financial Consolidation and Close sono esclusi dal file di esportazione. Nel file di esportazione sono inclusi solo i dati immessi dall'utente per sostituire le impostazioni predefinite.

Si applica a

Financial Consolidation and Close e Tax Reporting.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente

Utilizzo

`epmautomate exportOwnershipData Entity Scenario Year Period FILE_NAME` dove:

- `Entity` è il nome dell'entità dalla quale esportare i dati.
- `Scenario` è lo scenario dal quale esportare i dati.
- `Year` è l'anno dal quale esportare i dati.
- `Period` è il periodo dell'anno dal quale esportare i dati.
- `FILE_NAME` è il nome di un file CSV nel quale esportare i dati. Utilizzare il comando [downloadFile](#) per scaricare questo file in un server locale.

Esempio

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

exportQueryResults

Esegue una query definita in un'applicazione ed esporta i risultati in un file di testo.

Il file dei risultati della query viene memorizzato nella cartella `profitoutbox`. È possibile scaricarlo utilizzando il comando [downloadFile](#) oppure Explorer file in Profitability and Cost Management.

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

Utilizzo

```
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]. I parametri
sono descritti di seguito.
```

- `APPLICATION_NAME` indica il nome dell'applicazione Profitability and Cost Management per cui si desidera eseguire la query.
- `fileName` indica il nome del file in cui verranno memorizzati i risultati della query. Il valore di questo parametro è obbligatorio se non viene specificato il valore del parametro `queryName`. È facoltativo se viene specificato il valore del parametro

`queryName`. In questo caso, verrà utilizzato il nome della query come nome del file dei risultati della query.

Il formato dei dati specificato determina il formato del file di output. Se si utilizza `dataFormat=NATIVE` (valore predefinito), il processo di esportazione crea un file di testo. Se si utilizza `dataFormat=COLUMNAR`, il processo di esportazione crea più file di testo numerati in sequenza e li comprime in un file ZIP.

- `fileOutputOptions` (facoltativo) identifica il formato di output del file dei risultati della query. Il valore predefinito è `ZIP_ONLY`, che crea `fileName.ZIP` o `queryName.ZIP` a seconda che sia specificato un valore per il parametro `fileName`. Altre opzioni sono `TEXT_ONLY` per creare il file di output come file di testo e `ZIP_AND_TEXT` per generare sia un file di testo che un file ZIP.
- `queryName` è un parametro facoltativo che identifica una query definita nell'applicazione. I nomi di query contenenti spazi devono essere racchiusi tra virgolette. Non specificare un nome di query se si desidera esportare tutti i dati Oracle Essbase appartenenti all'applicazione.

Le condizioni seguenti possono far sì che questo comando crei un file di dati vuoto:

- una query con formato errato che non recupera alcun dato;
- una query che genera una quantità eccessiva di dati. In questo scenario valutare l'opportunità di restringere l'ambito della query in modo che recuperi una quantità minore di dati oppure di suddividere la query in query più piccole.
Fare riferimento alla sezione Gestione delle query di *Managing Oracle Profitability and Cost Management Cloud* nella Guida *Amministrazione di Profitability and Cost Management*.
- `exportOnlyLevel0Flg` (facoltativo) specifica se la query deve recuperare solo dati level0. Specificare questo valore di parametro in minuscolo. Questo parametro viene ignorato se si esportano tutti i dati dell'applicazione mediante l'omissione del nome della query.
- `roundingPrecision`, facoltativamente, specifica il numero di posizioni decimali (precisione di arrotondamento) da utilizzare quando i risultati delle query vengono esportati. Applicabile solo quando si specifica `queryName`. Il valore predefinito è 2.
- `dataFormat` (facoltativo) identifica il formato di output. I valori validi sono:
 - `NATIVE`, che mantiene il risultato della query come dati nel formato nativo Essbase. Questo è il valore predefinito.
 - `COLUMNAR`, che converte i dati con formato nativo Essbase e li ordina in colonne per semplificarne l'interpretazione e l'importazione in altre applicazioni. Questa opzione esporta tutti i dati Essbase e ignora il valore del parametro `queryName`. È possibile filtrare i dati impostando il valore del parametro `memberFilters`.

Nota:

Il comando tiene conto dei parametri facoltativi riportati di seguito solo se `dataFormat` è specificato come `COLUMNAR`.

- `memberFilters` (facoltativo) accetta una stringa con formattazione JSON per applicare un filtro in base ai membri dimensione e level0. Esempio, `{"\"Dim1\": [\"Mem1\"], \"Dim2\": [\"Mem21\", \"Mem22\"]}"`

- `includeHeader` (facoltativo) aggiunge nomi di dimensione come intestazioni di colonna. Impostare questo valore su `false` per escludere l'intestazione di colonna. Il valore predefinito è `true`.
- `delimiter` (facoltativo) identifica il delimitatore da utilizzare per separare i membri dimensione nel file dei risultati della query. Il delimitatore deve essere racchiuso tra virgolette. Il valore predefinito è lo spazio (" ").
- `keepDuplicateMemberFormat` (facoltativo) specifica se stampare il formato del membro nel formato del membro duplicato Essbase, ad esempio `[Account]@[Account]`. Impostare questo valore su `false` per stampare solo il nome del membro. Il valore predefinito è `true`.

Esempi

- **Esportare tutti i dati dell'applicazione:**

```
epmautomate exportQueryResults BksML12 fileName="BksML12_MyQuery1.txt"
fileOutputOptions=TEXT_ONLY
```
- **Esportare i risultati di una query specifica:**

```
epmautomate exportQueryResults BksML12 queryName="My Product Query"
roundingPrecision=3
```
- **Esportare i dati Level0 nel formato dati NATIVE:**

```
epmautomate exportQueryResults BksML30 fileName="BksML30_ExportLevel0-
Data" fileOutputOptions=ZIP_AND_TEXT exportOnlyLevel0Flg=true
```
- **Esportare i dati Level0 nel formato dati COLUMNAR con filtro per dimensione singola e membro singolo:**

```
epmautomate exportQueryResults BksML30 fileName="BksML30_Level0-Data"
dataFormat="COLUMNAR" memberFilters="{\"Period\": [\"December\"]}"
includeHeader="true" delimiter="," roundingPrecision="3"
```
- **Esportare i dati Level0 nel formato dati COLUMNAR con filtri per dimensione singola e più membri:**

```
epmautomate exportQueryResults BksML30 fileName="BksML30_Level0-Data"
dataFormat="COLUMNAR" memberFilters="{\"Period\":
[\"November\", \"December\"]}" includeHeader="true" delimiter=","
roundingPrecision="3"
```
- **Esportare i dati Level0 nel formato dati COLUMNAR con filtri per più dimensioni e più membri:**

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

exportSnapshot

Ripete un'operazione di esportazione eseguita in precedenza per creare uno snapshot del contenuto di Migrazione.

Mediante la migrazione,, è possibile selezionare ed esportare in uno snapshot, ad esempio `January16FullApp`, gli artifact desiderati. Utilizzare il nome dello snapshot con questo comando per ripetere successivamente l'operazione di esportazione, esportando solo gli artifact selezionati durante l'operazione di esportazione originale. Fare riferimento alla sezione Esportazione degli artifact e dell'applicazione nella Guida

Administering Migration for Oracle Enterprise Performance Management Cloud (in lingua inglese).

- Quanto riportato di seguito non fa parte degli snapshot delle applicazioni Planning, Planning Modules e FreeForm.
 - Dati di audit
 - Dati della console job

Utilizzare il comando [cloneEnvironment](#) o la funzione Clona ambiente se si desidera copiare i dati di audit e della console job nell'ambiente target.

- Gli snapshot non contengono i dati delle tabelle intermedie di Data Management. Per importare questi dati, utilizzare i comandi [exportDataManagement](#) e [importDataManagement](#) o l'interfaccia degli script di manutenzione del sistema di Data Management. È possibile utilizzare il comando [cloneEnvironment](#) o la funzione Clona ambiente per creare una copia identica dell'ambiente, inclusi i dati delle tabelle intermedie di Data Management.

È possibile eseguire il download dello snapshot esportato dalla posizione predefinita utilizzando il comando [downloadFile](#).

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato assegnati al ruolo applicazione Amministratore migrazione

Utilizzo

`epmautomate exportSnapshot SNAPSHOT_NAME dove SNAPSHOT_NAME è il nome di uno snapshot esistente in Migrazione. Questo snapshot viene sostituito dal nuovo snapshot.`

Esempio

```
epmautomate exportSnapshot January16FullApp
```

exportTemplate

Esporta un'applicazione come template in un file .ZIP. Il file esportato viene memorizzato nella cartella profitoutbox.

È possibile scaricare il file esportato in un computer locale utilizzando il comando [downloadFile](#).

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate exportTemplate APPLICATION_NAME File_Name` dove:

- `APPLICATION_NAME` è il nome dell'applicazione Profitability and Cost Management che si desidera esportare come template.
- `File_Name` è il nome del file template

Esempio

`epmautomate exportTemplate BksML12 template1`

exportTaskManagerAccessControl

Esporta il report relativo ai dettagli utente per le assegnazioni utente Task Manager, Dati supplementari e Giornale Enterprise in Financial Consolidation and Close e Tax Reporting. Il report contiene informazioni sugli utenti che dispongono di ruoli predefiniti nell'ambiente ed elenca in un file di Excel o CSV gli attributi di ciascun utente, ad esempio nome ed e-mail, nonché lo stato, i team, i ruoli predefiniti, i ruoli flusso di lavoro, le organizzazioni, i gruppi e gli indicatori orari dell'ultimo accesso.

Esempio di report sul controllo accesso 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	Organizati	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@example.com	Viewer									Viewer

Si applica a

Financial Consolidation and Close e Tax Reporting

Ruoli necessari

Amministratore servizi

Utilizzo

`epmAutomate exportTaskManagerAccessControl REPORT_NAME` dove `REPORT_NAME` è il nome del file di esportazione, comprensivo di un'estensione CSV o XLS valida, che conterrà il report.

È possibile generare il report in formato CSV o XSL. È possibile scaricare il file utilizzando il comando [downloadFile](#).

Esempi

- `epmAutomate exportTaskManagerAccessControl ac1report.csv`
- `epmAutomate exportTaskManagerAccessControl ac1report.xls`

exportValidIntersections

Esporta i gruppi di interazioni valide dal processo aziendale in un file ZIP che può essere scaricato in locale su un computer utilizzando il comando [downloadFile](#). Le intersezioni valide sono interazioni di celle filtrate in base a regole definite dall'utente, chiamate regole di intersezione valida. Queste regole filtrano determinate intersezioni delle celle per gli utenti quando immettono i dati o selezionano prompt runtime.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate exportValidIntersections FILE_NAME.zip [names=INTERSECTION_NAMES]
dove:
```

- *FILE_NAME* è il nome del file ZIP di esportazione. Tutte le intersezioni valide identificate nel comando vengono esportate in un file Microsoft Excel e compresse in formato ZIP per creare questo file.
- *names* (facoltativo) fornisce un elenco, con voci separate da virgole, delle intersezioni valide che si desidera esportare. Se questo parametro non è specificato, EPM Automate esporta tutte le intersezioni valide nell'applicazione.

Esempi

- **Esportare specifiche intersezioni valide**

```
epmautomate exportValidIntersections VI_export_File.zip
names=VIAccountPeriod,VIEntityPeriod,VIProductPeriod
```
- **Esportare tutte le intersezioni valide**

```
epmautomate exportValidIntersections VI_export_File.zip
```

extractDimension

Estrae una dimensione Oracle Enterprise Data Management Cloud in un file oppure in una connessione globale.

Si applica a

Oracle Enterprise Data Management Cloud

Ruoli necessari

Amministratore servizi, Utente (con autorizzazione Gestione dati)

Utilizzo

```
epmautomate extractDimension APPLICATION DIMENSION EXTRACT_PROFILE FILE_NAME
[connection=NAME] dove:
```

- `APPLICATION` è il nome di un'applicazione Oracle Enterprise Data Management Cloud.
- `DIMENSION` è il nome della dimensione da estrarre.
- `EXTRACT_PROFILE` è il nome di un profilo di estrazione definito nell'applicazione. Questo profilo viene utilizzato per estrarre la dimensione.
- `FILE_NAME` è il nome del file (CSV per l'esportazione in un file oppure ZIP per l'esportazione in Oracle Financials Cloud) per l'archiviazione dei dati esportati. Se il valore del parametro di connessione non è impostato, questo file viene creato nell'area temporanea. È possibile scaricare il file in un computer locale utilizzando il comando [downloadFile](#) oppure copiarlo in un altro ambiente Oracle Enterprise Data Management Cloud mediante il comando [copyFileFromInstance](#) .
- `connection=NAME` (facoltativo) identifica un nome di connessione (posizione dell'istanza) definito in Oracle Enterprise Data Management Cloud come posizione del file di importazione. Se specificato, il file di estrazione viene caricato nell'ambiente target (posta in entrata per Oracle Enterprise Performance Management Cloud e account documento specificato per Oracle ERP).

 **Note:**

Le credenziali specificate nella connessione globale devono disporre dei diritti di accesso in scrittura per l'ambiente target.

Esempi

- Eseguire l'estrazione nell'area temporanea di Oracle Enterprise Data Management Cloud: `epmautomate extractDimension USOperations Entity EntityExtProfile EntityData.CSV`
- Eseguire l'estrazione e il caricamento in Oracle ERP: `epmautomate extractDimension USOperations Entity EntityExtProfile EntityData.zip Connection=ora_fusion_gl`
- Eseguire l'estrazione e il caricamento nella posta in entrata target EPM Cloud: `epmautomate extractDimension USOperations Entity EntityExtProfile EntityData.CSV Connection=EPM_cloud_pln`

feedback

Invia il feedback a Oracle e agli amministratori dei servizi dell'ambiente e carica automaticamente tutti i file di log di EPM Automate creati nelle ultime 24 ore dalla directory corrente.

Facoltativamente, è possibile caricare file aggiuntivi, ad esempio file di tracce di Fiddler, che si desidera vengano utilizzati dal Supporto Oracle per diagnosticare le cause dei problemi riscontrati.

Questo comando, che simula la funzionalità che consente di fornire un feedback del servizio, è molto utile per fornire a Oracle un feedback qualora l'interfaccia utente non risponda o si verifichi un problema durante l'esecuzione di EPM Automate.

Per informazioni sulla funzionalità Feedback, fare riferimento alla sezione Raccolta delle informazioni diagnostiche mediante la utility Feedback in *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori*.

Questo comando restituisce un messaggio simile a quello riportato di seguito in cui si informa che il comando feedback non determina la creazione di una richiesta di servizio. Se si necessita dell'assistenza di Oracle per risolvere un problema, presentare una richiesta di servizio. Il comando visualizza un numero di riferimento UDR che deve essere incluso nella richiesta di servizio che viene presentata.

```
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>
```

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

Utilizzo

`epmautomate feedback "Comment" [Screenshot="FILE_PATH"] [File="FILE_PATH"]` dove:

- `Comment` è il testo che descrive il problema oggetto del feedback inviato. I commenti devono essere racchiusi tra virgolette.
- `Screenshot`, facoltativamente, identifica il nome del file grafico che illustra il problema per il quale è stato inviato il feedback. Per sottoporre più screenshot, ripetere questo parametro e il relativo valore in base alle necessità.
- `File` (facoltativo) identifica il nome di un file che si desidera venga utilizzato dal Supporto Oracle per risolvere il problema corrente. Utilizzare questo parametro per sottoporre file di tracce di Fiddler o altri file a Oracle. Per sottoporre più file, ripetere questo parametro e il relativo valore in base alle necessità.

Esempi

- **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

Controlla se l'applicazione è in modalità amministrazione con accesso limitato solo agli amministratori dei servizi.

Questo comando, che restituisce `true` se l'applicazione è in modalità amministrazione e `false` in caso contrario, è utile per verificare lo stato dell'applicazione prima dell'esecuzione di script di automazione. Il comando [refreshCube](#) ad esempio richiede che l'applicazione sia

in modalità amministrazione. È possibile utilizzare questo comando nello script di automazione come indicato di seguito per controllare se l'applicazione è in modalità amministrazione.

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

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Account Reconciliation, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate getApplicationAdminMode
```

Esempio

```
epmautomate getApplicationAdminMode
```

getDailyMaintenanceStartTime

Visualizza nella console l'ora in formato UTC (Coordinated Universal Time, Tempo coordinato universale) o facoltativamente il fuso orario in cui è programmato l'inizio della manutenzione giornaliera dell'ambiente.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate getDailyMaintenanceStartTime [timezone=true|false] dove
timezone=true (facoltativo) indica se visualizzare l'ora di inizio della manutenzione
giornaliera nel fuso orario specificato durante l'impostazione, ad esempio America/
Los_Angeles. L'impostazione predefinita è false.
```

Esempi

- Visualizzazione dell'ora della manutenzione nel fuso orario specificato al momento dell'impostazione:
epmautomate getDailyMaintenanceStartTime timezone=true
- Visualizzazione dell'ora della manutenzione in formato UTC:

```
epmautomate getDailyMaintenanceStartTime
```

getEssbaseQryGovExecTime

Visualizza il tempo massimo corrente, in secondi, che una query Oracle Essbase può utilizzare per recuperare e fornire informazioni prima che venga terminata.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Profitability and Cost Management, Enterprise Profitability and Cost Management, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate getEssbaseQryGovExecTime
```

Esempio di output del comando:

```
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
```

Esempio

```
epmautomate getEssbaseQryGovExecTime
```

getIdleSessionTimeout

Visualizza il timeout di sessione (in minuti) dell'ambiente Oracle Enterprise Performance Management Cloud. Dopo che una sessione è rimasta inattiva per questo intervallo di tempo, gli utenti vengono reindirizzati alla pagina di accesso.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmAutomate getIdleSessionTimeout
```

Esempio di output del comando:

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

getIPAllowlist

Per gli ambienti OCI (Gen 2), visualizza gli indirizzi IP e i CIDR (Classless Inter-Domain Routing) inclusi nella lista di inclusione corrente.

Questo comando è utile per controllare se uno specifico indirizzo IP o CIDR è attualmente autorizzato ad accedere a un ambiente OCI (Gen 2).

Note:

Non è possibile utilizzare questo comando per elencare gli indirizzi IP e i CIDR negli ambienti Classic. Per questi ambienti, utilizzare la schermata Dettagli servizio di My Services (Classic) per utilizzare le regole della lista di inclusione o di esclusione.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmAutomate getIPAllowlist
```

 **Note:**

Per scrivere in un file tutti gli indirizzi IP e i CIDR esistenti, reindirizzare l'output in un file di testo, che può essere modificato (per rimuovere alcuni o tutti gli indirizzi IP e i CIDR), effettuare il caricamento nell'ambiente, quindi utilizzare il comando `setIPAllowlist` per rimuovere le voci del file dall'elenco di inclusione. Esempio di esecuzione del comando:

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

Esempio

Visualizzazione degli indirizzi IP e dei CIDR inclusi nella lista di inclusione corrente:

```
epmAutomate getIPAllowlist
```

getSubstVar

Recupera i valori delle variabili di sostituzione e li visualizza sullo schermo.

Il formato di visualizzazione è `CUBE_NAME.SUBSTVAR=value`, ad esempio `Plan2.CurYear=2016`. I valori delle variabili di sostituzione a livello di applicazione vengono visualizzati nel formato `ALL.SUBSTVAR=value`, ad esempio `ALL.CurYear=2016`

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato (con accesso a esecuzione regola)

Utilizzo

`epmautomate getSubstVar CUBE_NAME|ALL [name=VARIABLE_NAME] dove:`

- `CUBE_NAME` è il cubo (ad esempio `Plan1`, `Plan2`) da cui si desidera recuperare la variabile di sostituzione. Utilizzare `ALL` per recuperare le variabili di sostituzione a livello di applicazione.
- `name=VARIABLE_NAME` (facoltativo) identifica la variabile di sostituzione per la quale si desidera recuperare il valore. Se non si specifica un nome di variabile, il comando recupera il valore di tutte le variabili di sostituzione.

Esempi

- Recuperare il valore di tutte le variabili di sostituzione a livello di applicazione e cubo:
`epmautomate getSubstVar ALL`

- Recuperare il valore di una variabile di sostituzione specifica a livello di applicazione: `epmautomate getSubstVar ALL name=CurYear`
- Recuperare il valore di tutte le variabili di sostituzione a livello di cubo: `epmautomate getSubstVar Plan2`
- Recuperare il valore di una variabile di sostituzione specifica a livello di cubo: `epmautomate getSubstVar Plan2 name=CurYear`

getVirusScanOnFileUploads

Controlla se l'ambiente OCI (Gen 2) è abilitato alla scansione di tutti i file caricati per garantire che siano privi di virus.

Questo comando controlla se la scansione antivirus viene applicata prima che i file vengano caricati nell'ambiente.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate getVirusScanOnFileUploads`

Questo comando restituisce `true` se l'ambiente è abilitato alla scansione antivirus dei file caricati, in caso contrario `false`.

groupAssignmentAuditReport

Crea un report in cui sono elencati utenti e gruppi aggiunti o rimossi dai gruppi di Controllo accesso per uno specifico intervallo di date.

Questo report, che viene generato come file CSV, può essere utilizzato per supportare le operazioni di audit della sicurezza. Ogni riga del file CSV generato specifica l'utente o il gruppo aggiunto o rimosso, il gruppo a cui l'utente o il gruppo è stato aggiunto o da cui è stato rimosso, l'amministratore servizi che ha eseguito l'operazione e la data e l'ora in cui è stata completata l'operazione. In questo report non sono contenute informazioni di audit su quando i gruppi sono stati aggiunti o eliminati da Controllo accesso.

	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

Si applica a

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, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmAutomate groupAssignmentAuditReport FROM_DATE TO_DATE REPORT_NAME`. I parametri sono descritti di seguito.

- `FROM_DATE` indica la data di inizio del periodo per cui deve essere generato il report, in formato YYYY-MM-DD.
- `TO_DATE` indica la data di fine del periodo per cui deve essere generato il report, in formato YYYY-MM-DD.
- `REPORT_NAME` indica il nome di un file CSV per il report. È possibile scaricare il report generato utilizzando il comando [downloadFile](#).

Esempio

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

help

Visualizza la Guida relativa a tutti i comandi di EPM Automate.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

Utilizzo

```
epmAutomate help
```

Esempio

```
epmAutomate help
```

importAppAudit

Importa i record di audit dati da un file ZIP creato esportando i dati di audit da un ambiente.

Per creare il file di importazione, si utilizza il comando `exportAppAudit` (`epmautomate exportAppAudit auditData ndays=All`). Utilizzare questo comando per clonare i record di audit da un ambiente a un altro durante la migrazione o la clonazione per il processo di disaster recovery.

Si applica a

Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Strategic Workforce Planninge Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate importAppAudit FILE_NAME [logFilename=LOG_FILE_NAME]`. I parametri sono descritti di seguito.

- `FILE_NAME` indica il nome di un file ZIP contenente i record di audit dati che si desidera importare nell'applicazione. Prima di eseguire questo comando, utilizzare il comando `uploadFile` per caricare il file nell'ambiente.
- `logFileName` (facoltativo) identifica il file di log degli errori in cui verranno registrati gli errori rilevati durante l'importazione. Se questo valore non viene specificato, il comando genera un file di errori che viene denominato utilizzando questa convenzione: `username_date_timestamp`. È possibile scaricare questo file utilizzando il comando `downloadFile`.

Esempio

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

importAppSecurity

Carica le autorizzazioni di accesso per utenti o gruppi di un'applicazione da un file CSV disponibile nella cartella in entrata.

Con l'importazione delle autorizzazioni di accesso vengono sovrascritte le assegnazioni esistenti solo per membri importati, form dati, cartelle di form dati, elenchi di task, regole business di Calculation Manager e cartelle di regole business di Calculation Manager. Tutte le altre autorizzazioni di accesso esistenti rimangono inalterate.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate importAppSecurity ACL_FILE_NAME ERROR_FILE [clearall=true|false] dove:
```

- `ACL_FILE_NAME` è il nome di un file CSV contenente le autorizzazioni di accesso che si desidera importare nell'applicazione. Prima di eseguire questo comando, utilizzare il comando [uploadFile](#) per aggiungere il file nella cartella inbox. Il contenuto di un file di input di esempio può essere simile a quello mostrato nell'immagine seguente:

	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

Per una descrizione delle intestazioni di colonna e i possibili valori, fare riferimento alla sezione Importazione della sicurezza in *API REST per Oracle Enterprise Performance Management Cloud*.

- `ERROR_FILE` è il nome di un file CSV che verrà creato da EPM Automate nella cartella in uscita per registrare gli errori rilevati durante questa operazione. È possibile scaricare questo file in un computer locale per analizzare e correggere gli errori segnalati. Il contenuto di un file di errore di esempio può essere simile a quello mostrato nell'immagine seguente. Le colonne del file corrispondono alle colonne di intestazione del file di input:

	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`, specifica facoltativamente se eliminare le autorizzazioni di accesso esistenti prima di caricare nuove autorizzazioni dal file. Il valore predefinito è `false`.

Esempio

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

importARApplicationProperties

Importa in un ambiente Account Reconciliation le impostazioni dell'applicazione (esperienza utente Redwood, tema, notifica e-mail e nome del processo aziendale) e le immagini di sfondo e del logo disponibili in un file JSON di esportazione.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate importARApplicationProperties FILE_NAME`, dove `FILE_NAME` indica il nome del file JSON esportato a partire da un ambiente.

Questo file, esportato a partire da un altro ambiente utilizzando il comando [exportARApplicationProperties](#), deve essere disponibile nell'ambiente in cui si stanno ripristinando le impostazioni dell'applicazione.

Esempio

```
epmautomate importARApplicationProperties myProp.JSON
```

importBackgroundImage

Importa in un ambiente Account Reconciliation l'immagine di sfondo da un file di esportazione.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi

Utilizzo

epmautomate importBackgroundImage *FILE_NAME*.jpg, dove *FILE_NAME* indica il nome del file dell'immagine di sfondo esportato da un altro ambiente.

Esempio

```
epmautomate importBackgroundImage image_file.jpg
```

importBalances

Utilizza Gestione dati per impostare i dati dei saldi da una definizione di caricamento dati.

Si applica a

Account Reconciliation.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore
Agli utenti che dispongono dei ruoli Utente avanzato, Utente e Visualizzatore deve essere concessa sicurezza supplementare mediante ACL.

Utilizzo

```
epmautomate importBalances DL_DEFINITION PERIOD dove:
```

- *DL_DEFINITION* è una definizione di caricamento dati esistente in Account Reconciliation.
- *PERIOD* è il nome di un periodo.

Esempio

```
epmautomate importBalances DailyLoad "January 2020"
```

importCellLevelSecurity

Importa le impostazioni della sicurezza a livello di cella da un file ZIP contenente un file Excel con i record di sicurezza a livello di cella nel processo aziendale. Prima di eseguire questo comando, utilizzare il comando [uploadFile](#) per caricare il file di importazione nell'ambiente.

Il file ZIP di importazione deve contenere un file Excel con due fogli di calcolo (Regole e Regole secondarie) per importare la sicurezza a livello di cella. Il foglio Regole contiene le definizioni della sicurezza a livello di cella incluse le dimensioni, le proprietà come "necessarie altre dimensioni e valide non specificate. Il foglio Regole secondarie contiene le esclusioni e le selezioni membro. Il metodo migliore per ottenere il template del formato del file di importazione è esportare la sicurezza a livello di cella dall'applicazione. Un formato campione è mostrato nelle seguenti immagini.

	A	B	C	D	E	F	G	H	I	J	K
	Name	Position	Description	Enabled	Valid Cubes	Anchor Dim Name	Anchor Dimension Apply to Selected	Dim1 Required	Dim1	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											

	A	B	C	D	E	F	G	H	I	J	K	L
	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												

Si applica a

Planning, Planning Modules, FreeForm, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate importCellLevelSecurity FILE_NAME.ZIP [ErrorFile=FILE_NAME.txt]
```

dove:

- FILE_NAME è il nome del file ZIP contenente il file Excel con le informazioni sulla sicurezza a livello di cella.

- `ErrorFile` (facoltativo) identifica il nome del file di testo in cui verranno scritti i record di errore. Se questo parametro non è specificato, EPM Automate genera automaticamente un file degli errori il cui nome può essere visualizzato nella console dei job.
Utilizzare il comando [downloadFile](#) per scaricare il file degli errori in un computer locale.

Esempio

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

importConsolidationJournals

Importa i giornali di consolidamento da un file .JLF in Financial Consolidation and Close.

- Utilizzare il comando [exportConsolidationJournals](#) per creare il file .JLF utilizzato come input per il comando.
- Prima di eseguire questo comando, utilizzare il comando [uploadFile](#) per caricare il file di input nell'ambiente.

Si applica a

Consolidamento finanziario e chiusura

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate importConsolidationJournals jobName [fileName=FILE_NAME]
[errorFileName=ERROR_FILE_NAME]. I parametri sono descritti di seguito.
```

- `jobName` indica il nome di un job di importazione giornale creato in Financial Consolidation and Close.
- `fileName` (facoltativo) indica il nome di un file .JLF da cui devono essere importate le scritture contabili.
- `errorFileName` (facoltativo) indica il nome del file di log in cui devono essere registrati i messaggi generati durante il processo di importazione.

Esempio

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

importData

Importa i dati da un file nell'applicazione utilizzando le impostazioni di importazione dati specificate in un job di tipo `import data`.

Utilizzare il comando [uploadFile](#) per caricare il file contenente i dati dell'applicazione nella posizione di caricamento predefinita.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate importData JOB_NAME [FILE_NAME] errorFile=ERROR_FILE.zip dove:
```

- *JOB_NAME* è il nome di un job definito nell'applicazione.
- *FILE_NAME* (facoltativo) identifica il nome del file ZIP, CSV o TXT (file di dati in formato Essbase) dal quale devono essere importati i dati. Se si specifica il nome di un file, il nome del file di importazione indicato nel job viene ignorato. Se nella definizione del job è specificato che i dati vengano importati nel formato Essbase, il file ZIP deve contenere un file TXT in formato Essbase. Per altri job di importazione, il file ZIP può contenere uno o più file CSV che identificano la sequenza di importazione nei nomi dei file, ad esempio data1-3.csv, data2-3.csv e data3-3.csv.
- *errorFile*, facoltativamente, identifica il nome di un file ZIP in cui verranno registrati, se presenti, i record rifiutati durante le operazioni di importazione. I file ZIP con lo stesso nome eventualmente presenti nella cartella outbox verranno sovrascritti. È possibile scaricare questo file utilizzando il comando [downloadFile](#).

Esempio

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

importDataManagement

Importa i record di Data Management da un file ZIP a un ambiente.

Questo comando importa i dati in tabelle di impostazione e intermedie da un file ZIP creato utilizzando il comando [exportDataManagement](#). Utilizzare il comando [uploadFile](#), ad esempio

```
epmAutomate uploadFile "C:/datafile/datafile.zip" inbox
```

per caricare il file ZIP di importazione nella cartella della posta in entrata di Data Management o in una cartella in essa contenuta.

Note:

Questo comando può importare solo i record di Data Management esportati da un altro ambiente in esecuzione nello stesso aggiornamento mensile. Ad esempio, i record esportati da un ambiente Oracle Enterprise Performance Management Cloud 21.11 possono essere importati solo in un altro ambiente 21.11.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Account Reconciliation, Tax Reporting, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate importDataManagement FILE_NAME.zip` dove *FILE_NAME* indica il nome del file ZIP che contiene i dati di Data Management da importare.

Esempi

- Importazione dalla cartella inbox di Data Management:
`epmautomate importDataManagement inbox/dataFile.zip`
- Importazione da una cartella contenuta nella cartella inbox:
`epmautomate importDataManagement inbox/dm_data/dataFile.zip`

importDimension

Importa una dimensione in un'applicazione Oracle Enterprise Data Management Cloud

Questo comando può importare un file di input da una connessione definita in Oracle Enterprise Data Management Cloud o nell'area temporanea.

Se il file deve essere importato dall'area temporanea di Oracle Enterprise Data Management Cloud, è necessario utilizzare il comando [uploadFile](#) per caricarlo nell'ambiente Oracle Enterprise Data Management Cloud target. È inoltre possibile utilizzare il comando [copyFileFromInstance](#) per copiare il file da un altro ambiente Oracle Enterprise Performance Management Cloud.

Si applica a

Oracle Enterprise Data Management Cloud.

Ruoli necessari

Amministratore servizi, Utente (con autorizzazione Gestione dati)

Utilizzo

`epmautomate importDimension APPLICATION DIMENSION IMPORT_TYPE FILE_NAME [connection=NAME]` dove:

- *APPLICATION* è il nome di un'applicazione Oracle Enterprise Data Management Cloud
- *DIMENSION* è il nome di una dimensione dell'applicazione in fase di importazione
- *IMPORT_TYPE* indica la modalità di esecuzione dell'importazione. I tipi di importazione validi sono i seguenti:
 - `ResetDimension`: consente di eliminare tutti i dati delle dimensioni esistenti e importare i nuovi dati

- `ReplaceNodes`: consente di aggiungere o aggiornare i nodi e sostituire le gerarchie esistenti durante l'importazione
- `Merge`: consente di elaborare modifiche incrementalmente ai nodi e alle gerarchie utilizzando una richiesta di importazione
- `FILE_NAME` è il nome del file (CSV o ZIP) contenente i dati della dimensione da importare. Il nome del file deve terminare con il nome della dimensione preceduto da `_` (carattere di sottolineatura), ad esempio `import_Entity.csv`. In caso di importazione da un file ZIP contenente più file di importazione, questo comando dipende dal nome di file all'interno del file ZIP per identificare il file di importazione corretto.
Se si specifica un valore per il parametro `connection`, è necessario importare la dimensione da un file ZIP, ad esempio `importdata_Entity.zip`.
- `connection=NAME` (facoltativo) identifica un nome di connessione (posizione dell'istanza) definito in Oracle Enterprise Data Management Cloud come posizione del file di importazione. Se il parametro non viene specificato, il processo di importazione cercherà il file di importazione nell'area temporanea locale.

Esempi

- Importare da un file caricato nell'area temporanea: `epmautomate importDimension USOperations Entity ReplaceNodes data_Entity.CSV`
- Importare dalla cartella outbox di un altro ambiente EPM Cloud: `epmautomate importDimension USOperations Entity ReplaceNodes data_Entity.ZIP Connection=EPM_Cloud_pln`

importJobConsole

Clona i record della console job utilizzando un file ZIP contenente i record della console job esportati a partire da un ambiente.

L'importazione dei record della console job tramite questo comando è un task che deve essere eseguito una sola volta dopo l'esecuzione del comando [recreate](#). Se si è già utilizzato questo comando per importare i record della console job, richiami successivi del comando avranno esito negativo finché non si ricrea l'ambiente.

Utilizzare il comando [exportJobConsole](#) (`epmAutomate exportJobConsole FILE_NAME.zip nDays=All jobTypes=All jobStatusCode=All`) per creare il file ZIP utilizzato come input per il comando.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate importJobConsole FILE_NAME.zip [logFileName=jobConsoleLog].I`
parametri sono descritti di seguito.

- *FILE_NAME* indica il nome del file ZIP contenente i record della console job che si desidera importare. Utilizzare il comando [uploadFile](#) per caricare questo file nell'ambiente.
- *logFileName* (facoltativo) identifica *jobConsoleLog* come file di log in cui verranno registrati gli errori rilevati durante l'importazione. Se non si specifica questo valore, il comando genera un file di errori denominato secondo questa convenzione: *usernameimportLog_date_timestamp.zip*. È possibile scaricare questo file utilizzando il comando [downloadFile](#).

Esempio

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

importLibraryArtifact

Importa gli artifact della libreria da un archivio o file nella libreria di Narrative Reporting.

Prima di eseguire questo comando, caricare l'archivio o il file di origine nell'ambiente utilizzando il comando [uploadFile](#).

Si applica a

Narrative Reporting

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore
Agli utenti che dispongono dei ruoli Utente avanzato, Utente e Visualizzatore deve essere concessa sicurezza supplementare mediante ACL.

Utilizzo

```
epmautomate importLibraryArtifact SOURCE_FILE [errorFile=ERROR_FILE.txt]
[importFormat=Native|File] [importFolder=FOLDER_PATH]
[ importPermission=true|false] [overwrite=true|false]. I parametri sono
descritti di seguito.
```

- *SOURCE_FILE* è il nome dell'archivio contenente gli artifact da importare nella libreria. Questo file deve essere disponibile nella cartella inbox.
- *errorFile* (facoltativo) è il nome univoco del file di testo in cui verranno memorizzati gli errori relativi all'importazione.
- *importFormat* (facoltativo) può essere impostato su uno dei valori descritti di seguito.
 - *Native* importa gli artifact da un file ZIP creato utilizzando il comando [exportLibraryArtifact](#) con l'opzione `exportFormat=Native`. Questo è il valore predefinito.
 - *File* importa un file binario.

 **Nota:**

Utilizzare il comando `importSnapshot` per importare file ZIP di artifact della libreria (creati utilizzando il comando `exportLibraryArtifact` con l'opzione `exportFormat=LCM`) in ambienti Financial Consolidation and Close, Planning, Planning Modules o Tax Reporting.

- `importFolder` (facoltativo) indica la posizione della libreria in cui verranno memorizzati gli artifact importati. Specificare questo percorso se la posizione è diversa da quella definita dal parametro `Library`, ovvero la posizione di importazione predefinita.
- `importPermission` indica se importare le autorizzazioni di accesso impostate per gli artifact. Il valore predefinito è `False`.
- `overwrite` determina se sovrascrivere gli eventuali artifact con lo stesso nome nella posizione specificata per la libreria. Il valore predefinito è `False`, ovvero il processo non importerà un artifact se nella posizione di importazione esiste un altro artifact con lo stesso nome.

Al completamento dell'importazione utilizzare il comando `downloadFile` per scaricare il file di esportazione e il file degli errori in un computer locale.

Esempi

- Importazione di un file nel formato binario:

```
epmautomate importLibraryArtifact newReports.doc
errorFile=report_imp_errors.txt importFormat=File importFolder="Library/My
Reports" importPermission=true overwrite=true
```
- Importazione di artifact da un file ZIP esportato:

```
epmautomate importLibraryArtifact newReports.zip
errorFile=report_imp_errors.txt importFormat=Native importFolder="Library/My
Reports" importPermission=true overwrite=true
```
- Importazione di report in un ambiente Financial Consolidation and Close, Planning, Planning Modules o Tax Reporting da un file ZIP esportato:

```
epmautomate importSnapshot newReports.zip
```

importLogoImage

Importa da un file di esportazione in un altro ambiente il logo aziendale utilizzato in un ambiente Account Reconciliation.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate importLogoImage IMAGE_NAME.jpg`, dove `IMAGE_NAME` indica il nome del file dell'immagine del logo.

È possibile utilizzare il comando [downloadFile](#) per scaricare l'immagine esportata. Caricarla nell'ambiente target utilizzando il comando [uploadFile](#) e quindi eseguire il comando [importLogoImage](#) per importarla.

Esempio

```
epmautomate importLogoImage corpLogo.jpg
```

importMapping

Importa i mapping da un file di importazione dei mapping precedentemente caricato nell'ambiente.

Utilizzare il comando [uploadFile](#) per caricare i file nella cartella inbox di Data Management oppure in una cartella al suo interno.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

```
epmautomate importMapping DIMENSION_NAME|ALL FILE_NAME IMPORT_MODE  
VALIDATION_MODE LOCATION dove:
```

- *DIMENSION_NAME|ALL* indica il destinatario del mapping. Specificare il nome della dimensione in cui i mapping devono essere importati oppure *ALL* per importare tutti i mapping inclusi nel file nelle dimensioni appropriate.
- *FILE_NAME* è il nome e la posizione del file di importazione dei mapping disponibile nella cartella inbox di Data Management oppure in una directory al suo interno. Specificare il nome di file (file TXT nel formato standard di Gestione dati) e il relativo percorso (ad esempio, inbox/AccountMap.txt oppure inbox/pbcs_maps/AccountMap.txt).
- *IMPORT_MODE* è *REPLACE* per cancellare le regole di mapping esistenti prima dell'importazione dei mapping oppure *MERGE* per aggiungere nuove regole di mapping alle regole esistenti.
- *VALIDATION_MODE* è *TRUE* per convalidare i membri target in base all'applicazione oppure *FALSE* per caricare il file di mapping senza eseguire le convalide.
- *LOCATION* è la posizione di Data Management per la quale è necessario caricare le regole di mapping.

Esempi

- ```
epmautomate importMapping Account inbox/AccountMap.txt MERGE FALSE
"France Sales"
```
- ```
epmautomate importMapping ALL "inbox/France Sales/AllMaps.txt" MERGE  
FALSE "France Sales" (carica i mapping dal file di importazione dei mapping in  
tutte le dimensioni mappate nell'ubicazione "France Sales")
```

importMetadata

Importa i metadati nell'applicazione utilizzando le impostazioni di importazione specificate in un job di tipo `import metadata`. Se lo si desidera, è possibile specificare il nome del file ZIP dal quale devono essere importati i metadati.

Utilizzare il comando [uploadFile](#) per caricare il file contenente i metadati nella posizione di caricamento predefinita.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate importMetadata JOB_NAME [FILE_NAME] errorFile=ERROR_FILE.zip dove:
```

- `JOB_NAME` è il nome di un job definito nell'applicazione.
- `FILE_NAME` (facoltativo) identifica il nome del file ZIP da cui devono essere importati i metadati. Se specificato, il contenuto di questo file ZIP ha la precedenza sui nomi di file definiti nel job. Il file ZIP può contenere uno o più file CSV. I nomi dei file contenenti i metadati per le dimensioni devono essere uguali ai nomi di file definiti nel job o terminare con `_DIMENSIONNAME.csv`; ad esempio, `metadata_Entity.csv`, `metadata_HSP_Smart Lists.csv` e `metadata_Exchange Rates.csv`.
- `errorFile`, facoltativamente, identifica il nome di un file ZIP in cui verranno registrati, se presenti, i record rifiutati durante le operazioni di importazione. I file ZIP con lo stesso nome eventualmente presenti nella cartella `outbox` verranno sovrascritti. È possibile scaricare questo file utilizzando il comando [downloadFile](#).

 **Nota:**

- Non è possibile rinominare i membri eseguendo un job di importazione dei metadata con un file di caricamento in cui viene modificata la proprietà `old_name` o `unique_name`. La ridenominazione di membri verrà ignorata.
- Non è possibile eliminare le dimensioni attributo durante l'importazione di metadata con l'uso di questo comando.
- Vengono importati solo i metadata per le dimensioni per le quali è impostata l'importazione dei metadata. I metadata per le altre dimensioni, se presenti nel file ZIP, vengono ignorati.

Si creerà una situazione di importazione ambigua se per il file ZIP risultano vere entrambe le seguenti condizioni:

- Il file ZIP contiene un file di metadata con un nome che corrisponde al nome di file definito nel job.
- Il file ZIP contiene uno o più file di metadata con nomi che terminano con `_DIMENSIONNAME.CSV` oppure `_DIMENSIONNAME.TXT`, dove `DIMENSIONNAME` è il nome della dimensione in cui verranno importati i metadata.

Oracle consiglia di disporre di un file ZIP contenente un file di metadata con un nome identico a quello a cui viene fatto riferimento nel job oppure un file con un nome che termina con `_DIMENSIONNAME.CSV` (o `_DIMENSIONNAME.TXT`), ma non entrambi. Ad esempio, se si carica un job che fa riferimento al file di metadata `Employees_A-Z.CSV` nella dimensione `Employees`, il file ZIP può includere il file `Employees_A-Z.CSV` o `New_Employees.CSV`, ma non entrambi. Se il file ZIP contiene sia `Employees_A-Z.CSV` che `New_Employees.CSV`, EPM Automate può selezionare uno dei due file per l'importazione a seconda dell'ordine dei file all'interno del file ZIP. Il file `Employees_A-Z.CSV` è una possibile corrispondenza per l'importazione perché il relativo nome corrisponde al nome di file a cui viene fatto riferimento nel job. Anche il file `New_Employees.CSV` è una possibile corrispondenza perché il relativo nome corrisponde al pattern `_DIMENSIONNAME.CSV`.

Esempio

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

importOwnershipData

Importa i dati di proprietà da un file CSV disponibile nell'ambiente in un periodo.

Prima di eseguire questo comando, utilizzare il comando [uploadFile](#) per caricare il file CSV di origine importazione nell'ambiente.

L'intestazione di questo file CSV è la seguente:

```
Scenario, Year, Period, Entity, Parent, POwn, Control, Method
```

I valori `POwn`, `Control` e `Method` sono facoltativi.

I dati proprietà importati verranno uniti a eventuali dati esistenti e questo potrebbe creare voci proprietà non valide. Se un'entità è presente in più rami di una gerarchia, i dati proprietà importati possono far salire la % di proprietà combinata dell'entità oltre il 100%. È necessario correggere manualmente la % di proprietà per garantire che non superi il 100%.

Si applica a

Financial Consolidation and Close e Tax Reporting.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente
Gli utenti devono disporre dell'accesso in scrittura all'entità.

Utilizzo

`epmautomate importOwnershipData Scenario Year Period FILE_NAME` dove:

- `Scenario` è lo scenario nel quale importare i dati proprietà.
- `Year` è l'anno nel quale importare i dati.
- `Period` è il periodo dell'anno nel quale importare i dati proprietà.
- `FILE_NAME` è il nome del file CSV dal quale importare i dati.

Esempio

```
epmautomate importOwnershipData FCCS_TotalActual FY19 Jan importfile.csv
```

importPreMappedBalances

Importa i dati di saldo premappati da un file nel repository di Account Reconciliation.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore
È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

`epmautomate importPreMappedBalances PERIOD FILE_NAME BALANCE_TYPE CURRENCY_BUCKET` dove:

- `PERIOD` è il nome di un periodo
- `FILE_NAME` è il nome del file CSV contenente i dati da importare.
- `BALANCE_TYPE` è SRC o SUB
- `CURRENCY_BUCKET` è Entered, Functional o Reporting

Esempio

```
epmautomate importPreMappedBalances "January 2015" dailydata.csv SRC  
Reporting
```

importPreMappedTransactions

Importa le transazioni premappate da un file CSV nel repository di Account Reconciliation.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore
È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

```
epmautomate importPreMappedTransactions PERIOD TRANSACTION_TYPE FILE_NAME  
DATE_FORMAT dove:
```

- *PERIOD* è il nome di un periodo
- *TRANSACTION_TYPE* è uno dei seguenti valori:
 - BEX per il caricamento delle spiegazioni saldo
 - SRC per il caricamento gli adeguamenti del sistema di origine
 - SUB per il caricamento degli adeguamenti del sottosistema
 - VEX per il caricamento delle spiegazioni dell'analisi della varianza
- *FILE_NAME* è il nome del file CSV dal quale importare i dati
- *DATE_FORMAT* è la stringa di testo per il formato della data, ad esempio MMM d, yyyy.

Esempio

```
epmautomate importPreMappedTransactions "January 2015" "BEX"  
transactions.csv "MMM d, yyyy"
```

importProfiles

Importa le nuove definizioni di profilo da un file CSV nel repository di Account Reconciliation.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

`epmautomate importProfiles FILE_NAME PROFILE_TYPE METHOD DATE_FORMAT` dove:

- `FILE_NAME` è il nome del file CSV dal quale importare i dati
- `PROFILE_TYPE` è `profiles` o `children`
- `METHOD` è `Replace` o `Update`
- `DATE_FORMAT` è una stringa di testo per il formato della data, ad esempio `MMM d, yyyy`

Esempio

```
epmautomate importProfiles NewRecProfiles.csv Profiles Replace "MMM d, yyyy"
```

importRates

Importa i tassi valuta da un file CSV nel repository di Account Reconciliation.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

`epmautomate importRates PERIOD RATE_TYPE REPLACE_MODE FILE_NAME` dove:

- `PERIOD` è il nome di un periodo
- `RATE_TYPE` è un tipo di tasso predefinito.
- `REPLACEMENT_MODE` è `Replace` o `ReplaceAll`
- `FILE_NAME` è il nome del file CSV dal quale importare i tassi

Esempio

```
epmautomate importRates "January 2015" Actual ReplaceAll avgrates.csv
```

importRCAttributeValues

Importa i valori di attributo nell'elenco Conformità riconciliazione o negli attributi di gruppo di Account Reconciliation.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato

Gli utenti avanzati possono richiedere sicurezza supplementare tramite gli elenchi ACL.

Utilizzo

```
epmautomate importRCAttributeValues ATTRIBUTE_NAME FILE_NAME
[METHOD=REPLACE|REPLACE ALL|UPDATE] [DATEFORMAT=DD/MM/YYYY|DD-MMM-YYYY|MMM
d,yyyy|All], dove:
```

- *ATTRIBUTE_NAME* è il nome di un elenco o di un attributo di gruppo nel quale importare i valori.
- *FILE_NAME* è un file di importazione CSV dal quale importare i valori. Utilizzare il comando [uploadFile](#) per caricare il file nell'ambiente prima di eseguire il comando.
- *METHOD*, facoltativo, è la modalità di importazione dei valori. Valori validi:
 - *Replace* per aggiungere tutti i valori dal file di importazione come valore di attributo in Conformità riconciliazione. I valori di attributo esistenti verranno sostituiti con quelli nel file di importazione mentre verranno aggiunti i valori non presenti nell'attributo ma presenti nel file di importazione. Non verranno modificati i valori esistenti nell'attributo ma non nel file di importazione. Si noti che tutti i dati di attributo per un determinato valore chiave verranno sostituiti con i contenuti del file oppure verranno cancellati. I nuovi valori verranno aggiunti in fondo, nell'ordine in cui compaiono nel file.
Questo tipo di importazione è particolarmente utile quando si spostano solo le modifiche più recenti da un sistema di origine, ad esempio quando si aggiungono nuovi dati di memorizzazione da un'acquisizione per sostituire solo gli eventuali valori di attributo specificati con i valori del file di importazione. Si tratta dell'impostazione predefinita.
 - *Replace All* per sostituire il valore di attributo esistente con i valori dell'importazione. Verranno eliminati i valori esistenti nell'attributo ma non presenti nel file di importazione.
Questo tipo di importazione è molto utile per il mirroring dei valori da un sistema di origine con un aggiornamento completo, ad esempio per completare gli aggiornamenti settimanali da sincronizzare con i dati memorizzati dal sistema di origine.
 - *Update* per sostituire o aggiungere all'attributo tutti i valori del file di importazione. I valori esistenti dell'attributo verranno sostituiti con quelli contenuti nel file di importazione. Verranno aggiunti i valori contenuti nel file di importazione ma non presenti nell'attributo. I valori presenti nell'attributo ma non contenuti nel file di importazione, non verranno modificati. Solo i dati di attributo per un determinato valore chiave verranno sostituiti con il contenuto del file; i dati degli attributi non disponibili nel file non verranno toccati.
L'eventuale presenza di una chiave nel file di importazione ma non nell'attributo determinerà un errore.
Questo tipo di importazione è utilissimo per aggiornare alcuni attributi su tutti i valori, ad esempio per aggiornare la gestione dell'area di memorizzazione dopo una riorganizzazione senza influenzare il resto dei dati della memoria.
- *Dateformat*, opzionale, consente di specificare i formati data validi (ad esempio, DD/MM/YYYY, DD-MMM-YYYY (predefinito), MMM d,yyyy e All) per l'analisi. È possibile specificare più formati data separati da un punto e virgola.

Esempio

```
epmautomate importRCAttributeValues Stores StoreData.csv METHOD=Replace
DATEFORMAT="All"
```

importReconciliationAttributes

Importa gli attributi di riconciliazione nelle riconciliazioni esistenti da un file caricato nell'ambiente Account Reconciliation utilizzando il comando [uploadFile](#) .

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

```
epmautomate importReconciliationAttributes FILE.CSV Period [Rules=RULE_NAME]
[Reopen=true|false] [Dateformat=DATE_FORMAT] dove:
```

- **FILE** è il nome del file CSV contenente gli attributi di riconciliazione da importare nelle riconciliazioni.
- **Period** identifica il periodo a cui appartengono le riconciliazioni.
- **Rules** (facoltativo) identifica le regole da eseguire sulle riconciliazioni interessate dopo l'importazione degli attributi. Utilizzare la virgola per separare più nomi di regole. I valori validi sono:
 - **None**: non esegue alcuna regola sulle riconciliazioni interessate. Questa è l'impostazione predefinita; non deve essere combinata con altri valori.
 - **ALL**: esegue tutte le regole definite per le riconciliazioni per il periodo specificato. Questo valore deve essere utilizzato da solo; non può essere combinato con altri nomi di regole.
 - **SET_ATTR_VAL**: esegue la regola predefinita per impostare il valore dell'attributo.
 - **CRT_ALT**: esegue la regola predefinita per creare un avviso.
 - **AUTO_APP**: esegue la regola predefinita per approvare automaticamente la riconciliazione.
 - **AUTO_SUB**: esegue la regola predefinita per sottomettere automaticamente la riconciliazione.
 - **EMAIL_ON_SAVE**: esegue la regola predefinita per inviare automaticamente un messaggio e-mail dopo l'aggiornamento della riconciliazione.
- **Reopen** (facoltativo) specifica se riaprire le riconciliazioni modificate una volta completata l'operazione di importazione. Il valore predefinito è `false`.
- **Dateformat** (facoltativo) specifica i formati data validi (ad esempio, `MM-dd-yyyy`, `dd-MMMM-yy`, `MMM d e yyyy` da analizzare). È possibile specificare più formati data separati da un punto e virgola.

Esempi

- **Importazione dei valori degli attributi per un periodo ed esecuzione di più regole con vari formati data:**

```
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"
```
- **Importazione dei valori degli attributi per un periodo senza eseguire regole:**

```
epmAutomate importReconciliationAttributes Reconciliations.csv "July 2020"
```
- **Importazione dei valori degli attributi per un periodo, esecuzione di tutte le regole applicabili e riapertura delle riconciliazioni interessate:**

```
epmAutomate importReconciliationAttributes Reconciliations.csv "July 2020" Rules=ALL Reopen=true
```

importSnapshot

Importa il contenuto di uno snapshot nell'ambiente del servizio. Lo snapshot importato deve essere disponibile nella posizione di caricamento predefinita.

Utilizzare il comando [uploadFile](#) per caricare uno snapshot oppure il comando [copySnapshotFromInstance](#) per copiarlo da un'altra istanza.

- Quanto riportato di seguito non fa parte degli snapshot delle applicazioni Planning, Planning Modules e FreeForm.
 - Dati di audit
 - Dati della console job

Utilizzare il comando [cloneEnvironment](#) o la funzione Clona ambiente se si desidera copiare i dati di audit e della console job nell'ambiente target. L'importazione di snapshot potrebbe non riuscire se il processo aziendale di Planning contiene un membro del periodo generato rinominato che è stato sostituito da un membro del periodo customizzato. Ad esempio, si è rinominato il membro periodo *YearTotal* generato in *unused_YearTotal* a e quindi è stato aggiunto un membro periodo di tipo alternativo con il nome del membro originale (*YearTotal* in questo esempio). In questo scenario, l'importazione dello snapshot potrebbe non riuscire.

- Gli snapshot non contengono i dati delle tabelle intermedie di Data Management. Per importare questi dati, utilizzare i comandi [exportDataManagement](#) e [importDataManagement](#) o l'interfaccia degli script di manutenzione del sistema di Data Management. È possibile utilizzare il comando [cloneEnvironment](#) o la funzione Clona ambiente per creare una copia identica dell'ambiente, inclusi i dati delle tabelle intermedie di Data Management.

Le attività che è possibile completare utilizzando questo comando dipendono dal ruolo.

- Gli amministratori dei servizi possono importare solo gli artifact dell'applicazione in un ambiente.
- Sono necessari entrambi i ruoli Amministratore servizi e Amministratore del dominio di Identity per importare il contenuto dell'applicazione nell'ambiente del servizio e gli artifact del dominio di Identity (utenti e le relative assegnazioni ruolo) nel dominio di Identity dell'ambiente.

Se nello snapshot in fase di importazione viene fatto riferimento a un utente non incluso nel dominio di Identity, EPM Automate crea un utente nel dominio di Identity e assegna la password predefinita specificata nel comando oppure una password univoca temporanea a ciascun utente, se non si specifica una password nel comando. Per impostazione predefinita, l'utente dovrà reimpostare la password durante il primo accesso.

Nota:

- Per i processi aziendali diversi da Account Reconciliation, Profitability and Cost Management e Oracle Enterprise Data Management Cloud: durante il caricamento dei metadati, è possibile che Oracle Enterprise Performance Management Cloud ripeta più volte il caricamento se il tentativo precedente ha dato origine a record rifiutati perché membri condivisi precedono i membri di base nel profilo. Questi tentativi possono determinare un aumento dei tempi di elaborazione del comando.
- Gli utenti membri di gruppi in Controllo accesso devono essere assegnati a un ruolo predefinito. Non sono consentiti tentativi di assegnare a un gruppo un utente non assegnato a un ruolo predefinito.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato assegnati al ruolo applicazione Amministratore migrazione

Il ruolo amministratore del dominio di Identity è necessario per importare gli utenti e le assegnazioni dei ruoli predefiniti.

Utilizzo

```
epmautomate importSnapshot SNAPSHOT_NAME [importUsers=true|false]
[userPassword=DEFAULT_PASSWORD] [resetPassword=true|false] dove:
```

- *SNAPSHOT_NAME* è il nome di uno snapshot nella posizione di caricamento predefinita.
- *importUsers* (facoltativo) specifica se importare gli utenti e le loro assegnazioni predefinite dei ruoli dallo snapshot. Il valore predefinito è *false*. Utilizzare *importUsers=true* per importare gli utenti e le assegnazioni dei ruoli predefiniti nel dominio di Identity se nello snapshot di origine sono contenuti dati sui nuovi utenti o se sono stati assegnati nuovi ruoli agli utenti correnti.
Per i valori di accesso utente non è prevista la distinzione tra maiuscole e minuscole. Ad esempio, il valore di accesso utente *jane.doe@example.com* viene considerato uguale a *Jane.Doe@Example.com* o a qualsiasi variazione di maiuscole e minuscole. Se una variante corrisponde a un valore di accesso utente esistente nel dominio di Identity, il comando non importa l'utente dallo snapshot.

 **Nota:**

- Se un utente che non è amministratore del dominio di Identity importa gli utenti e i loro ruoli predefiniti, l'operazione non riuscirà. Nel report sullo stato della migrazione viene registrato il seguente errore: `Impossibile importare l'artifact di directory esterna ARTIFACT_NAME. L'utente USER_NAME non è autorizzato a eseguire questa operazione. L'utente deve disporre del ruolo Amministratore del dominio di Identity per poter eseguire questa operazione.`
 - Se non si stanno importando utenti e un utente nello snapshot di origine non è assegnato a un ruolo predefinito nell'ambiente target, viene visualizzato un errore (`EPMIE-00070: Impossibile trovare il membro utente durante l'importazione dei ruoli assegnati`).
-
- Le modifiche apportate ai ruoli predefiniti dell'utente verranno aggiornate in base ai ruoli assegnati nello snapshot di origine. Le assegnazioni dei ruoli nel target tuttavia non verranno rimosse per garantire la corrispondenza con quelle nello snapshot di origine. Si supponga ad esempio che l'utente `jdoe` sia assegnato al ruolo predefinito Utente avanzato nell'ambiente target, ma che disponga solo del ruolo Utente nello snapshot di origine. In questo caso, questo comando assegna `jdoe` al ruolo Utente e non rimuove l'assegnazione del ruolo Utente avanzato nell'ambiente target.
 - Questo comando non elimina gli utenti esistenti dall'ambiente target se non sono presenti nello snapshot di origine. Si supponga ad esempio che `jdoe` disponga di un account nell'ambiente target, ma che questo account non sia presente nello snapshot di origine. In questo caso, l'account di `jdoe` nell'ambiente target non viene eliminato.
 - Questo comando aggiunge utenti non presenti nell'ambiente target. Non aggiorna le proprietà degli utenti correnti nell'ambiente target anche se sono diverse nello snapshot di origine. Se ad esempio il cognome di `jdoe` nello snapshot di origine è scritto in modo diverso rispetto all'ambiente target, non verranno apportate modifiche nell'ambiente target.
 - Questo comando non cambia le password degli utenti esistenti nell'ambiente target anche se sono diverse nello snapshot di origine.
- `userPassword` (facoltativo) indica la password predefinita da assegnare ai nuovi utenti creati nel dominio di Identity. La password specificata deve soddisfare i requisiti minimi per le password. Se non si specifica un valore per questo parametro, a ogni utente viene assegnata una password temporanea univoca.
 - `resetPassword` (facoltativo) indica se il nuovo utente deve modificare la password al primo accesso. Il valore predefinito è `true` e ciò significa che i nuovi utenti devono modificare la password al primo accesso. Se questo valore è impostato su `true`, i nuovi utenti riceveranno e-mail di attivazione degli account in cui verrà richiesto di cambiare le password.

Esempi

- Importare solo gli artifact dell'applicazione: `epmautomate importSnapshot April16FullApp`
- Importare gli artifact di applicazioni e dominio di Identity (sono necessari i ruoli amministratore servizi e amministratore del dominio di Identity):
 - Assegnare una password univoca temporanea a ogni nuovo utente e richiedere che gli utenti la reimpostino dopo il primo accesso:
`epmautomate importSnapshot April16FullApp importUsers=true`
 - Assegnare una password specifica e consentire agli utenti di non modificarla, se lo desiderano. Questa impostazione non è raccomandata per le importazioni negli ambienti di produzione:
`epmautomate importSnapshot April16FullApp importUsers=true
userPassword=P@ssw0rd1 resetPassword=false`

importSupplementalCollectionData

Importa dati raccolta supplementari da un file nell'applicazione.

Utilizzare il comando [uploadFile](#) per caricare il file contenente i dati nella posizione di caricamento predefinita. Il formato del file di importazione è il seguente:

```
#Workflow
Workflow_Dimension_1_Name,Workflow_Dimension_2_Name,Workflow_Dimension_n_Name
Workflow_Dimension_1_Member,Workflow_Dimension_2_Member,Workflow_Dimension_n_Member
#Collection
Collection_Attribute_1,Collection_Attribute_2,Collection_Attribute_n
Record1_Attr_Value_1,Record1_Attr_Value_2, Record1_Attr_Value_n
```

Ad esempio:

```
#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
```

Si applica a

Financial Consolidation and Close, e Tax Reporting.

Ruoli necessari

Amministratore servizi

Utilizzo



Nota:

Tutti i parametri dei comandi devono essere racchiusi tra virgolette doppie.

```
epmautomate importSupplementalCollectionData "FILE_NAME" "COLLECTION_NAME"  
"YEAR" "PERIOD" "[FREQUENCY_DIMENSION=MEMBER]" dove:
```

- FILE_NAME è il nome di un file CSV, disponibile nella posizione di caricamento predefinita, contenente i dati supplementari formattati correttamente.
- COLLECTION_NAME è il nome della raccolta in cui importare i dati supplementari nel file.
- YEAR è il membro dimensione anno da usare per la raccolta.
- PERIOD è il nome della dimensione periodo da usare per la raccolta.
- FREQUENCY_DIMENSION (facoltativo) è il nome della dimensione frequenza da usare per la raccolta. È possibile specificare il numero di dimensioni frequenza richieste nel formato "FREQUENCY_DIMENSION1=MEMBER" "FREQUENCY_DIMENSION2=MEMBER".

Esempio

```
epmautomate importSupplementalCollectionData "datafile.csv" "Journal Data  
Collection" "FY20" "Jan" "Account=PAYROLL" "JournalID=LNR 113"
```

importSupplementalData

Importa dati supplementari da un file nell'applicazione.

Utilizzare il comando [uploadFile](#) per caricare il file contenente i dati nella posizione di caricamento predefinita.

Si applica a

Financial Consolidation and Close e Tax Reporting.

Ruoli necessari

Amministratore servizi

Utilizzo



Nota:

Tutti i parametri dei comandi devono essere racchiusi tra virgolette doppie.

```
epmautomate importSupplementalData "FILE_NAME" "DATA_SET_NAME" "YEAR"  
"PERIOD_NAME" "SCENARIO_NAME" dove:
```

- `FILE_NAME` è il nome di un file CSV, disponibile nella posizione di caricamento predefinita, contenente i dati supplementari formattati correttamente.
- `DATA_SET_NAME` è il nome del set di dati in cui devono essere importati i dati supplementari nel file.
- `YEAR` è l'anno per il quale il set di dati viene distribuito.
- `PERIOD_NAME` è il nome del periodo in cui il set di dati viene distribuito.
- `SCENARIO_NAME` è il nome dello scenario in cui il set di dati viene distribuito.

Esempio

```
epmautomate importSupplementalData "DatasetImport.csv" "EmployeeDataSet" "FY17"
"Jan" "Actual"
```

importTemplate

Crea una struttura di applicazione mediante l'importazione da un file template esistente nella cartella profitinbox.

È possibile caricare un file template nella cartella profitinbox utilizzando il comando [uploadFile](#).

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

```
epmautomate importTemplate APPLICATION_NAME File_Name
isApplicationOverwrite=true|false dove:
```

- `APPLICATION_NAME` è il nome dell'applicazione Profitability and Cost Management che si desidera creare importando il template.
- `File_Name` è il nome del file .ZIP contenente il template dell'applicazione. Questo file deve esistere nella cartella profitinbox.
- `isApplicationOverwrite` specifica se sovrascrivere l'applicazione esistente, se disponibile Specificare questo valore di parametro in minuscolo.

Esempio

```
epmautomate importTemplate BksML12 template1.zip isApplicationOverwrite=true
```

importTMAttributeValues

Importa valori negli attributi di gruppo Corrispondenza transazione di Account Reconciliation.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato

Gli utenti avanzati possono richiedere sicurezza supplementare tramite gli elenchi ACL.

Utilizzo

```
epmautomate importTMAtributeValues ATTRIBUTE_NAME FILE_NAME
[METHOD=REPLACE|REPLACE ALL|UPDATE] [DATEFORMAT=DD/MM/YYYY|DD-MMM-YYYY|MMM
d,yyyy|All], dove:
```

- *ATTRIBUTE_NAME* è il nome di un attributo di gruppo nel quale importare i valori.
- *FILE_NAME* è un file di importazione CSV dal quale importare i valori in Corrispondenza transazione. Utilizzare il comando [uploadFile](#) per caricare il file nell'ambiente prima di eseguire il comando.
- *METHOD*, facoltativo, è la modalità di importazione dei valori. Valori validi:
 - *Replace* per aggiungere tutti i valori del file di importazione negli attributi di gruppo di Corrispondenza transazione. I valori di attributo esistenti verranno sostituiti con quelli nel file di importazione mentre verranno aggiunti i valori non presenti nell'attributo ma presenti nel file di importazione. Non verranno modificati i valori esistenti nell'attributo ma non nel file di importazione. Si noti che tutti i dati di attributo per un determinato valore chiave verranno sostituiti con i contenuti del file oppure verranno cancellati. I nuovi valori verranno aggiunti in fondo, nell'ordine in cui compaiono nel file. Questo tipo di importazione è particolarmente utile quando si spostano solo le modifiche più recenti da un sistema di origine, ad esempio quando si aggiungono nuovi dati di memorizzazione da un'acquisizione per sostituire solo gli eventuali valori di attributo specificati con i valori del file di importazione. Si tratta dell'impostazione predefinita.
 - *Replace All* per sostituire il valore di attributo esistente con i valori dell'importazione. Verranno eliminati i valori esistenti nell'attributo ma non presenti nel file di importazione. Questo tipo di importazione è molto utile per il mirroring dei valori da un sistema di origine con un aggiornamento completo, ad esempio per completare gli aggiornamenti settimanali da sincronizzare con i dati memorizzati dal sistema di origine.
 - *Update* per sostituire o aggiungere all'attributo tutti i valori del file di importazione. I valori esistenti dell'attributo verranno sostituiti con quelli contenuti nel file di importazione. Verranno aggiunti i valori contenuti nel file di importazione ma non presenti nell'attributo. I valori presenti nell'attributo ma non contenuti nel file di importazione, non verranno modificati. Solo i dati di attributo per un determinato valore chiave verranno sostituiti con il contenuto del file; i dati degli attributi non disponibili nel file non verranno toccati. L'eventuale presenza di una chiave nel file di importazione ma non nell'attributo determinerà un errore. Questo tipo di importazione è utilissimo per aggiornare alcuni attributi su tutti i valori, ad esempio per aggiornare la gestione dell'area di memorizzazione dopo una riorganizzazione senza influenzare il resto dei dati della memoria.
- *Dateformat*, opzionale, consente di specificare i formati data validi (ad esempio, DD/MM/YYYY, DD-MMM-YYYY (predefinito), MMM d,yyyy e All) per l'analisi. È possibile specificare più formati data separati da un punto e virgola.

Esempio

```
epmautomate importTMAttributeValues TMGA TMGA.csv METHOD=Replace
DATEFORMAT="All"
```

importTmPremappedTransactions

Per un'origine dati specifica, importa i dati delle transazioni premappate da un file nel repository di Account Reconciliation in Corrispondenza transazione.

Utilizzare il comando [uploadFile](#) per caricare il file delle transazioni nel servizio.

Questo comando visualizza lo stato di importazione e un file di log dell'importazione nella console. Utilizzare il comando [downloadFile](#) per scaricare il file di log in un computer locale.

Vedere Importazione dei dati in *Riconciliazione dei conti con Account Reconciliation* per i requisiti relativi al formato dei file di importazione e per informazioni sull'importazione dei dati.

Nota:

- È possibile importare transazioni per un solo tipo di corrispondenza per volta. Tuttavia è possibile eseguire importazioni parallele in tipi di corrispondenza diversi.
- A differenza della schermata Job, è possibile importare i dati delle transazioni premappate solo da un file alla volta.
- Dopo aver importato le transazioni premappate per tutte le origini dati, eseguire il comando `runautomatch`.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

```
epmautomate importTmPremappedTransactions MATCH_TYPE DATA_SOURCE FILE_NAME
[DATE_FORMAT] dove:
```

- `MATCH_TYPE` è un tipo di corrispondenza definito in Account Reconciliation.
- `DATA_SOURCE` è l'identificativo dell'origine dati associata al tipo di riconciliazione specificato.
- `FILE_NAME` è il nome del file CSV contenente le transazioni da importare. Questo file deve essere disponibile nel servizio.

- DATE_FORMAT è un parametro facoltativo che indica il formato dei campi data inclusi nel file di importazione delle transazioni. Il valore predefinito è dd-*MMM*-YYYY. Gli altri formati di data supportati sono: MM/dd/yyyy, dd/MM/yyyy, MM-dd-yyyy, d-M-yyyy e MMM d.yyyy.

Esempio

```
epmautomate importTmPremappedTransactions "INTERCOMPANY" "AP"
dailydata.csv d-M-yyyy
```

importValidIntersections

Importa i gruppi di intersezioni valide da un file ZIP contenente un file Excel con le definizioni delle intersezioni valide nel processo aziendale. Prima di eseguire questo comando, utilizzare il comando [uploadFile](#) per caricare il file di importazione nell'ambiente.

Il file ZIP di importazione deve contenere un file Excel con due fogli di calcolo (Regole e Regole secondarie) per importare le intersezioni valide. Il primo foglio, Regole, deve definire il gruppo di intersezioni, incluse le dimensioni, e le proprietà come "necessarie altre dimensioni e valide non specificate". L'altro foglio, Regole secondarie, deve contenere le esclusioni e le selezioni membro. Per ulteriori informazioni, fare riferimento agli argomenti della Guida *Amministrazione di Planning* indicati di seguito.

- Dimensioni di ancoraggio e non
- Esempi di intersezioni valide

Il metodo migliore per ottenere il template del formato del file di importazione è esportare le interazioni valide dall'applicazione. Un formato campione è mostrato nelle seguenti immagini.

	A	B	C	D	E	F	G	H	I	J
	Name	Position	Description	Enabled	Anchor Dim Name	Anchor Dimension Apply	Dim1	Dim1 Required	Dim2	Dim2 Required
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
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							

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate importValidIntersections FILE_NAME.zip
[ErrorFile=ERROR_FILE_NAME.txt] dove:
```

- FILE_NAME è il nome del file ZIP contenere il file Excel con le definizioni delle intersezioni valide.
- ErrorFile (facoltativo) identifica il nome del file di testo in cui verranno scritti i record di errore. Se questo parametro non è specificato, EPM Automate genera automaticamente un file degli errori il cui nome può essere visualizzato nella console dei job.

Esempio

```
epmautomate importValidIntersections VI_Import_File.zip
ErrorFile=VI_Import_Log.txt
```

invalidLoginReport

Negli ambienti OCI (Gen 2), crea il report degli accessi non validi, che elenca i tentativi di accesso all'ambiente non riusciti su un periodo di tempo specifico corrispondente al periodo di conservazione audit specificato per l'ambiente. Il periodo di conservazione predefinito è di 30 giorni. È possibile estenderlo a un massimo di 90 giorni modificando l'impostazione di **Periodo di conservazione audit (giorni)** nella console di Oracle Cloud Identity. Per conservare i dati di audit per un periodo superiore a 90 giorni, scaricare e archiviare periodicamente questo report e il [report di audit di assegnazioni ruoli](#).

Il report degli accessi non validi contiene informazioni come ad esempio quelle elencate di seguito.

- Nome dell'utente che ha tentato di eseguire l'accesso
- Indirizzo IP remoto da cui l'utente ha tentato di eseguire l'accesso
- Indicatore orario del tentativo di accesso

In questo report vengono mostrati tutti i tentativi di accesso non riusciti all'istanza di Identity Cloud Service corrispondente. Potrebbero non essere tutti correlati a un'istanza di Oracle Enterprise Performance Management Cloud.

	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

Utilizzare il comando [downloadFile](#) per eseguire il download dei report in un computer locale.

Si applica a

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, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore del dominio di Identity e qualsiasi ruolo predefinito (Amministratore servizi, Utente avanzato, Utente o Visualizzatore)

Utilizzo

`epmAutomate invalidLoginReport FROM_DATE TO_DATE FILE_NAME.CSV`. I parametri sono descritti di seguito.

- `FROM_DATE` indica la data di inizio del periodo per cui deve essere generato il report (in formato YYYY-MM-DD). Questa data deve rientrare nel periodo di conservazione dei dati dell'audit specificato nella console di Oracle Cloud Identity.
- `TO_DATE` indica la data di fine del periodo per cui deve essere generato il report (in formato YYYY-MM-DD).
- `FILE_NAME` indica il nome di un file CSV per il report.



Note:

Questo report può essere generato solo per gli ultimi 90 giorni.

Esempio

```
epmAutomate invalidLoginReport 2021-06-01 2021-06-30
invalidLoginReport.CSV
```

listBackups

Elenca gli snapshot di backup disponibili degli ambienti OCI (Gen 2) per determinare se è disponibile un backup specifico, in modo che sia possibile archivarlo o utilizzarlo per ripristinare personalmente l'ambiente corrente. Questo comando non funziona negli ambienti Oracle Enterprise Performance Management Cloud Classic.

Prima di provare a ripristinare un backup specifico, utilizzare questo comando per controllare se il backup richiesto è disponibile in Oracle Object Storage. Se il backup è disponibile, è possibile ripristinarlo (copiarlo nell'ambiente) eseguendo il comando [restoreBackup](#). Dopo aver copiato il backup, è possibile importarlo utilizzando il comando [importSnapshot](#). Il ripristino self-service dell'ambiente consente di ridurre il tempo di elaborazione.

Per i servizi diversi da Narrative Reporting, questo comando elenca gli snapshot di backup disponibili (creati dal processo di manutenzione giornaliera) utilizzando la convenzione di denominazione YYYY-MM-DDTHH:MM:SS/Artifact_Snapshot.zip, ad

esempio 2022-02-16T21:00:02/Artifact_Snapshot.zip. Per Narrative Reporting, gli snapshot disponibili utilizzano la convenzione di denominazione YYYY-MM-DDTHH:MM:SS/EPRCS_Backup.tar.gz, ad esempio 2022-02-16T21:00:02/EPRCS_Backup.tar.gz. In entrambi i casi, l'indicatore orario riporta l'ora di creazione dello snapshot in formato UTC. La figura riportata di seguito illustra un esempio di output del comando.

```
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
```

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato assegnati al ruolo applicazione Amministratore migrazione

Utilizzo

```
epmAutomate listBackups
```

Esempio

```
epmAutomate listBackups
```

listFiles

Elenca i nomi dei file nella posizione predefinita, nelle cartelle di Gestione dati e in profitinbox/profitoutbox (Profitability and Cost Management).

Questo comando elenca inoltre i file di esportazione incrementali e di backup, gli snapshot di Migrazione, i log di accesso e i report attività. Questa figura illustra una versione troncata dell'output del comando.

```
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
```

Questo comando non elenca lo snapshot corrente se viene eseguito durante la generazione dello snapshot dell'ambiente, ad esempio durante la manutenzione giornaliera.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato assegnati al ruolo applicazione
Amministratore migrazione

Utilizzo

```
epmautomate listFiles
```

Esempio

```
epmautomate listFiles
```

loadData

Carica i dati in un cubo di calcolo utilizzando un file disponibile nella cartella profitinbox.

Utilizzare il comando [uploadFile](#) per caricare i file nella cartella profitinbox.

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate loadData APPLICATION_NAME dataFileName=File_Name PARAMETER=VALUE`
dove:

- `APPLICATION_NAME` è il nome dell'applicazione Profitability and Cost Management in cui si desidera caricare i dati
- `dataFileName=File_Name` specifica un file di caricamento dei dati disponibile nella cartella profitinbox.
Il nome del file di dati deve essere racchiuso tra virgolette.
- `PARAMETER=VALUE` indica i parametri runtime e i relativi valori per il caricamento dei dati. Specificare il numero di parametri e coppie di valori richiesto dal processo. I parametri validi e i relativi valori sono indicati di seguito.
 - `clearAllDataFlag=true|false` specifica se cancellare i dati esistenti nel cubo dell'applicazione
 - `dataLoadValue=OVERWRITE_EXISTING_VALUES|ADD_TO_EXISTING` specifica come gestire i dati esistenti

Esempio

```
epmautomate loadData BksML12 dataFileName="data1.txt"clearAllDataFlag=true
dataLoadValue="OVERWRITE_EXISTING_VALUES"
```

loadDimData

Carica i metadati delle dimensioni da uno o più file nella cartella profitinbox in un'applicazione.

Utilizzare il comando [uploadFile](#) per caricare i metadati nella cartella profitinbox.

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate loadDimData APPLICATION_NAME dataFileName=File_Name`
[stringDelimiter="DELIMITER"] dove:

- `APPLICATION_NAME` è il nome dell'applicazione Profitability and Cost Management in cui si desidera caricare i metadati delle dimensioni.
- `dataFileName` specifica un file di caricamento dei metadati delle dimensioni nella cartella profitinbox. Per caricare i metadati da più file, elencare i nomi di file separando ciascun nome con un delimitatore.

- `stringDelimiter` specifica il delimitatore utilizzato per separare i nomi di file di metadati. Il delimitatore deve essere racchiuso tra virgolette.

Esempio

```
epmautomate loadDimData BksML12 dataFileName="dimdata1.txt#dimdata1.txt"
stringDelimiter="#"
```

loadViewpoint

Carica un punto di vista (un subset di nodi) da un file di caricamento in un'applicazione Oracle Enterprise Data Management Cloud.

I caricamenti in angolazioni vista consentono di caricare dati nelle angolazioni vista non associate, associate o parzialmente associate. Il file di caricamento dell'angolazione vista, un file CSV, Excel (XLSX) o un file ZIP contenente un file CSV o XLSX, deve essere disponibile nell'ambiente in cui stai caricando l'angolazione vista. È possibile caricare il file di caricamento nell'ambiente utilizzando il comando [uploadFile](#) o [copyFileFromInstance](#).

Si applica a

Oracle Enterprise Data Management Cloud

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate loadViewpoint VIEW VIEWPOINT PURPOSE FILE_NAME
[loadType=ReplaceNodes|Merge]
```

dove:

- `VIEW` è il nome di una vista Oracle Enterprise Data Management Cloud.
- `VIEWPOINT` è il nome dell'angolazione vista che si desidera caricare.
- `PURPOSE` è una stringa di testo, racchiusa tra virgolette doppie, che indica il motivo per cui viene caricata l'angolazione vista.
- `FILE_NAME` è il nome del file, con estensione, da cui caricare l'angolazione vista.
- `loadType`, facoltativamente, indica come caricare l'angolazione vista. I valori validi sono `Merge` e `ReplaceNodes`.
 - Utilizzare `Merge` per preservare le relazioni esistenti elaborando modifiche incrementalì.
 - Utilizzare `ReplaceNodes` per cancellare tutte le relazioni (incluse le relazioni orfane e quelle utilizzate da altre angolazioni vista con lo stesso set di gerarchie) dalla gerarchia diversa da quelle del file di caricamento. Questo è il tipo di caricamento predefinito.

Esempi

- **Unire modifiche incrementalì:** `epmautomate loadViewpoint USOperations Entity "Daily Upstream Load" data_Entity.CSV loadType=Merge`

- **Sostituire le gerarchie esistenti:** `epmautomate loadViewpoint USOperations Entity "Replace US Operations data" data_Entity.CSV`

login

Stabilisce una connessione sicura a un ambiente. Questo comando supporta l'accesso a un ambiente con una password in testo normale o un password file cifrato contenente la password o il token di aggiornamento OAuth 2.0. L'accesso tramite il token di aggiornamento OAuth 2.0 è supportato solo per gli ambienti OCI (Gen 2).

La sessione ha inizio dopo aver eseguito l'accesso e rimane attiva fino a quando l'utente non si scollega.

Nota:

- EPM Automate non supporta l'accesso tramite le credenziali SSO dell'organizzazione.
- EPM Automate non funziona con il proxy SOCKS; funziona solo con il proxy HTTP/HTTPS.
- Quando si utilizza questo comando in file batch per automatizzare le attività, Oracle consiglia di utilizzare password cifrate o il token di aggiornamento OAuth 2.0 per evitare di registrare password con testo in chiaro nei file batch.
- Nei computer Windows, questo comando identifica automaticamente il certificato di sicurezza intermedia mancante di un server proxy che può impedire di stabilire una connessione e lo aggiunge all'ambiente JRE installato in `C:\Oracle\EPM Automate`. In questo modo si evitano errori di accesso correlati ai certificati di sicurezza quando si utilizzano server proxy per accedere a Internet.
Nei computer Linux, il comando di accesso identifica il certificato di sicurezza mancante dal server proxy, lo scarica e visualizza un errore. Il certificato scaricato può quindi essere installato da un utente con accesso `root` nell'ambiente JRE disponibile nella posizione `JAVA_HOME` identificata nelle variabili di ambiente. Fare riferimento alle fonti di informazione seguenti.
 - [Java Runtime Environment e EPM Automate](#)
 - Documentazione Keytool Java

Durante la procedura di accesso viene visualizzato un messaggio che richiede di aggiornare EPM Automate se si sta utilizzando una versione non aggiornata. È possibile utilizzare il comando [upgrade](#) per eseguire l'aggiornamento in background dell'installazione.

Se si prevede di eseguire il comando [addUsers](#), [removeUsers](#), [assignRole](#) o [unassignRole](#), non effettuare l'accesso utilizzando il token di aggiornamento OAuth. Questi comandi richiedono l'utilizzo dell'autenticazione di base. Tutti gli altri comandi funzionano con OAuth 2.0 negli ambienti OCI (Gen 2).

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

Utilizzo

- **Utilizzo della password non cifrata:** `epmautomate login USERNAME PASSWORD URL [IDENTITYDOMAIN] [ProxyServerUserName=PROXY_USERNAME ProxyServerPassword=PROXY_PASSWORD ProxyServerDomain=PROXY_DOMAIN] [KeystorePassword=PASSWORD]`
- **Utilizzo del file cifrato:** `epmautomate login USERNAME PASSWORD_FILE URL [IDENTITYDOMAIN] [ProxyServerUserName=PROXY_USERNAME] [ProxyServerPassword=PROXY_PASSWORD] [ProxyServerDomain=PROXY_DOMAIN] [KeystorePassword=KEYSTORE_PASSWORD]`

In questi comandi:

- `USERNAME` è il nome utente dell'utente.
- `PASSWORD` è la password dell'utente.
- `PASSWORD_FILE` indica il nome e la posizione del file contenente la password cifrata o il token di aggiornamento OAuth 2.0 dell'utente. Fare riferimento al comando [encrypt](#).
- `URL` è l'URL dell'ambiente con cui stabilire la connessione. È possibile utilizzare un URL customizzato o di reindirizzamento a microsito anziché l'URL di Oracle Enterprise Performance Management Cloud. Per creare un URL di reindirizzamento per un ambiente, utilizzare uno strumento di abbreviazione dei collegamenti di terze parti, ad esempio, T.ly, Bitly, Rebrandly, TinyUrl, is.gd e così via. Per informazioni dettagliate sulla creazione di URL di reindirizzamento, fare riferimento alla documentazione dello strumento di abbreviazione di terze parti.

Nota:

Se si utilizza un gateway o un proxy inverso API, utilizzare il relativo URL e il contesto definito per l'ambiente in uso anziché l'URL di EPM Cloud.

- `IDENTITYDOMAIN`, facoltativamente, è il dominio di Identity dell'ambiente. Questo valore viene ricavato automaticamente dall'URL di EPM Cloud. Qualsiasi valore specificato personalmente verrà ignorato. Tuttavia, questo valore è obbligatorio se si utilizza l'URL di un gateway o di un proxy inverso API per la connessione a un ambiente EPM Cloud classico.
- `ProxyServerUserName` è il nome utente per l'autenticazione di una sessione sicura con il server proxy HTTP che controlla l'accesso a Internet. Specificare il nome utente senza anteporre come prefisso un nome di dominio. Obbligatorio solo se l'autenticazione a livello di server proxy è abilitata per la rete in uso.
- `ProxyServerPassword` è la password per l'autenticazione dell'utente con il server proxy. Obbligatorio solo se l'autenticazione a livello di server proxy è abilitata per

la rete in uso. Questa password può essere cifrata. Fare riferimento al comando [encrypt](#). Se la password è cifrata, viene letta da `PASSWORD_FILE`.

- `ProxyServerDomain` indica il nome del dominio definito per il server proxy HTTP (non il nome del server o il nome host del server proxy). Obbligatorio solo se l'autenticazione a livello di server proxy è abilitata per la rete in uso ed è configurato un dominio del server proxy.
- `KeystorePassword` (facoltativo) indica la password keystore necessaria per importare il certificato di sicurezza di un server proxy. Utilizzare questo parametro solo in Windows e solo se si verificano gli errori seguenti in ambienti in cui viene utilizzato un server proxy per incanalare l'accesso a Internet.

EPMAT-7: Impossibile connettersi poiché mancano nel keystore alcuni certificati SSL

EPMAT-7: Impossibile connettersi poiché mancano nel keystore i certificati SSL sopra menzionati

Nota:

EPM Automate rileva e utilizza le impostazioni proxy HTTP/HTTPS nel computer in uso.

EPM Automate supporta i meccanismi di autenticazione elencati di seguito per la connessione al server proxy.

- Autenticazione di base
- Autenticazione del digest
- Autenticazione Kerberos
- Autenticazione proxy con negoziazione
- Autenticazione NTLM

Il metodo di autenticazione disponibile e la relativa configurazione dipendono dal server proxy in uso.

Nei computer Linux, se le impostazioni proxy richiedono l'autenticazione con il server proxy, è necessario immettere il dominio, il nome utente e la password del server proxy come parametri di questo comando. Contattare l'amministratore della rete per ulteriori informazioni sul nome e sulle credenziali del dominio del server proxy.

Esempi

- Utilizzo di una password EPM Cloud non cifrata, nessuna autenticazione proxy:
`epmautomate login serviceAdmin P@ssword1 https://test-cloud-pln.pbcs.us1.oraclecloud.com`
- Utilizzo di un file cifrato, nessuna autenticazione proxy:
`epmautomate login serviceAdmin C:\mySecuredir\password.epw https://test-cloud-pln.pbcs.us1.oraclecloud.com`
- Utilizzo di un file cifrato, se è abilitata l'autenticazione a livello di server proxy con il dominio del server:

```
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
```

- **Utilizzo di un file cifrato, se è abilitata l'autenticazione a livello di server proxy senza un dominio del server:**
epmautomate login serviceAdmin C:\mySecuredir\password.epw https://
test-cloud-pln.pbc.s.us1.oraclecloud.com
ProxyServerUserName=john.doe@example.com ProxyServerPassword=example
- **Utilizzo della password cifrata di EPM Cloud e del server proxy, se è abilitata l'autenticazione nel server proxy con un dominio del server:**
epmautomate login serviceAdmin C:\mySecuredir\password.epw https://
test-cloud-pln.pbc.s.us1.oraclecloud.com
ProxyServerUserName=john.doe@example.com ProxyServerDomain=example
- **Utilizzo della password cifrata di EPM Cloud e del server proxy, se è abilitata l'autenticazione nel server proxy senza un dominio del server:**
epmautomate login serviceAdmin C:\mySecuredir\password.epw https://
test-cloud-pln.pbc.s.us1.oraclecloud.com
ProxyServerUserName=john.doe@example.com
- **Utilizzo di un file cifrato con il gateway API APIGEE:**
epmautomate login serviceAdmin C:\mySecuredir\password.epw https://
exampleapigee.apigee.com/epm example_ID_DOM

logout

Termina la connessione corrente a un ambiente.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

Utilizzo

```
epmautomate logout
```

Esempio

```
epmautomate logout
```

maskData

Maschera i dati dell'applicazione a garanzia della privacy dei dati. Utilizzare questo comando solo sugli ambienti di test per nascondere i dati riservati agli sviluppatori dell'applicazione.

AVVISO: non utilizzare questo comando negli ambienti di produzione perché randomizza i dati applicazione correnti, rendendoli non significativi. Non è possibile

annullare gli effetti di questo comando. Se i dati in un ambiente del servizio sono stati erroneamente mascherati, è necessario ripristinarli da una copia di backup oppure dallo snapshot di manutenzione.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate maskData [-f]` dove `-f` è un'opzione per forzare l'avvio del processo di mascheramento senza la conferma dell'utente. Se non si utilizza l'opzione `-f`, EPM Automate richiede di confermare l'azione.

Esempio

```
epmautomate maskData [-f]
```

mergeDataSlices

Unisce tutte le sezioni di dati incrementali di un cubo di memorizzazione di aggregazione nella sezione del database principale e rimuove facoltativamente le celle con valore zero.

Si applica a

Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate mergeDataSlices CUBE_NAME [keepZeroCells=true|false] dove:`

- `CUBE_NAME` identifica il cubo di memorizzazione di aggregazione per il quale devono essere unite tutte le sezioni di dati.
- `keepZeroCells` specifica facoltativamente se rimuovere le celle con valore zero (la cancellazione logica di dati da un'area restituisce una cella con valore zero). Il valore predefinito è `true`.

Esempio

```
epmautomate mergeDataSlices repl keepZeroCells=false
```

mergeSlices

Unisce sezioni di dati incrementali nel cubo di database principale e, facoltativamente, rimuove le celle Oracle Essbase contenenti 0 (zero) come valore per comprimere il cubo.

La rimozione delle celle contenenti 0 ottimizza le prestazioni del cubo.

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate mergeSlices applicationName [removeZeroCells=true|false] dove:`

- `applicationName` è il nome di un'applicazione Profitability and Cost Management.
- `removeZeroCells`, facoltativamente, specifica se rimuovere le celle contenenti 0. Il valore predefinito di questo parametro è `false`.

Esempi

- Unire le sezioni senza rimuovere le celle contenenti 0:
 - `epmautomate mergeSlices BksML30`
 - `epmautomate mergeSlices BksML30 removeZeroCells=false`
- Unire le sezioni e rimuovere le celle contenenti 0: `epmautomate mergeSlices BksML30 removeZeroCells=true`

optimizeASOCube

Ottimizza la performance delle query per la selezione di viste di aggregazione per l'estrazione di dati da cubi ASO.

Questo comando consente di eseguire operazioni di ottimizzazione delle query su cubi ASO nei casi in cui l'aggregazione predefinita è considerata insufficiente per soddisfare le esigenze di estrazione dati o reporting a causa delle elevate dimensioni dei dati. Il processo di ottimizzazione tipico è descritto di seguito.

- Eliminare le aggregazioni predefinite e basate su query.
- Avviare il tracciamento query.
- Eseguire query campione da Profitability and Cost Management Query Manager, Oracle Smart View for Office o Data Management e qualsiasi altra query MDX rappresentativa del tipo di query per cui è necessaria l'ottimizzazione per perfezionare Oracle Essbase .
- Creare l'aggregazione basata su query ottimizzate o predefinite.

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate optimizeASOCube APPLICATION_NAME OPTIMIZATION_TYPE`. I parametri sono descritti di seguito.

- `APPLICATION_NAME` indica il nome dell'applicazione Profitability and Cost Management a cui appartiene il cubo ASO.
- `OPTIMIZATION_TYPE` indica un'operazione di ottimizzazione del cubo. Di seguito sono riportati i valori consentiti.
 - `clearAggregations`, che rimuove le viste predefinite e basate su query.
 - `createAggregations`, che crea viste di aggregazione Essbase predefinite. Utilizzare questa opzione per eseguire l'aggregazione predefinita anziché quella basata su query.
 - `startQueryTracking`, che avvia il tracciamento query.
 - `stopQueryTracking`, che arresta il tracciamento query. Utilizzare questa opzione per impedire a Essbase di raccogliere informazioni di ottimizzazione. Essbase continua a raccogliere informazioni di ottimizzazione finché non si arresta il tracciamento query o Essbase. Essbase può aggregare le viste in base ai dati raccolti fino all'arresto del tracciamento query.
 - `createQBOAggregations`, che crea viste di aggregazione Essbase basate sulle query ottimizzate eseguite dopo l'abilitazione del tracciamento query.

Esempi

- Eliminazione delle viste di aggregazione predefinite e basate su query:
`epmautomate optimizeASOCube BksML12 clearAggregations`
- Avvio del tracciamento query:
`epmautomate optimizeASOCube BksML12 startQueryTracking`
- Creazione di viste di aggregazione Essbase basate sulle query ottimizzate eseguite dopo l'avvio del tracciamento query:
`epmautomate optimizeASOCube BksML12 createQBOAggregations`

programDocumentationReport

Crea un report della documentazione del programma contenente la logica dell'applicazione Profitability and Cost Management.

È possibile scaricare il report in un computer locale utilizzando il comando [downloadFile](#).

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

Utilizzo

```
epmautomate programDocumentationReport APPLICATION_NAME POV_NAME
[fileName=FILE_NAME] [fileType=PDF|WORD|EXCEL|HTML] [useAlias=true|false]
stringDelimiter="DELIMITER" dove:
```

- *APPLICATION_NAME* è il nome dell'applicazione Profitability and Cost Management per la quale viene creato il report della documentazione del programma.
- *POV_NAME* è il nome del POV del modello nell'applicazione per la quale viene generato il report.
- *fileName* (facoltativo) è un nome univoco (inclusa l'estensione) del file del report. Il nome predefinito del file del report è `HPCMMLProgramDocumentationReport_APPLICATION_NAME_POV_NAME.pdf`.
- *fileType* (facoltativo) è il formato del file di output. Il valore predefinito è `PDF`.
- *useAlias* (facoltativo) specifica se stampare gli alias anziché i nomi dei membri. Il valore predefinito è `false`.
- *stringDelimiter* è il delimitatore utilizzato nei valori POV. Il delimitatore deve essere racchiuso tra virgolette.

Esempio

```
epmautomate programDocumentationReport BksML30 2019_Feb_Actual
fileName=Feb-Actual.xls fileType=Excel useAlias=true stringDelimiter="_"
```

provisionReport

Genera un report assegnazioni ruoli (file .CSV) e lo memorizza nella posizione di scaricamento predefinita.

Il report elenca i ruoli predefiniti (ad esempio Utente avanzato di nome servizio) e i ruoli di applicazione (ad esempio Allocazione di massa, che è un ruolo dell'applicazione Planning) assegnati agli utenti. Utilizzare il comando [downloadFile](#) per scaricare il report.

È possibile generare due versioni del report: semplificata o classica. Nel report semplificato, che è identico al Report assegnazioni ruoli disponibile nella schermata Controllo accesso, non vengono elencati i ruoli applicazione inclusi nei ruoli predefiniti o i ruoli componente dei ruoli applicazione assegnati all'utente. Nella versione classica del report vengono elencati i ruoli del componente che vengono inclusi nei ruoli predefiniti a cui sono assegnati gli utenti. Vengono elencati inoltre i ruoli dell'applicazione assegnati all'utente (direttamente o mediante gruppi).

La generazione di questo report aggiorna le informazioni sull'utente e sul ruolo disponibili nel controllo accesso.

Solo per OCI (Gen 2): Oracle Enterprise Performance Management Cloud considera gli utenti disattivati identici agli utenti a cui non è stato assegnato alcun ruolo predefinito anche se tali utenti potrebbero aver avuto ruoli predefiniti al momento della disattivazione. Le informazioni relative agli utenti disattivati non sono incluse in questo report.

 **Nota:**

Questo comando non sarà più valido in una prossima release. Invece di questo comando, utilizzare il comando [roleAssignmentReport](#), che genera un report equivalente.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate provisionReport REPORT_NAME [format=classic|simplified]
[userType=serviceUsers|IDAdmins] dove:
```

- *REPORT_NAME* è il nome del report.
- *format*, facoltativamente, definisce il formato del report. Valori validi:
 - *simplified*, ovvero l'opzione predefinita, crea un report identico al Report assegnazioni ruoli generato nella schermata Controllo accesso.
 - *classic* crea un report in cui vengono elencati i ruoli del componente che vengono inclusi nei ruoli predefiniti a cui vengono assegnati gli utenti. Elenca inoltre i ruoli dell'applicazione assegnati all'utente (direttamente o mediante gruppi).
- *userType*, facoltativamente, identifica gli utenti da includere nel report. Se per questo parametro non viene specificato alcun valore, viene utilizzato il valore predefinito *serviceUsers*. Valori validi:
 - *serviceUsers* crea un report contenente le informazioni su tutti gli utenti funzionali (non sono inclusi gli amministratori del dominio di Identity se non sono assegnati a un ruolo predefinito che concede loro l'accesso all'applicazione)
 - *IDAdmins* crea un report in cui sono elencati solo gli utenti assegnati al ruolo Amministratore del dominio di Identity. Il report è identico sia nel formato classico che nel formato semplificato.

Esempi

- **Creare un report classico:** `epmautomate provisionReport myProvReport.CSV format=classic`
- **Creare un report semplificato:**
 - `epmautomate provisionReport myProvReport.CSV format=simplified`
 - `epmautomate provisionReport myProvReport.CSV userType=serviceUsers`
- **Creare solo un elenco di report (amministratori del dominio di Identity):**
 - `epmautomate provisionReport myProvReport.CSV userType=IDAdmins`

- `epmautomate provisionReport myProvReport.CSV userType=IDAdmins
format=classic`

purgeArchivedTmTransactions

Rimuove dall'applicazione Account Reconciliation le transazioni con corrispondenza archiviate.

Utilizzare periodicamente il comando [archiveTmTransactions](#) per archiviare le transazioni con corrispondenza meno recenti e quindi eseguire questo comando per rimuoverle da Account Reconciliation e garantire dimensioni ottimali dell'applicazione.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

`epmautomate purgeArchivedTMTransactions JobID=JOB_ID` dove JobID indica l'identificativo del job di archiviazione delle transazioni di Corrispondenza transazione eseguito per archiviare le transazioni con corrispondenza. Questo ID di job viene visualizzato nella console di EPM Automate quando si esegue il comando [archiveTmTransactions](#). È disponibile anche nella console job.

Esempio

```
epmautomate purgeArchivedTMTransactions JobID=100000002655003
```

purgeTmTransactions

Rimuove le transazioni con corrispondenza da Account Reconciliation.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

`epmautomate purgeTmTransactions matchType age [filterOperator=VALUE]
[filterValue=VALUE] [logFilename=FILE_NAME]`. I parametri sono descritti di seguito.

- `matchType` indica l'identificativo (TextID) del tipo di corrispondenza da cui devono essere eliminate le transazioni con corrispondenza.

- `age` indica da quanti giorni è stata eseguita la corrispondenza della transazione. Una transazione con corrispondenza che risale a un numero di giorni superiore o uguale a questo valore verrà eliminata.
- `filterOperator` (facoltativo) è una delle condizioni di filtro elencate di seguito per l'identificazione dei conti contenenti le transazioni con corrispondenza da eliminare. Questo valore viene utilizzato in combinazione con il valore `filterValue` per identificare i conti da cui devono essere rimosse le transazioni con corrispondenza:
 - `equals`
 - `not_equals`
 - `starts_with`
 - `ends_with`
 - `contains`
 - `not_contains`
- `filterValue` (facoltativo) indica un valore di filtro che identifica le transazioni da rimuovere. Se il valore di `filterOperator` è `equals` o `not_equals`, è possibile utilizzare un elenco separato da spazi per specificare più valori, ad esempio `filterValue=101-120 filterValue=102-202`. Se vengono specificati più valori, vengono selezionate per la rimozione le transazioni contenute in conti che soddisfano la combinazione operatore e valore di filtro.
- `logFilename` (facoltativo) indica il nome di un file di log per la registrazione di informazioni sull'attività dei comandi. Se non viene specificato un nome di file, viene generato automaticamente un file di log denominato `PurgeTransactions_JOB_ID`.

 **Note:**

Se non vengono specificati i valori `filterOperator` e `filterValue`, vengono rimosse tutte le transazioni con corrispondenza risalenti a un numero di giorni superiore o uguale al valore specificato per `age` da tutti i conti per il valore di `matchType` specificato.

Esempi

- Rimozione di transazioni con corrispondenza che risalgono ad almeno 180 giorni prima per il tipo di corrispondenza `cashrecon`:
`epmautomate purgeTMTransactions cashrecon 180 logFile=tmlogs.log`
- Rimozione di transazioni con corrispondenza che risalgono ad almeno 180 giorni prima per il tipo di corrispondenza `cashrecon` per il conto `101-120` or `102-202`:
`epmautomate purgeTMTransactions cashrecon 180 filterOperator>equals filterValue=101-120 FilterValue=102-202`
- Rimozione di transazioni con corrispondenza che risalgono ad almeno 180 giorni prima per il tipo di corrispondenza `cashrecon` per qualsiasi conto contenente la stringa `11`:
`epmautomate purgeTMTransactions cashrecon 180 filterOperator=contains filterValue=11`

recomputeOwnershipData

Ricalcola i dati di proprietà.

È necessario ricalcolare i dati di proprietà in Financial Consolidation and Close nelle situazioni riportate di seguito.

- Dopo l'aggiunta o l'eliminazione delle regole di sostituzione per i conti di Gestione proprietà.
- Dopo la modifica delle impostazioni dell'intervallo di Metodi consolidamento.
- Dopo l'aggiornamento del database, indipendentemente dall'eventuale modifica della struttura dell'entità.

È necessario ricalcolare i dati di proprietà in Tax Reporting dopo l'aggiornamento di ogni database, anche se la struttura dell'entità non è stata modificata.

Si applica a

Financial Consolidation and Close e Tax Reporting.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente

Utilizzo

`epmautomate recomputeOwnershipData Scenario Year Period` dove:

- `Scenario` è il nome dello scenario da ricalcolare.
- `Year` è l'anno da ricalcolare.
- `Period` è il primo periodo dell'anno da ricalcolare.
Vengono ricalcolati il periodo selezionato e tutti i periodi successivi.



Nota:

È possibile ricalcolare un POV che richiede il ricalcolo solo dopo il ricalcolo dei dati proprietà.

Esempio

`epmautomate recomputeOwnershipData FCCS_total_Actual FY19 Jan`

recreate

Ripristina lo stato originario di un ambiente ricreando la distribuzione.

Ricreare la distribuzione per completare i seguenti task:

- Eseguire il cleanup di un ambiente prima di importare uno snapshot completo.
- Modificare il processo aziendale distribuibile in un ambiente.

- Modificare la versione di Oracle Essbase in uso in ambienti Oracle Enterprise Performance Management Cloud diversi da Narrative Reporting, Oracle Enterprise Data Management Cloud e Account Reconciliation, che non utilizzano Essbase. Per impostazione predefinita, gli ambienti EPM Standard Cloud Service ed EPM Enterprise Cloud Service vengono distribuiti con istanze Essbase con abilitazione ibrida, mentre gli ambienti precedenti vengono distribuiti con istanze Essbase non ibride.

L'aggiornamento della distribuzione di istanze Essbase non ibride negli ambienti precedenti è necessario nelle situazioni riportate di seguito.

- Supporto della dimensionalità estesa in ambienti Financial Consolidation and Close precedenti esistenti.
- Abilitazione delle applicazioni BSO ibride di memorizzazione a blocchi in ambienti Enterprise Planning e Planning Modules precedenti.

È necessario eseguire il downgrade di Essbase con abilitazione ibrida negli ambienti EPM Enterprise Cloud Service se si importa uno snapshot da un ambiente con una versione Essbase senza abilitazione ibrida.

Per informazioni dettagliate su Hybrid Essbase e considerazioni sull'aggiornamento a Hybrid Essbase, fare riferimento alla sezione Informazioni su Essbase in EPM Cloud del manuale *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori*.

▲ **Attenzione:**

- Questo comando elimina l'applicazione esistente e, facoltativamente, tutti gli artifact definiti dall'utente dall'ambiente. Ricrea inoltre il database e rimuove tutti i dati esistenti. Dopo aver ricreato il servizio, è possibile creare un nuovo processo aziendale o importarne uno utilizzando Migrazione o EPM Automate.
- Questo comando elimina la cronologia della migrazione. Di conseguenza, il report sullo stato della migrazione disponibile in Migrazione non conterrà informazioni di cronologia.
- Prima di utilizzare questo comando, eseguire un backup completo dell'ambiente. È possibile creare uno snapshot di backup eseguendo il comando `runDailyMaintenance`.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate recreate [-f] [removeAll=true|false] [EssbaseChange=upgrade|downgrade] [TempServiceType=Service_type]`. I parametri sono descritti di seguito.

- `-f` forza il processo di ricreazione per l'avvio senza la conferma dell'utente. Se non si utilizza l'opzione `-f`, EPM Automate richiede di confermare l'azione.
- `removeAll` rimuove facoltativamente tutti gli snapshot e il contenuto delle cartelle in entrata (file caricati) e in uscita (file esportati dall'ambiente). Il valore predefinito è `false`, che mantiene gli snapshot e il contenuto delle cartelle in entrata e in uscita.
- `EssbaseChange` (facoltativo) esegue l'aggiornamento o il downgrade della versione corrente di Essbase.
Se non si specifica questa opzione, EPM Automate mantiene la versione di Essbase corrente. I valori consentiti sono i seguenti:
 - `upgrade` per passare da Essbase non ibrido a Essbase ibrido
 - `downgrade` per passare da Essbase ibrido a Essbase non ibrido.

▲ Attenzione:

Prima di utilizzare questa opzione, leggere e comprendere le informazioni disponibili nella sezione Informazioni su Essbase in EPM Cloud del manuale *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori*.

- `TempServiceType` (facoltativo) converte un ambiente in un ambiente di servizio diverso.
I processi aziendali che è possibile distribuire in un ambiente dipendono dalla sottoscrizione di cui si dispone. Ad esempio, chi è titolare di una sottoscrizione a EPM Standard Cloud Service non è autorizzato a creare un'applicazione in formato libero dopo aver convertito l'ambiente da Account Reconciliation a Planning. Chi è titolare di una sottoscrizione a EPM Enterprise Cloud Service è autorizzato a creare qualsiasi processo aziendale nell'ambiente, dopo aver modificato il tipo di servizio in modo appropriato. Fare riferimento alla sezione Informazioni sui nuovi servizi EPM Cloud nella *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori*

Il comportamento di questo parametro dipende dalla sottoscrizione.

– **Sottoscrizioni diverse da EPM Standard Cloud Service e EPM Enterprise Cloud Service:**

È possibile utilizzare l'opzione `TempServiceType` per convertire temporaneamente un ambiente Planning, Enterprise Planning, Tax Reporting o Financial Consolidation and Close in un ambiente Account Reconciliation, Oracle Enterprise Data Management Cloud o Profitability and Cost Management. Ad esempio, in caso di acquisto di un ambiente Planning, è possibile convertirlo in un ambiente Account Reconciliation eseguendo il seguente comando:

```
epmautomate recreate -f removeAll=true TempServiceType=ARCS
```

Dopo aver convertito un ambiente in Account Reconciliation, è possibile convertirlo in un ambiente Oracle Enterprise Data Management Cloud o Profitability and Cost Management utilizzando il valore di `TempServiceType`

appropriato. Ad esempio, per convertirlo in un ambiente Profitability and Cost Management, utilizzare il seguente comando:

```
epmautomate recreate -f removeAll=true TempServiceType=PCMCS
```

Per riconvertire l'ambiente nel tipo di servizio di origine, utilizzare il seguente comando:

```
epmautomate recreate -f
```

Profitability and Cost Management: è possibile convertire l'ambiente Profitability and Cost Management in un ambiente Planning, Enterprise Planning o Enterprise Profitability and Cost Management eseguendo questo comando:

```
epmautomate recreate -f removeAll=true TempServiceType=PBCS
```

Per convertire l'ambiente all'ambiente Profitability and Cost Management originale, utilizzare il seguente comando:

```
epmautomate recreate -f TempServiceType=PCMCS
```

 **Nota:**

Gli ambienti Profitability and Cost Management non possono essere convertiti in ambienti Account Reconciliation, Oracle Enterprise Data Management Cloud o Narrative Reporting.

- Sottoscrizioni di **EPM Standard Cloud Service e EPM Enterprise Cloud Service:** È possibile utilizzare l'opzione `TempServiceType` per convertire un ambiente EPM Cloud in un qualsiasi altro ambiente supportato.

Le sottoscrizioni di EPM Enterprise Cloud Service utilizzano una piattaforma EPM Cloud comune. Inizialmente è possibile distribuire qualsiasi processo aziendale EPM Cloud supportato.

Per passare da un processo aziendale distribuito a un altro, ricreare l'ambiente specificando il nuovo tipo di servizio per l'ambiente. Ad esempio, se è stato creato un processo aziendale Account Reconciliation, ma ora si desidera creare un ambiente Oracle Enterprise Data Management Cloud, sarà necessario eseguire il comando `recreate` procedendo come segue.

```
epmautomate recreate -f removeAll=true TempServiceType=EDMCS
```

Per convertire un processo aziendale, ad esempio di Account Reconciliation, in Planning, Tax Reporting o Financial Consolidation and Close, non specificare un valore per `TempServiceType`. Ad esempio, se è stato creato un processo aziendale Account Reconciliation ma ora si desidera creare un ambiente Moduli Planning, sarà necessario procedere come segue.

```
epmautomate recreate -f removeAll=true
```

Valori validi per l'opzione `TempServiceType`:

- ARCS: consente di convertire un ambiente in un ambiente Account Reconciliation

- EDMCS: consente di convertire un ambiente in un ambiente Oracle Enterprise Data Management Cloud
- EPRCS: consente di convertire un ambiente in un ambiente Narrative Reporting
- PCMCS per convertire un ambiente in un ambiente Profitability and Cost Management

Esempi

- Ricreare l'ambiente corrente, ripristinare il tipo di servizio di origine (se un'operazione di ricreazione è stata eseguita in precedenza con un parametro TempServiceType) e passare a Essbase abilitato per ibrido senza rimuovere contenuti e snapshot creati dagli utenti nelle cartelle di posta in entrata e in uscita:

```
epmautomate recreate -f EssbaseChange=upgrade
```
- Ricreare l'ambiente corrente, ripristinare il tipo di servizio di origine (se un'operazione di ricreazione è stata eseguita in precedenza con un parametro TempServiceType) e rimuovere contenuti e snapshot nelle cartelle di posta in entrata e in uscita:

```
epmautomate recreate -f removeAll=true
```
- Ricreare l'ambiente corrente come ambiente Oracle Enterprise Data Management Cloud e rimuovere il contenuto delle cartelle posta in entrata e in uscita e gli snapshot correnti:

```
epmautomate recreate -f removeAll=true TempServiceType=EDMCS
```
- Ricreare l'ambiente Account Reconciliation corrente di EPM Enterprise Cloud Service in un ambiente Financial Consolidation and Close e rimuovere il contenuto delle cartelle inbox e outbox e gli snapshot esistenti:

```
epmautomate recreate -f removeAll=true
```

refreshCube

Aggiorna il cubo dell'applicazione. Di norma, si aggiorna il cubo dopo l'importazione di metadati nell'applicazione.

Il tempo richiesto per completare un'operazione di aggiornamento del cubo dipende dalle modifiche apportate alla struttura dell'applicazione e dal relativo impatto sul cubo. Ad esempio, l'aggiornamento in seguito all'aggiornamento di un cubo di memorizzazione a blocchi sparso non richiede molto tempo, mentre l'aggiornamento di un cubo dopo l'aggiornamento di un membro di un cubo di memorizzazione a blocchi denso o di un membro di un cubi di memorizzazione di aggregazione potrebbe richiedere una quantità considerevole di tempo. È necessario assicurarsi che l'operazione di aggiornamento del cubo sia completata prima che venga eseguito il backup dell'applicazione durante la successiva finestra di manutenzione.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate refreshCube [JOB_NAME]` dove `JOB_NAME` (facoltativo) indica il nome di un job di aggiornamento del database definito nell'applicazione.

Lo stato dell'operazione è ripetuto nella console da cui viene eseguito il comando. È anche possibile visualizzare lo stato dalla pagina **Attività recente** della schermata **Job** nell'applicazione.

Esempio

```
epmautomate refreshCube DaliyCubeRefresh
```

removeUserFromGroups

Rimuove l'appartenenza di un utente dai gruppi di controllo dell'accesso definiti in un file CSV con codifica ANSI o UTF-8.

Il formato del file è il seguente:

```
Group Name
Group1
Group2
```



Nota:

Questi gruppi devono esistere nell'area Controllo accesso. Per i valori di Group Name non è prevista la distinzione tra maiuscole e minuscole.

Utilizzare il comando [uploadFile](#) per caricare il file in un ambiente.

Quando termina l'esecuzione del comando, EPM Automate invia in stampa alla console le informazioni su ogni singola voce non riuscita. Esaminare queste informazioni per comprendere il motivo per cui l'esecuzione del comando non è riuscita per alcune voci nel file CSV.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi o Gestione controllo accesso

Utilizzo

```
epmautomate removeUserFromGroups FILE_NAME User_Login dove:
```

- `FILE_NAME` è il nome di un file CSV contenente i nomi dei gruppi di controllo dell'accesso da cui deve essere rimossa l'appartenenza dell'utente

- `User_Login` indica l'ID di accesso di un utente di Oracle Enterprise Performance Management Cloud la cui appartenenza deve essere rimossa dai gruppi Controllo accesso. Questo ID di accesso dell'utente deve esistere nel dominio di Identity che gestisce l'ambiente e deve disporre di un ruolo predefinito assegnato. Per questo valore non è prevista la distinzione tra maiuscole e minuscole.

Esempio

```
epmautomate removeUserFromGroups groups.CSV jdoe@examle.com
```

removeUsers

Elimina gli account del dominio di Identity identificati in un file CSV con codifica ANSI o UTF-8 caricato nell'ambiente.

Il formato del file è il seguente:

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

Utilizzare il comando [uploadFile](#) per aggiornare il file nell'ambiente. Per i valori di User Login non è prevista la distinzione tra maiuscole e minuscole. Ad esempio, il valore `jane.doe@example.com` viene considerato uguale a `Jane.Doe@Example.com` o a qualsiasi variante di maiuscole e minuscole.



Nota:

- Il file CSV non deve includere l'account dell'utente che esegue il comando.
- Poiché gli account utente sono comuni a tutti gli ambienti di servizi supportati da un amministratore del dominio di Identity, l'eliminazione di un account per un ambiente ne determina l'eliminazione per tutti gli ambienti che condividono l'amministratore del dominio di Identity.

Quando termina l'esecuzione del comando, EPM Automate invia in stampa alla console le informazioni su ogni singola voce non riuscita. Esaminare queste informazioni per comprendere il motivo per cui l'esecuzione del comando non è riuscita per alcune voci nel file CSV.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore del dominio di Identity e qualsiasi ruolo predefinito (Amministratore servizi, Utente avanzato, Utente o Visualizzatore)

Utilizzo

`epmautomate removeUsers FILE_NAME` dove *FILE_NAME* è il nome di un file CSV contenente gli ID di accesso degli utenti da rimuovere dal dominio di Identity.

Esempio

```
epmautomate removeUsers remove_users.CSV
```

removeUsersFromGroup

Rimuove gli utenti elencati in un file CSV con codifica ANSI o UTF-8 da un gruppo gestito in Controllo accesso.

Il formato del file è il seguente:

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

Per i valori di User Login non è prevista la distinzione tra maiuscole e minuscole. Ad esempio, il valore `jane.doe@example.com` viene considerato uguale a `Jane.Doe@Example.com` o a qualsiasi variante di maiuscole e minuscole. Utilizzare il comando [uploadFile](#) per caricare il file contenente gli accessi utente nell'ambiente.

Quando termina l'esecuzione del comando, EPM Automate invia in stampa alla console le informazioni su ogni singola voce non riuscita. Esaminare queste informazioni per comprendere il motivo per cui l'esecuzione del comando non è riuscita per alcune voci nel file CSV.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi o Gestione controllo accesso

Utilizzo

```
epmautomate removeUsersFromGroup FILE_NAME GROUP_NAME dove:
```

- *FILE_NAME* rappresenta il nome di un file CSV contenente i nomi di accesso degli utenti che si desidera rimuovere da un gruppo gestito in Controllo accesso.
- *GROUP_NAME* è il nome del gruppo Controllo accesso da cui si desidera rimuovere gli utenti. Per questo valore non è prevista la distinzione tra maiuscole e minuscole.

 **Nota:**

L'utente viene rimosso da un gruppo solo se vengono soddisfatte entrambe le seguenti condizioni:

- Gli accessi utente inclusi nel file esistono nel dominio di Identity che gestisce l'ambiente.
- L'utente è assegnato a un ruolo predefinito nel dominio di Identity

Esempio

```
epmautomate removeUsersFromGroup user_file.CSV example_group
```

removeUsersFromTeam

Rimuove da un team gli utenti di Oracle Enterprise Performance Management Cloud elencati in un file CSV.

Se un utente incluso nel file CSV non è membro del team, questo comando lo ignora. Per i valori in questo file non è prevista la distinzione tra maiuscole e minuscole. Il formato del file CSV è il seguente:

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

Utilizzare il comando [uploadFile](#) per caricare il file .CSV nell'ambiente.

Si applica a

Financial Consolidation and Close, Tax Reporting e Account Reconciliation.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore
È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

```
epmautomate removeUsersFromTeam FILE.CSV TEAM_NAME dove:
```

- `FILE` definisce un file CSV in formato UTF-8 in cui sono elencati gli ID di accesso degli utenti da rimuovere dal team.
- `TEAM_NAME` identifica un nome di team definito in Controllo accesso. Per questo valore non è prevista la distinzione tra maiuscole e minuscole.

Esempio

```
epmautomate removeUsersFromTeam example_users.csv example_team
```

renameSnapshot

Rinomina uno snapshot caricato o creato in un ambiente.

Se questo comando viene eseguito per rinominare uno snapshot in fase di generazione o archiviazione, verrà visualizzato uno degli errori descritti di seguito.

- `File not found` se lo snapshot è in fase di generazione
- Processo di archiviazione in corso. Impossibile rinominare o eliminare se lo snapshot è in fase di archiviazione

Non rinominare lo snapshot di manutenzione in un ambiente. Per conservare un backup dello snapshot di manutenzione, è necessario scaricare lo `snapshot artifact` dall'ambiente in un computer locale e quindi rinominarlo. Fare riferimento alla sezione Panoramica dello snapshot di manutenzione nella *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori*.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato assegnati al ruolo applicazione Amministratore migrazione

Utilizzo

`epmautomate renameSnapshot SNAPSHOT_NAME NEW_SNAPSHOT_NAME dove:`

- `SNAPSHOT_NAME` rappresenta il nome di uno snapshot esistente. Questo valore non deve contenere caratteri speciali quali, ad esempio, spazi, \ (barra rovesciata), / (barra), * (asterisco), ? (punto interrogativo), " (punto esclamativo), < (minore) e > (maggiore).
- `NEW_SNAPSHOT_NAME` rappresenta il nome univoco che si desidera assegnare allo snapshot.

Esempio

```
epmautomate renameSnapshot "Example Snapshot" Example_Snapshot_18_09_25
```

replay

Ripete il caricamento di Oracle Smart View for Office, dell'API REST o di EPM Automate in un ambiente per consentire l'esecuzione del test delle performance in caso di carichi di particolare entità al fine di verificare che l'esperienza utente sia accettabile quando il servizio deve gestire un carico specifico.

È necessario creare un file di ripetizione che identifica le attività che devono essere eseguite sul servizio. Fare riferimento alla sezione [Preparazione per l'esecuzione del comando di ripetizione](#) per informazioni dettagliate su come creare il file di ripetizione.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

Utilizzo

```
epmautomate replay REPLAY_FILE_NAME.csv [duration=N] [trace=true]
[lagTime=t] [encrypt=true|false]. I parametri sono descritti di seguito.
```

- *REPLAY_FILE_NAME* è un file CSV in cui sono memorizzate le attività da eseguire nell'ambiente.
- *Duration* è un parametro facoltativo che indica il numero di minuti che definiscono la durata dell'esecuzione delle attività nell'ambiente. Le attività nel file HAR vengono eseguite una volta se questo valore non viene impostato. Se le attività nel file HAR sono completate nel tempo specificato da questo parametro, EPM Automate riesegue il file HAR fino al completamento delle attività. Ad esempio, si supponga di aver impostato il parametro *duration=10* per la ripetizione di un file HAR la cui esecuzione richiede tre minuti. In questo scenario, il comando di ripetizione esegue le attività del file HAR quattro volte (per una durata di 12 minuti) fino al completamento della quarta iterazione.
- *trace=true* è un'impostazione facoltativa che indica a EPM Automate di creare file di traccia in formato XML. Se questa impostazione facoltativa viene specificata, EPM Automate crea una cartella per ogni file HAR incluso nel file CSV di ripetizione e memorizza tutti i file di traccia correlati in tale cartella. Per ogni attività inclusa nel file HAR, EPM Automate genera un file di traccia contenente la risposta di Smart View. I file di traccia vengono denominati *trace-N.xml*; ad esempio *trace-1.xml*, dove *N* rappresenta un contatore che inizia da 1.

Le cartelle in cui sono memorizzati i file di traccia vengono create nella directory di esecuzione di EPM Automate. Per denominare le cartelle, EPM Automate utilizza una combinazione di ora di sistema corrente e nome file HAR nel formato *YYYY_MM_DD_HH_MM_SS_HAR_FILE_NAME*. Ad esempio, se il nome del file HAR è *forecast1.har*, il nome della cartella può essere *2016_06_08_10_21_42_forecast1*.

- *[lagTime=*t*]* (facoltativo) specifica il numero di secondi che il comando deve attendere tra l'esecuzione di ciascun file HAR incluso nel file di ripetizione. Il valore predefinito è 5 secondi. Il comando visualizza un errore se si specifica un valore inferiore a 5 secondi. I numeri negativi (ad esempio -1) e le frazioni (ad esempio 1/2) non sono accettabili come valore del parametro. I valori decimali sono supportati.

Dopo aver avviato l'esecuzione del primo file HAR, il comando attende il numero di secondi specificato da questo parametro per avviare l'elaborazione del file HAR successivo. Poiché le attività dell'utente solitamente non vengono avviate contemporaneamente, l'impostazione di questo parametro consente di creare una simulazione più realistica del carico in un ambiente.

Si supponga, ad esempio, di voler simulare il carico di 1000 utenti che accedono a un ambiente durante l'ora di punta per eseguire delle attività. È possibile creare file HAR per simulare queste sessioni e quindi eseguire questo comando con un tempo di ritardo di 6 secondi per replicare il carico esercitato sull'ambiente.

- `encrypt=true|false` consente di specificare in modo facoltativo se cifrare tutte le password incluse nel file di ripetizione. Il valore predefinito è `true`. Per cifrare la password viene utilizzata una chiave di cifratura casuale.

Fare riferimento alla sezione [Sessione di ripetizione di esempio](#) per la procedura dettagliata relativa all'esecuzione di questo comando.

Esempio

```
epmautomate replay forecast1.CSV duration=60 lagTime=5.6
```

resetService

Riavvia l'ambiente. Se lo si desidera, è possibile ottimizzare automaticamente l'ambiente prima di riavviarlo in modo da garantire l'ottimizzazione delle cache indici di Oracle Essbase per i cubi BSO (Block Storage Option, opzione di memorizzazione a blocchi) per l'applicazione in uso.

Per impostazione predefinita, gli ambienti vengono riavviati subito dopo il completamento della manutenzione giornaliera. L'ottimizzazione dell'ambiente è un processo importante, ad esempio dopo l'importazione di uno snapshot in un ambiente. Utilizzare questo comando solo se si nota un calo netto delle performance o se si ricevono messaggi di errore che indicano che non è possibile utilizzare l'ambiente. Il riavvio di un ambiente non influenza le customizzazioni dell'applicazione, ad esempio la modifica delle impostazioni nazionali, le impostazioni associate al tema e alla valuta e così via. Il riavvio richiede fino a 15 minuti.

Prima di utilizzare questo comando, assicurarsi che nell'ambiente non siano in esecuzione regole business.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate resetService "comment" [AutoTune=true|false] [-f] dove:
```

- `Comment` è la descrizione del problema che ha comportato il ripristino dell'ambiente. I commenti devono essere racchiusi tra virgolette.
- `AutoTune` (facoltativo) definisce se ottimizzare automaticamente l'ambiente in modo da ottimizzare le cache di Essbase per i cubi BSO dell'applicazione in uso. Il valore predefinito è `false`.

Utilizzare questo parametro solo negli ambienti in cui vengono utilizzati cubi BSO di Essbase: Planning (compreso Planning Modules), Financial Consolidation and Close e Tax Reporting.

- `-f`, facoltativamente, specifica che si desidera forzare il riavvio dell'ambiente senza ulteriori interazioni utente. Se non si utilizza questa opzione, EPM Automate richiede di confermare l'azione. Questa opzione risulta utile se si programma uno script che utilizza questo comando.

Esempi

- `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

Copia uno snapshot di backup disponibile di un ambiente OCI (Gen 2) in modo che possa essere importato nell'ambiente. Questo comando non funziona negli ambienti Oracle Enterprise Performance Management Cloud Classic.

Utilizzare il comando [listBackups](#) per determinare se il backup che si desidera ripristinare è disponibile. Il ripristino self-service di uno snapshot nell'ambiente consente di ridurre il tempo di elaborazione. Dopo aver ripristinato lo snapshot, utilizzare il comando [importSnapshot](#) per importarlo nell'ambiente.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato assegnati al ruolo applicazione
Amministratore migrazione

Utilizzo

`epmAutomate restoreBackup SNAPSHOT_NAME [targetName=TARGET_SNAPSHOT_NAME]`.
I parametri sono descritti di seguito.

- `SNAPSHOT_NAME` indica il nome di uno snapshot di backup disponibile nell'ambiente come elencato dal comando [listBackups](#).
- `targetName` (facoltativo) indica il nome dello snapshot di backup, senza estensione, nell'ambiente target. Se non si specifica questo valore, lo snapshot di backup viene ripristinato nell'ambiente target utilizzando il valore di `SNAPSHOT_NAME`, ma con un carattere di sottolineatura (`_`) invece della barra (`/`). Se ad esempio il valore di `SNAPSHOT_NAME` è `2022-05-14T00:08:56/Artifact_Snapshot.zip` il valore target di `targetName` sarà `2022-05-14T00:08:56_Artifact_Snapshot.zip`.

Esempi

- Servizi diversi da Narrative Reporting:

```
epmAutomate restoreBackup 2022-05-14T00:08:56/Artifact_Snapshot.zip
targetName=example_Artifact_Snapshot
```

- **Solo Narrative Reporting:**
epmAutomate restoreBackup 2022-02-16T21:00:02/EPRCS_Backup
targetName=Example_EPRCS_Backup

restructureCube

Consente di eseguire una ristrutturazione completa di un cubo di memorizzazione a blocchi per eliminare o ridurre la frammentazione. La ristrutturazione comporta anche la rimozione dei blocchi vuoti e non estende le eventuali modifiche dall'applicazione al cubo.



Nota:

Prima di eseguire questo comando, accertarsi che nessuno stia usando l'applicazione.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

epmautomate restructureCube *CUBE_NAME* dove *CUBE_NAME* è il nome di un cubo come in un'applicazione

Esempio

```
epmautomate restructureCube Plan1
```

roleAssignmentAuditReport

Negli ambienti OCI (Gen 2), crea un report di audit che elenca le modifiche apportate ad assegnazioni di ruoli predefiniti e applicazione su un determinato periodo di tempo corrispondente al periodo di conservazione audit specificato per l'ambiente. Il periodo di conservazione predefinito è di 30 giorni. È possibile estenderlo a un massimo di 90 giorni modificando l'impostazione di **Periodo di conservazione audit (giorni)** nella console di Oracle Cloud Identity. Per conservare i dati di audit per un periodo superiore a 90 giorni, scaricare e archiviare periodicamente questo report e il [report degli accessi non validi](#).

Il report di audit di assegnazioni ruoli elenca il nome accesso utente per cui è stata effettuata una modifica di ruolo (nella colonna Azione). Include inoltre il ruolo assegnato o per cui è stata annullata l'assegnazione, l'utente che ha effettuato la modifica del ruolo (colonna Amministratore) e l'indicatore orario (UTC) in formato 24 ore di quando è stata completata l'azione.

	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

Le informazioni sugli utenti eliminati precedentemente assegnati a ruoli predefiniti nell'ambiente sono elencate con il nome visualizzato (nome e cognome) dell'utente nella colonna Nome utente. In questi casi, la colonna Ruolo indica il ruolo predefinito di cui disponeva l'utente prima dell'eliminazione dell'account. Questa modifica non si applica agli eventuali ruoli applicazione assegnati all'utente eliminato. Queste assegnazioni vengono mostrate con il nome di accesso dell'utente. Per un esempio, fare riferimento alle informazioni racchiuse nel riquadro in rosso nella figura riportata di seguito.

	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

Si applica a

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, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi o amministratore del dominio di Identity e qualsiasi ruolo predefinito (amministratore servizi, utente avanzato, utente o visualizzatore)

Utilizzo

`epmAutomate roleAssignmentAuditReport FROM_DATE TO_DATE FILE_NAME.CSV.` I parametri sono descritti di seguito.

- *FROM_DATE* indica la data di inizio del periodo per cui deve essere generato il report (in formato YYYY-MM-DD). Questa data deve rientrare nel periodo di conservazione dell'audit specificato nella console di Oracle Cloud Identity.
- *TO_DATE* indica la data di fine del periodo per cui deve essere generato il report (in formato YYYY-MM-DD).
- *FILE_NAME* indica il nome di un file CSV per il report. È possibile scaricare il report generato utilizzando il comando [downloadFile](#).

Esempio

```
epmAutomate roleAssignmentAuditReport 2021-06-01 2021-07-30
RoleAuditReport.CSV
```

roleAssignmentReport

Genera un report assegnazione ruoli (.CSV).

In questo report vengono elencati i ruoli predefiniti, ad esempio Amministratore servizi, e i ruoli applicazione, ad esempio Assegnatario proprietà approvazioni, Supervisore approvazioni, Amministratore approvazioni e Designer processo approvazioni, che sono ruoli applicazione di Planning, assegnati agli utenti. È possibile generare questo report anche per elencare gli amministratori del dominio di Identity dell'ambiente. Questo report corrisponde alla versione CSV del Report assegnazioni ruoli generato da Controllo accesso.

	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

La generazione di questo report aggiorna le informazioni sull'utente e sul ruolo disponibili nel controllo accesso.

Solo per OCI (Gen 2): Oracle Enterprise Performance Management Cloud considera gli utenti disattivati identici agli utenti a cui non è stato assegnato alcun ruolo predefinito anche se tali utenti potrebbero aver avuto ruoli predefiniti al momento della disattivazione. Le informazioni relative agli utenti disattivati non sono incluse in questo report.



Nota:

Questo comando genera un report equivalente a quello creato con il comando [provisionReport](#).

È possibile scaricare il report utilizzando il comando [downloadFile](#).

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi o Gestione controllo accesso

Utilizzo

```
epmautomate roleAssignmentReport REPORT_NAME.CSV [userType=IDAdmins|
serviceUsers] dove:
```

- `REPORT_NAME` è il nome del report.
- `userType`, facoltativo, identifica il tipo di utenti le cui informazioni devono essere incluse nel report. Il valore predefinito è `serviceUsers`. I valori validi sono:
 - `serviceUsers` crea un report contenente le informazioni su tutti gli utenti funzionali (non sono inclusi gli amministratori del dominio di Identity se non sono assegnati a un ruolo predefinito che concede loro l'accesso all'applicazione).
 - `IDAdmins` crea un report in cui sono elencati solo gli utenti assegnati al ruolo Amministratore del dominio di Identity.

Esempi

- Generare il report che elenca solo utenti funzionali:
 - `epmautomate roleAssignmentReport myReport.CSV`
 - `epmautomate roleAssignmentReport myReport.CSV userType=serviceUsers`
- Generare il report che elenca solo gli amministratori del dominio di Identity:
 - `epmautomate roleAssignmentReport myReport.CSV userType=IDAdmins`

runAutomatch

Esegue il processo di abbinamento automatico per trovare le corrispondenze tra transazioni utilizzando le regole definite da un amministratore del servizio.



Nota:

Eeguire questo comando dopo aver importato i dati delle transazioni in Corrispondenza transazione utilizzando il comando [importTmPremappedTransactions](#) o [runDataRule](#).

È possibile monitorare lo stato del processo di abbinamento automatico nella scheda **Cronologia job** in Account Reconciliation.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore
È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

`epmautomate runAutomatch RECONCILIATION_TYPE dove RECONCILIATION_TYPE è un tipo di riconciliazione definito in Account Reconciliation.`

Esempio

`epmautomate runAutomatch INTERCOMPANY`

runBatch

Esegue un batch di Gestione dati.

Se la modalità di esecuzione batch in Gestione dati è impostata su Seriale, il controllo viene restituito quanto vengono completati tutti i job nel batch; è impostata su Parallelo, il controllo viene restituito quanto vengono sottomessi per l'esecuzione tutti i job nel batch.



Nota:

Questo comando non può essere utilizzato per eseguire l'integrazione del caricamento di dati diretto da origini dati in Oracle Enterprise Performance Management Cloud. Utilizzare l'agente di integrazione EPM per integrare i caricamenti di dati diretti. Per informazioni dettagliate, fare riferimento alla sezione [Esecuzione di un caricamento dati diretto tramite l'agente di integrazione EPM](#) in *Amministrazione di Integrazione dati per Oracle Enterprise Performance Management Cloud*.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

`epmautomate runBatch BATCH_NAME dove BATCH_NAME` è il nome di un batch definito in Gestione dati.

Esempi

```
epmautomate runBatch Accounting_batch
```

runBusinessRule

Avvia una regola business.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato (se è concesso l'accesso a esecuzione regola)

Utilizzo

```
epmautomate runBusinessRule RULE_NAME [PARAMETER=VALUE] dove:
```

- *RULE_NAME* è il nome di una regola business così come è definito nell'ambiente.
- *PARAMETER=VALUE* indica i parametri runtime facoltativi e i relativi valori necessari per l'esecuzione della regola business.

 **Nota:**

- Questo comando può solo eseguire una singola regola aziendale. Per eseguire un set di regole, utilizzare il comando `runRuleSet`.
- La regola viene eseguita per il tipo di piano nel quale è stata distribuita.
- Vengono utilizzati i valori predefiniti se non si specifica alcun valore per i parametri runtime. Il comando ignora i prompt runtime non corrispondenti a quelli definiti nella regola.
- Utilizzare la combinazione *PARAMETER=VALUE* per specificare tutti i prompt runtime richiesti dalla regola business. Nell'esempio seguente vengono utilizzati due prompt runtime (*Period* ed *Entity*) e i relativi valori (*Q1* e *USA*). Per informazioni sull'immissione di più valori per un parametro, fare riferimento alla sezione [Definizione di valori multipli per un parametro](#).

Esempio

```
epmautomate runBusinessRule RollupUSSales Period=Q1 Entity=USA
```

runCalc

Esegue i calcoli in un'applicazione.

Utilizzando questo comando è possibile eseguire calcoli utilizzando regole in un POV modello in base ai dati di un POV dati diverso senza dover copiare le regole tra POV.

Si applica a

Profitability and Cost Management

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

```
epmautomate runCalc APPLICATION_NAME POV_NAME [DATA_POV_NAME]
PARAMETER=VALUE [comment="comment"] stringDelimiter="DELIMITER". I parametri
sono descritti di seguito.
```

- *APPLICATION_NAME* è il nome dell'applicazione Profitability and Cost Management contenente il POV da calcolare.
- *POV_NAME* è il nome del POV modello da calcolare.

- *DATA_POV_NAME* (facoltativo) è il nome del POV dati da calcolare utilizzando le regole del POV modello.
Se *DATA_POV_NAME* non viene specificato, per impostazione predefinita verrà utilizzato *POV_NAME*.
È possibile utilizzare solo *exeType=ALL_RULES* se si specifica *DATA_POV_NAME*.
- *PARAMETER=VALUE* indica i parametri runtime e i relativi valori per l'esecuzione del calcolo. Specificare il numero di parametri e coppie di valori richiesto dal processo. I parametri validi e i relativi valori sono indicati di seguito.
 - *exeType=ALL_RULES|RULESET_SUBSET|SINGLE_RULE* identifica il tipo di esecuzione della regola. Questo parametro è obbligatorio.
A seconda del valore impostato per *exeType*, è necessario specificare i parametri indicati di seguito.
 - * Se si specifica *exeType=ALL_RULES*, non includere parametri del subset di regole o correlati a una singola regola come *subsetStart*, *subsetEnd*, *ruleSetName* e *ruleName*. È necessario utilizzare *exeType* se si imposta il parametro *DATA_POV_NAME*.
 - * Se si specifica *exeType=SINGLE_RULE*, specificare solo i valori per *ruleSetName* e *ruleName*.
 - * Se si specifica *exeType=RULESET_SUBSET*, specificare i valori per *subsetStart* e *subsetEnd*.
 - *subsetStart* specifica il numero di sequenza della prima regola nel set di regole da eseguire
 - *subsetEnd* specifica il numero di sequenza dell'ultima regola nel set di regole da eseguire
 - *ruleSetName* identifica il set di regole contenente i calcoli che si desidera eseguire
 - *ruleName* indica il nome della regole da eseguire (per eseguire un'unica regola)
 - *isClearCalculated=true|false* specifica se cancellare i calcoli esistenti
 - *isExecuteCalculations=true|false* specifica se eseguire i calcoli
 - *isRunNow=true|false* impostare questo valore su *true* per eseguire ora il processo
 - *optimizeReporting=true|false* impostare questo valore facoltativo su *false* se i calcoli devono essere eseguiti senza l'ottimizzazione per il reporting. Il valore predefinito è *true*.
Procedura consigliata:
 - * Impostare *optimizeReporting=false* solo quando è necessario per risparmiare tempo di elaborazione, ad esempio durante l'esecuzione di una singola regola o di una serie sequenziale di diversi POV.
 - * Durante l'esecuzione di più job di calcolo concorrenti, impostare *optimizeReporting=true* per tutti i job. Solo l'ultimo job da completare eseguirà l'aggregazione, evitando un'elaborazione ridondante e impedendo il rallentamento dei job in esecuzione.

 **Nota:**

I valori dei parametri (`true` o `false`) devono essere in minuscolo.

- `comment` è un commento facoltativo racchiuso tra virgolette
- `stringDelimiter` è il delimitatore utilizzato nei valori POV. Il delimitatore deve essere racchiuso tra virgolette.

Esempio

```
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

Genera un report definito in Conformità riconciliazione.

Fare riferimento alle seguenti fonti di informazione disponibili in *Amministrazione di Account Reconciliation*:

- Per istruzioni sulla definizione dei report, fare riferimento alla sezione Utilizzo dei report.
- Per un elenco di report predefiniti di Conformità riconciliazione e dei parametri per generarli, fare riferimento alla sezione Generazione di report predefiniti in Conformità riconciliazione.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

```
epmautomate runComplianceReport FILE_NAME GROUP_NAME REPORT_NAME
REPORT_FORMAT [Param=value] dove:
```

- `FILE_NAME` è un nome di file univoco per il report che verrà generato. Se sul server è presente un report con lo stesso nome, esso verrà sovrascritto. Utilizzare il comando [downloadFile](#) per scaricare il report in un computer locale.
- `GROUP_NAME` è il nome del gruppo associato al report.
- `REPORT_NAME` è il nome univoco del report da generare.
- `REPORT_FORMAT` è uno dei seguenti formati del report:
 - PDF
 - HTML (non supportato per grafici e diagrammi)

- XLSX (non supportato per i grafici)
- CSV
- CSV2

 **Nota:**

REPORT_FORMAT CSV non consente la formattazione di dati in base a un template mentre CSV2 sì. La generazione di report CSV2 formattati richiede più tempo rispetto all'output CSV.

- `Param=value`, facoltativamente, identifica i parametri obbligatori per la generazione del report. Ad esempio, il report Saldo per tipo di conto richiede due parametri `Period` con valore `July 2017` e il parametro `Currency Bucket` con il valore `Entered`. Specificare questi parametri nel modo indicato di seguito: `"Period=July 2017" "Currency Bucket=Entered"`.

Esempio

```
epmautomate runComplianceReport "Example_File Name""Reconciliation Manager"
"Balance By Account Type" PDF "Period=July 2017" "Currency Bucket=Entered"
```

runDailyMaintenance

Avvia immediatamente il processo di manutenzione giornaliera del servizio anziché attendere la finestra di manutenzione giornaliera programmata.

Questo comando consente di forzare la creazione di uno snapshot di backup e di aggiornare l'ambiente. Prima di eseguire questo comando, accertarsi che nessuno stia usando l'ambiente. Il programma di manutenzione giornaliero non viene modificato da questo comando. Utilizzare questo comando se non si desidera attendere la successiva finestra di manutenzione per rendere valide le modifiche apportate all'ambiente, ad esempio dopo aver applicato una patch singola.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate runDailyMaintenance [skipNext=true|false] [-f] dove:
```

- `skipNext`, facoltativamente, indica se ignorare l'occorrenza successiva del processo di manutenzione giornaliera. Il valore predefinito è `false`.
- `-f`, facoltativamente, indica se forzare l'avvio del processo di manutenzione senza la conferma dell'utente. Se non si utilizza l'opzione `-f`, EPM Automate richiede di confermare l'azione.

Esempi

- Per forzare l'avvio di una manutenzione giornaliera non inclusa nel ciclo senza ignorare la successiva manutenzione programmata: `epmautomate runDailyMaintenance -f`
- Per forzare l'avvio di una manutenzione giornaliera non inclusa nel ciclo e ignorare la successiva manutenzione programmata: `epmautomate runDailyMaintenance skipNext=true -f`
- Per avviare una manutenzione giornaliera non inclusa nel ciclo e ignorare la successiva manutenzione programmata: `epmautomate runDailyMaintenance skipNext=true -f`

runDataRule

Esegue una regola di caricamento dati di Gestione dati in base al periodo di inizio e a quello di fine, nonché in base alle opzioni di importazione o esportazione che si specificano.

Nota:

Questo comando non può essere utilizzato per eseguire l'integrazione del caricamento di dati diretto da origini dati in Oracle Enterprise Performance Management Cloud. Utilizzare l'agente di integrazione EPM per integrare i caricamenti di dati diretti. Per informazioni dettagliate, fare riferimento alla sezione [Esecuzione di un caricamento dati diretto tramite l'agente di integrazione EPM](#) in *Amministrazione di Integrazione dati per Oracle Enterprise Performance Management Cloud*.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

```
epmautomate runDataRule RULE_NAME START_PERIOD END_PERIOD IMPORT_MODE
EXPORT_MODE [FILE_NAME] dove:
```

- *RULE_NAME* è un nome di una regola di caricamento dati definita in Gestione dati. Se il nome regola contiene spazi, racchiuderlo tra virgolette.
- *START_PERIOD* è il primo periodo per il quale caricare i dati. Questo periodo di tempo deve essere definito nel mapping dei periodi in Gestione dati.
- *END_PERIOD* e, nel caso di un caricamento dati per più periodi, l'ultimo periodo per il quale caricare i dati. Per un caricamento in un periodo singolo, immettere lo stesso periodo come periodo di inizio. Questo periodo di tempo deve essere definito nel mapping dei periodi in Gestione dati.

- *IMPORT_MODE* determina in che modo importare i dati in Gestione dati.

Le impostazioni della modalità di importazione fanno distinzione tra maiuscole e minuscole. Di seguito sono riportati i valori consentiti.

- *APPEND* per aggiungere i POV esistenti a Gestione dati
- *REPLACE* per eliminare i dati POV e sostituirli con quelli del file
- *RECALCULATE* per ricalcolare i dati
- *NONE* per saltare l'importazione dei dati nella tabella intermedia di Gestione dati

- *EXPORT_MODE* determina in che modo esportare i dati nell'applicazione

Le impostazioni della modalità di esportazione fanno distinzione tra maiuscole e minuscole. Di seguito sono riportati i valori consentiti.

- *STORE_DATA* per unire i dati della tabella intermedia di Gestione dati con i dati esistenti. Utilizzare sempre questa opzione di esportazione nei job di Gestione dati utilizzati per caricare i metadati.
- *ADD_DATA* per aggiungere i dati della tabella intermedia di Gestione dati all'applicazione.
- *SUBTRACT_DATA* per sottrarre i dati della tabella intermedia di Gestione dati dai dati esistenti.
- *REPLACE_DATA* per cancellare i dati POV e sostituirli con quelli nella tabella intermedia di Gestione dati. Vengono cancellati i dati per Scenario, Versione, Anno, Periodo ed Entità.
- *NONE* per saltare l'esportazione dei dati dalla tabella intermedia di Gestione dati all'applicazione.

Nota:

Per Financial Consolidation and Close, sono supportate solo le modalità di esportazione elencate di seguito.

- *MERGE* per unire i dati della tabella intermedia di Gestione dati con i dati esistenti
- *REPLACE* per rimuovere le voci dalla tabella intermedia di Gestione dati e sostituirle con quelle provenienti dal caricamento dati
- *NONE* per saltare l'esportazione dei dati dalla tabella intermedia di Gestione dati all'applicazione

Per Oracle Fusion Cloud come target, sono supportate solo le modalità di esportazione riportate di seguito.

- *MERGE* per unire i dati della tabella intermedia di Gestione dati con i dati esistenti
- *NONE* per saltare l'esportazione dei dati dalla tabella intermedia di Gestione dati all'applicazione

- *FILE_NAME* è un nome file facoltativo. Se non si specifica un nome file, EPM Automate importa i dati contenuti nel nome file specificato all'interno della regola di caricamento

dati. Questo file deve essere disponibile nella cartella dei file in entrata oppure in una cartella al suo interno.

Quando si caricano file in formato BAI (Bank Administration Institute) per Account Reconciliation, non specificare un valore per questo parametro. È necessario specificare sempre il nome file per caricare file BAI nella definizione della regola dati.

 **Nota:**

Se viene specificato un percorso nella regola dati, non specificare il percorso del file nel comando; specificare solo il nome del file. Se non viene specificato un percorso nella regola dati, specificare il percorso completo del file di dati.

Esempi

- Importazione di dati di più periodi:

```
epmautomate runDataRule VisionActual Mar-15 Jun-15 REPLACE STORE_DATA
inbox/Vision/GLActual.dat
```
- Importazione di dati di un singolo periodo:

```
epmautomate runDataRule "Vision Actual" Mar-15 Mar-15 REPLACE
STORE_DATA inbox/Vision/GLActual.dat
```

runDMReport

Crea un report di Gestione dati e lo memorizza nella cartella `outbox/reports`.

Il report generato viene denominato in base all'ID del job di Gestione dati che genera il report e il relativo formato. Ad esempio, se l'ID del job del report è 2112 e il formato di output del report specificato è PDF, il nome del report è `2112.pdf`. Il nome del report è visualizzato nella console in seguito alla generazione del report. È inoltre possibile identificare il nome del report dalla scheda Dettagli processo in Data Management oppure utilizzando il comando [listFiles](#).

Utilizzare il comando [downloadFile](#) per eseguire il download dei report in un computer locale.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

```
epmautomate runDMReport REPORT_NAME PARAMETER=Value "Report Output
Format=[PDF|HTML|XLS|XLSX]" dove:
```

- `REPORT_NAME` è il nome del template di report di Data Management da utilizzare per la generazione del report.
- `PARAMETER=Value` indica i parametri del report e i relativi valori. Specificare il numero di parametri richiesto nel formato `PARAMETER=Value`. L'elenco di parametri obbligatori dipende dal report che si desidera generare.

 **Nota:**

I parametri runtime dei report vengono definiti durante la progettazione dei report. Per eseguire questo comando, è necessario generare e copiare questi parametri e valori in EPM Automate dalla scheda Flusso di lavoro. Per generare i parametri runtime di un report, nella scheda Flusso di lavoro di Gestione dati fare clic su **Esecuzione report**, quindi selezionare un gruppo in **Gruppo report**. Selezionare il report per il quale si desidera generare i parametri, quindi fare clic su **Crea script report**. Se lo si desidera, specificare i valori dei parametri del report, selezionare un formato di output, quindi fare clic su **OK**. Utilizzare i parametri visualizzati nella finestra **Genera script report** per specificare i parametri runtime e i relativi valori per generare il report.

- `Report Output Format` indica il formato di output del report. Le opzioni valide sono PDF, HTML, XLS e XLSX. Il formato di report predefinito è PDF.

Esempio

```
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

Esegue un job di Data Integration per importare dati in un processo aziendale Oracle Enterprise Performance Management Cloud o per esportare dati da un processo aziendale a un sistema esterno.

Questo comando invalida il comando `runDataRule`. Oracle consiglia di iniziare a utilizzare questo comando invece del comando `runDataRule`.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato

Utilizzo

```
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]
```

- Per le integrazioni in modalità standard, è necessario specificare i valori per `importMode`, `exportMode` e `periodName`

- Per le integrazioni in modalità rapida, è necessario specificare il valore per `exportMode`
- Per i nomi dei parametri e i relativi valori è prevista la distinzione tra maiuscole e minuscole

In questo comando, considerare quanto segue.

- `JOB_NAME` indica il nome di un job di integrazione definito in Data Integration.
- `importMode` determina come vengono importati i dati in Data Integration. Di seguito sono descritte le modalità di importazione accettabili.
 - `Append` per l'aggiunta ai dati POV esistenti in Data Integration.
 - `Replace` per eliminare i dati POV e sostituirli con i dati del file.
 - `Map and Validate` per saltare l'importazione dei dati e rielaborare invece i dati con i mapping aggiornati e i conti logici.
 - `No Import` per saltare l'importazione dei dati nella tabella intermedia di Data Integration.
- `exportMode` determina come vengono caricati i dati nell'applicazione target. Per le integrazioni in modalità rapida, non è possibile utilizzare `Check` e `No Export` come valori del parametro `exportMode`. Di seguito sono elencati i valori accettabili per le modalità di esportazione.
 - `Merge` per aggiornare i dati esistenti e aggiungere nuovi dati.
 - `Replace` per cancellare i dati esistenti nel POV e caricarvi nuovi dati. Per la modalità standard, vengono cancellati i dati per le dimensioni Scenario, Versione, Anno, Periodo ed Entità. Per la modalità rapida, vengono cancellati i dati per le dimensioni Anno, Periodo ed Entità. È possibile definire aree di cancellazione customizzate per entrambe le modalità.
 - `Accumulate` per aggiungere i dati ai dati esistenti. Si applica a Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting, Profitability and Cost Management ed Enterprise Profitability and Cost Management.
 - `Subtract` per sottrarre i dati dal saldo esistente. Applicabile a Profitability and Cost Management ed Enterprise Profitability and Cost Management. Per integrazioni in modalità rapida:
 - * Non è possibile utilizzare `Check` e `No Export` come valore per questo parametro.
 - * Per Planning, Planning Modules e Financial Consolidation and Close, gli unici valori validi sono `Replace`, `Merge` e `Accumulate`.
 - `No Export` per saltare l'esportazione dei dati. Utilizzare questa modalità per caricare i dati nella tabella intermedia per la revisione prima del caricamento nell'applicazione target.
 - `Check` per eseguire solo una verifica di convalida dei dati.

Per Oracle Fusion Cloud come target, sono supportate solo le modalità di esportazione riportate di seguito.

- * `MERGE` per unire i dati della tabella intermedia di Data Integration con i dati esistenti
- * `NONE` per saltare l'esportazione dei dati da Data Integration all'applicazione

- `periodName` è il nome di uno o più periodi o intervalli di periodi, ognuno racchiuso tra parentesi graffe, per cui importare o esportare i dati. Di seguito sono descritte le convenzioni di denominazione accettabili per i periodi.
 - Per caricamenti di singoli periodi, specificare il nome del periodo tra parentesi graffe, ad esempio `{Jan-21}`
 - Per caricamenti di dati di più periodi, racchiudere i nomi del periodo di inizio e del periodo di fine tra parentesi graffe, ad esempio `{Jan-21}{Mar-21}` (per caricare i dati per tutti i periodi a partire dal 21 gennaio fino al 21 marzo)
 - **Per Planning, Planning Modules, Financial Consolidation and CloseFreeForm e Tax Reporting:** è possibile specificare il nome e l'anno del periodo del processo aziendale nel formato `{Jan#FY21}{Mar#FY21}` per caricare i dati di tutti i periodi a partire dal 21 gennaio fino al 21 marzo.
I nomi dei periodi devono essere racchiusi tra parentesi graffe.
 - * Periodo singolo: fa riferimento al nome del periodo di Data Management per un singolo periodo definito nel mapping di periodi.
 - * Più periodi: fa riferimento al caricamento di dati di più periodi. Il parametro è specificato nel formato `{Month-Year}{Month-Year}`. Ad esempio, `{Jan-20}{Mar-20}` per un caricamento di dati di più periodi dal 20 gennaio al 20 marzo.
 - * Nome periodo di Planning: fa riferimento a un nome di periodo di Planning nel formato `{Month#Year}`, ad esempio `{Jan#FY20}{Mar#FY20}`. Utilizzando questa convenzione, non è necessario specificare i nomi dei periodi di Data Integration. Specificare invece i nomi dei membri per le dimensioni Anno e Scenario. Questo parametro è supportato nei processi aziendali di Planning, Tax Reporting e Financial Consolidation and Close. È funzionale sia per le applicazioni di servizio che per le distribuzioni cloud che derivano da origini dati in locale.

L'utilizzo di questa convenzione è utile in caso di attivazione da uno script Groovy di EPM Cloud mediante acquisizione dei nomi dei membri Anno e Periodo. Il mapping dei periodi dell'applicazione o il mapping dei periodi globali deve essere presente con Anno e Mese nei valori target del mapping dei periodi.
 - * Variabile di sostituzione: estensione del formato del nome dei periodi di Planning precedente in cui è possibile specificare una variabile di sostituzione anziché i nomi dei membri effettivi Anno e Mese nel formato `{Month#&CurYr}{&FcstMonth#&CurYr}`, ad esempio `{Jan#&CurYr}{&FcstMonth#&CurYr}`. È supportato l'utilizzo di una combinazione di nomi di membri effettivi e variabili di sostituzione.

Questo formato è supportato nei processi aziendali di Planning, Tax Reporting e Financial Consolidation and Close.

Il mapping dei periodi dell'applicazione o il mapping dei periodi globali deve essere presente in Data Integration nell'ambiente in cui viene eseguito il comando, con i valori di Anno e Mese disponibili nei valori target del mapping dei periodi. In questo caso, Anno e Mese fanno riferimento al valore corrente della variabile di sostituzione durante l'esecuzione.
 - * GLOBAL POV: esegue il caricamento dei dati per il periodo POV globale. Utilizzare il formato `{GLOBAL_POV}`.

 **Note:**

Se si utilizza un parametro di denominazione dei periodi diverso dai parametri descritti in questo documento, verrà visualizzato un messaggio di errore `Input non valido - HTTP 400`.

Copiare la sezione `periodName` dal documento dell'API REST. In questo documento vengono definiti i diversi formati, ad esempio `Global POV`, `Subvar` e così via.

- `{GLOBAL_POV}` per eseguire il caricamento dei dati per il periodo definito nel `POV` globale nel sistema o in Impostazioni applicazione in `Data Integration`.

 **Note:**

Il formato della convenzione di denominazione dei periodi `{Month#Year}` è supportato per `Planning`, `Planning Modules`, `Financial Consolidation and Close` e `Tax Reporting`. Secondo questa convenzione, è possibile specificare i nomi dei membri per le dimensioni `Anno` e `Scenario` anziché i nomi dei periodi di `Data Integration`. Questo approccio è utile se il comando viene attivato da uno script `Groovy` acquisendo i nomi dei membri `Anno` e `Periodo`. La convenzione di denominazione delle variabili di sostituzione `{Jan#&CurYr}{&FcstMonth#&CurYr}` è un'estensione della convenzione di denominazione dei periodi precedente. È possibile specificare la variabile di sostituzione anziché i nomi dei membri `Anno` e `Mese` se si esegue il comando in `Planning`, `Planning Modules`, `Financial Consolidation and Close` e `Tax Reporting`. È supportato l'utilizzo anche di una combinazione di nomi di membri e variabili di sostituzione.

Le convenzioni di denominazione dei periodi e delle variabili di sostituzione precedenti funzionano solo se i mapping dei periodi dell'applicazione o i mapping dei periodi globali con `Anno` e `Mese` nei valori target esistono già in `Data Integration`.

- `inputFileName`, per caricamenti di dati basati su file, specifica il nome del file, disponibile nella cartella `inbox`, da cui devono essere importati i dati. Se si specifica il nome di `directory` nella definizione dell'integrazione, passare solo il nome file. Se non si include un nome di `directory` nella definizione dell'integrazione, utilizzare il formato `inbox/DIR_NAME/FILE_NAME`, ad esempio `inbox/GLBALANCES.txt` o `inbox/EBSGL/GLBALANCES.txt`. Se il file è stato caricato nella posizione predefinita nell'ambiente, utilizzare la convenzione `#epminbox/FILE_NAME`, ad esempio `#epminbox/GLBALANCES.txt`, per identificare il file dei dati di input.

Questo parametro è applicabile solo ai caricamenti di dati basati su file nativi. Se non si specifica questo valore di parametro per i caricamenti di dati basati su file, questo comando importa i dati dal file specificato nella definizione dell'integrazione. Se si specifica questo valore di parametro per i caricamenti di dati non basati su file, il comando lo ignora.

- *PARAMETERS* (facoltativo) identifica i parametri runtime nel formato *PARAMETER_NAME="PARAMETER"*. I parametri includono sia i filtri di origine che le opzioni target.

 **Note:**

L'unico parametro che è possibile utilizzare in questa fase per un tipo dimensione (metadati) dell'applicazione target è "Refresh Database"=Yes|No.

Esempi

- **Importazione di dati di un periodo singolo:**

```
epmAutomate runIntegration VisionDataLoad importMode=Replace
exportMode=Merge period="{Mar-15}"
```
- **Importazione di dati di più periodi:**

```
epmAutomate runIntegration VisionDataLoad importMode=Replace
exportMode=Merge period="{Mar-15} {Jun-15}"
```
- **Integrazione di dati basati su file incrementale:**

```
epmAutomate runIntegration IncrementalFileLoad importMode=Replace
exportMode=Merge period="{Jan-20}{Mar-20}" inputFile=File1.txt
```

runMatchingReport

Genera un report definito in Corrispondenza transazione.

Per un elenco di report Corrispondenza transazione predefiniti e dei parametri per generarli, fare riferimento alla sezione Generazione di report predefiniti in Corrispondenza transazione in *Amministrazione di Account Reconciliation*.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

`epmautomate runMatchingReport FILE_NAME GROUP_NAME REPORT_NAME REPORT_FORMAT [Param=value] dove:`

- *FILE_NAME* è un nome di file univoco per il report che verrà generato. Se sul server è presente un report con lo stesso nome, esso verrà sovrascritto. Utilizzare il comando [downloadFile](#) per scaricare il report in un computer locale.
- *GROUP_NAME* è il nome del gruppo associato al report.
- *REPORT_NAME* è il nome univoco del report da generare.
- *REPORT_FORMAT* è uno dei seguenti formati del report:
 - PDF

- HTML (non supportato per grafici e diagrammi)
- XLSX (non supportato per i grafici)
- CSV
- CSV2

 **Nota:**

REPORT_FORMAT CSV non consente la formattazione di dati in base a un template mentre CSV2 sì. La generazione di report CSV2 formattati richiede più tempo rispetto all'output CSV.

- *Param=Value*, facoltativamente, identifica i parametri obbligatori per la generazione del report. Ad esempio, per il report Configurazione tipo corrispondenza che usa il parametro *status* con valore *approved*, specificare il parametro e il valore come *status=Approved*.

Esempio

```
epmautomate runMatchingReport Example_FileName "Transaction Matching"
"Match Type Configuration" HTML "status=Approved"
```

runPlanTypeMap

Copia i dati da un database di memorizzazione a blocchi a un database di memorizzazione di aggregazione o da un database di memorizzazione a blocchi in un altro un database di memorizzazione a blocchi in base alle impostazioni specificate in un job di tipo *plan type map*.

Si applica a

Planning, Planning Modules, FreeForm, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate runPlanTypeMap JOB_NAME [clearData=true|false] dove:
```

- *JOB_NAME* è il nome del job di tipo *plan type map* definito nell'applicazione.
- *clearData* è un'impostazione facoltativa che indica se prima di copiare i dati devono essere rimossi i dati nel database target. Se il valore di questo parametro non viene impostato, viene utilizzato il valore predefinito *true*.

I valori dei parametri (*true* o *false*) devono essere in minuscolo.

Esempio

```
epmautomate runPlanTypeMap CampaignToReporting clearData=false
```

runRuleSet

Esegue un set di regole business.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato (se è concesso l'accesso a esecuzione regola)

Utilizzo

`epmautomate runRuleSet RULESET_NAME [PARAMETER=VALUE] dove:`

- `RULESET_NAME` è il nome di un set di regole business così come è definito nell'ambiente.
- `PARAMETER=VALUE` indica i parametri runtime facoltativi e i relativi valori necessari per l'esecuzione del set di regole.

Nota:

Il set di regole viene eseguito per il tipo di piano nel quale è stato distribuito.

Utilizzare la combinazione `PARAMETER=VALUE` per specificare tutti i prompt runtime richiesti dal set di regole. Nell'esempio seguente vengono utilizzati due prompt runtime (`Period` ed `Entity`) e i relativi valori (`Q1` e `USA`).

Vengono utilizzati i valori predefiniti se non si specifica alcun valore per i parametri runtime. Il comando ignora i prompt runtime non corrispondenti a quelli definiti per il set di regole.

Per informazioni sull'immissione di più valori per un parametro, fare riferimento alla sezione [Definizione di valori multipli per un parametro](#).

Esempio

```
epmautomate runRuleSet RollupUSSales Period=Q1 Entity=USA
```

runSupplementalDataReport

Genera report relazionali che visualizzano i dati da Supplemental Data Manager.

I report dei dati supplementari sono raggruppati come report di non consolidamento in Financial Consolidation and Close e Tax Reporting. Per visualizzare l'elenco di report che è possibile generare e dei parametri per generarli, fare riferimento alla sezione "Elenco di report e parametri predefiniti" in Generazione di report per Financial Consolidation and Close e Tax Reporting in *API REST per Oracle Enterprise Performance Management Cloud*.

Si applica a

Financial Consolidation and Close, e Tax Reporting.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

Utilizzo

```
epmautomate runSupplementalDataReport FILE_NAME GROUP_NAME REPORT_NAME  
REPORT_FORMAT [Param=value] dove:
```

- FILE_NAME è un nome di file univoco per il report.
- GROUP_NAME è il nome del gruppo associato al report.
- REPORT_NAME è il nome univoco del report da generare.
- REPORT_FORMAT è uno dei seguenti formati del report:
 - PDF
 - HTML (non supportato per grafici e diagrammi)
 - XLSX (non supportato per i grafici)
 - CSV
 - CSV2

REPORT_FORMAT CSV non consente la formattazione di dati in base a un template mentre CSV2 sì. La generazione di report CSV2 formattati richiede più tempo rispetto all'output CSV.

- Param=value, facoltativamente, identifica i parametri obbligatori per la generazione del report. Ad esempio, per generare il report Task a rischio, per il quale nel parametro schedule name è richiesto il valore monthly e nel parametro period il valore Jan, specificare "schedule name"=monthly period=Jan.

Esempio

```
epmautomate runSupplementalDataReport Example_File_name Group1 "At Risk  
Tasks" html "schedule name"=monthly period=Jan
```

runTaskManagerReport

Genera report relazionali che visualizzano i dati da Task Manager.

I report Task Manager sono raggruppati come report di non consolidamento in Financial Consolidation and Close e Tax Reporting.

Per visualizzare l'elenco di report che è possibile generare e dei parametri per generarli, fare riferimento alla sezione "Elenco di report e parametri predefiniti" in Generazione di report per Financial Consolidation and Close e Tax Reporting in *API REST per Oracle Enterprise Performance Management Cloud*.

Si applica a

Financial Consolidation and Close, e Tax Reporting.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

Utilizzo

`epmautomate runTaskManagerReport FILE_NAME GROUP_NAME REPORT_NAME REPORT_FORMAT [Param=value] dove:`

- `FILE_NAME` è un nome di file univoco per il report.
- `GROUP_NAME` è il nome del gruppo associato al report.
- `REPORT_NAME` è il nome univoco del report da generare.
- `REPORT_FORMAT` è uno dei seguenti formati del report:
 - PDF
 - HTML (non supportato per grafici e diagrammi)
 - XLSX (non supportato per i grafici)
 - CSV
 - CSV2

Nota:

`REPORT_FORMAT` CSV non consente la formattazione di dati in base a un template mentre CSV2 sì. La generazione di report CSV2 formattati richiede più tempo rispetto all'output CSV.

- `Param=value`, facoltativamente, identifica i parametri obbligatori per la generazione del report. Ad esempio, per generare il report `Task` in anticipo, per il quale nel parametro `schedule name` è richiesto il valore `monthly` e nel parametro `period` il valore `Jan`, specificare `"schedule name"=monthly period=Jan`.

Esempio

```
epmautomate runTaskManagerReport Example_File_name Group1 "Early Tasks" PDF
"schedule name"=monthly period=Jan
```

sendMail

Invia un messaggio e-mail, con la possibilità di allegare dei file, da Oracle Enterprise Performance Management Cloud.

È possibile incorporare questo comandi in script per informare gli utenti di varie condizioni o per inviare i report.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate sendMail ToAddress Subject [Body="MessageBody"] [Attachments=FILE1,FILE2]`. I parametri sono descritti di seguito.

- `ToAddress` identifica gli indirizzi e-mail dei destinatari racchiusi tra virgolette doppie e separati da punti e virgola, ad esempio `"jdoe@example.com;jane.doe@example.com"`.
- `Subject` identifica l'oggetto del messaggio e-mail.
- `Body="MessageBody"` (facoltativo) è il contenuto del messaggio e-mail. Se non viene specificato, il corpo del messaggio e-mail è vuoto.

Note:

Utilizzare tag HTML validi per formattare il corpo del messaggio in modo da creare un formato e-mail desiderato. L'intero corpo del messaggio (inclusi tutti i tag HTML) deve essere specificato come unica riga e non deve contenere caratteri di nuova riga. Fare riferimento alla sezione [Esempi](#).

- `Attachments` (facoltativo) identifica un elenco di file separati da virgole disponibili in EPM Cloud da allegare al messaggio e-mail, ad esempio `outbox/errorFile.txt,inbox/users.csv`.

Note:

- Utilizzare * (asterisco) come carattere jolly per un carattere nel nome file. Specificare ad esempio `outbox/user*.csv` per allegare tutti i file della cartella `outbox` con nomi file di cinque lettere che corrispondono al pattern.
- È possibile allegare a un messaggio e-mail qualsiasi file, diverso da `snapshot`, elencato dal comando [listFiles](#). Le dimensioni dell'allegato non devono superare i 10 MB.

Esempi

- **E-mail non formattata:** `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`
- **E-mail formattata:** `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

Imposta l'applicazione in modalità amministrazione in modo che l'accesso all'applicazione sia limitato solo agli amministratori del servizio.

Questo comando è utile per impedire agli utenti di utilizzare l'applicazione quando gli amministratori del servizio stanno eseguendo operazioni amministrative. L'applicazione rimane in modalità amministrazione fino a quando non la si modifica in modo che tutti gli utenti possano accedervi.



Note:

Questo comando sostituisce il comando [applicationAdminMode](#) che non è più valido, ma non è stato ancora rimosso da EPM Automate.

Utilizzare il comando [getApplicationAdminMode](#) per controllare lo stato corrente dell'ambiente.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Account Reconciliation, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate setApplicationAdminMode true|false
```

In questo comando specificare `true` per attivare per l'applicazione la modalità amministrazione e `false` per ripristinare la modalità normale in modo che tutti gli utenti possano accedere all'applicazione.

Esempi

- Attivazione della modalità amministrazione per l'applicazione:
`epmautomate setApplicationAdminMode true`
- Ripristino della modalità normale dell'applicazione:
`epmautomate setApplicationAdminMode false`

setDailyMaintenanceStartTime

Imposta l'ora di inizio della manutenzione giornaliera dell'ambiente (in formato UTC o in un altro fuso orario).

Per evitare interferenze con i requisiti Oracle per la creazione di backup, questo comando non modificherà l'ora di inizio della manutenzione se il processo di manutenzione giornaliera non è stato eseguito nelle ultime 36 ore.

 **Nota:**

Gli amministratori dei servizi che sono attualmente connessi all'ambiente dopo aver effettuato l'accesso tramite browser potranno visualizzare la nuova ora di inizio della manutenzione giornaliera solo dopo essersi scollegati e ricollegati.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate setDailyMaintenanceStartTime StartTime` dove *StartTime* indica l'ora (con notazione HH:00 in formato 24 ore) in cui deve iniziare il processo di manutenzione, con l'indicazione facoltativa di un fuso orario. L'intervallo di valori accettabili per l'ora di inizio è compreso tra 00:00 e 23:00. Se l'ora di inizio non deve essere impostata nel formato UTC, specificare un fuso orario standard valido, ad esempio `"14:00 America/Los_Angeles"` per le 14.00, Ora standard Pacifico.

Esempi

- Impostazione dell'ora di inizio della manutenzione alle 14.00 UTC:
`epmautomate setDailyMaintenanceStartTime 14:00`
- Impostazione dell'ora di inizio della manutenzione alle 14.00, Ora standard Pacifico:
`epmautomate setDailyMaintenanceStartTime "14:00 America/Los Angeles"`

setDemoDates

Aggiorna i dati demo interni di Oracle in base alle necessità.

Utilizzare questo comando solo su installazione impostate con dati demo interni di Oracle.

Solo per Account Reconciliation: questo comando reimposta le date per tutte le riconciliazioni a cui sono associati valori di attributo `Demo Code` di `setdemodates` o `setdemodatesnostatuschange`. Questo comando gestisce le riconciliazioni fino a un massimo di 12 periodi, ovvero un periodo corrente e 11 periodi (cronologici) precedenti. Se le riconciliazioni di più di due periodi vengono contrassegnate con l'attributo `Demo Code`, il comando considera questi periodi come inclusi nel periodo precedente. Le riconciliazioni senza questo valore di attributo non vengono interessate.

- Se il valore è `setdemodates`, il comando reimposta le date di riconciliazione in base alla data specificata e a uno stato casuale.

- Se il valore è `setdemodatesnostatuschange`, il comando reimposta le date di riconciliazione in base alla data specificata senza modificare lo stato della riconciliazione.

Solo per Financial Consolidation and Close e Tax Reporting: questo comando reimposta le date di inizio e fine dei task, nonché altre informazioni correlate sulle date, in modo tale che i task siano idonei per una demo. Calcola le nuove date del task in base al valore dell'attributo `SETDEMODATES` impostato nella programmazione del task assieme al valore dell'attributo `Demo Date` specificato. Se il valore `Demo Date` non viene specificato, il comando utilizza la data odierna per calcolare le nuove date del task.



Nota:

Non sono interessati i task nelle programmazioni che non dispongono del valore `SETDEMODATES`.

In base all'attributo `Demo Date` specificato, questo comando sposta in avanti tutte le date associate al task. Ciò include le date di runtime di base (data di inizio, data di fine e così via), nonché le date secondarie, comprese le date della cronologia, le date di scadenza dei singoli flussi di lavoro e la data di inizio (effettiva). Lo stato del task non risulta interessato.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, and Account Reconciliation, Sales Planning e Strategic Workforce Planning

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore
È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

`epmautomate setDemoDates [DEMO_DATE]` dove `DEMO_DATE` è una data facoltativa nel formato `YYYY-MM-DD`. Le riconciliazioni vengono reimpostate sulla data corrente se non si specifica questo valore.

Esempio

```
epmautomate setDemoDates 2020-02-15
```

setEJJournalStatus

In Financial Consolidation and Close, imposta il risultato della contabilizzazione dei giornali Enterprise dal sistema ERP. Utilizzare questo comando per aggiornare lo stato della contabilizzazione dei giornali con stato `Post in Progress` indipendentemente dal relativo stato del flusso di lavoro.

Questo comando utilizza un file CSV che identifica lo stato dell'importazione nel sistema ERP. Utilizzare il comando [uploadFile](#) per caricare nell'ambiente il file di importazione. Il formato del file CSV è il seguente:

```
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"
```

La colonna Message è facoltativa e può essere omessa.

Questo comando non esporta i dati dei giornali Enterprise da Financial Consolidation and Close e non li importa nel sistema ERP.

Si applica a

Consolidamento finanziario e chiusura

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate setEJJournalStatus FILE_NAME.csv` dove `FILE_NAME` identifica il file CSV che contiene lo stato dell'importazione nel sistema ERP.

Esempio

```
epmautomate setEJJournalStatus JournalStatus.csv
```

setEncryptionKey

Imposta una chiave di cifratura customizzata per l'accesso al database.

Utilizzando questo comando fornisce una soluzione Bring Your Own Key (BYOK) che consente ai clienti di includere Oracle Enterprise Performance Management Cloud nella rotazione della gestione delle chiavi standard.

La chiave di cifratura customizzata diventa valida dopo la manutenzione giornaliera successiva dell'ambiente. Per attivarla, eseguire il comando [resetService](#).

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate setEncryptionKey key=key` dove `key` è una stringa customizzata senza limiti di lunghezza che può essere utilizzata come chiave di cifratura.

Esempi

- Impostare la chiave di cifratura: `epmautomate setEncryptionKey key=se!m+a2J`
- Rimuovere la chiave di cifratura: `epmautomate setEncryptionKey key=`

setEssbaseQryGovExecTime

Imposta il tempo massimo, in secondi, che una query Oracle Essbase può utilizzare per recuperare e fornire informazioni prima che venga terminata.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Profitability and Cost Management, Enterprise Profitability and Cost Management, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate setEssbaseQryGovExecTime TIME` dove `TIME` identifica il numero di secondi dopo i quali le query Essbase devono essere terminate. Questo valore deve essere un numero intero non superiore a 70.000.

Oracle consiglia di non impostare questo valore su 0 (zero) per evitare che le query Essbase vengano eseguite per un tempo indeterminato.

Esempio

```
epmautomate setEssbaseQryGovExecTime 600
```

setIdleSessionTimeout

Modifica il timeout di sessione (in minuti) dell'ambiente Oracle Enterprise Performance Management Cloud. Il nuovo timeout di sessione diventa attivo dopo la successiva manutenzione giornaliera dell'ambiente. Utilizzare questo comando per modificare il timeout di sessione predefinito (75 minuti) impostandolo su un altro valore. Dopo che una sessione è rimasta inattiva per l'intervallo di tempo specificato con questo comando, l'utente viene reindirizzato alla pagina di accesso.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate setIdleSessionTimeout MINUTES`, dove *MINUTES* indica il nuovo timeout di sessione inattiva in minuti (da un minimo di 15 a un massimo di 150).

Esempio

```
epmautomate setIdleSessionTimeout 30
```

setIPAllowlist

Per gli ambienti OCI (Gen 2), configura una lista di inclusione di indirizzi IP e CIDR (Classless Inter-Domain Routing) autorizzati ad accedere a Oracle Enterprise Performance Management Cloud. Questo comando aggiunge o rimuove indirizzi IPv4 e CIDR.

Questo comando fornisce un metodo self service per configurare una lista di inclusione per ambienti EPM Cloud ospitati in OCI (Gen2).

Note:

Non è possibile utilizzare questo comando per configurare la lista di inclusione negli ambienti Classic. Per gli ambienti Classic, utilizzare la schermata Dettagli servizio di My Services (Classic) per creare regole della lista di inclusione o di esclusione per gestire la modalità di accesso degli utenti agli ambienti EPM Cloud.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmAutomate setIPAllowlist add|remove FILE_NAME.txt`. I parametri sono descritti di seguito.

- `add` aggiunge alla lista di inclusione gli indirizzi IP e i CIDR elencati in un file di testo.
- `remove` elimina dalla lista di inclusione gli indirizzi IP e i CIDR elencati in un file di testo.
- *FILE_NAME* indica il nome di un file di testo in cui sono elencati gli indirizzi IP e i CIDR da aggiungere o da rimuovere dalla lista di inclusione. Ogni voce nel file deve essere separata da un carattere di nuova riga. Utilizzare il comando

[uploadFile](#) per caricare il file nell'ambiente. Ogni riga del file deve corrispondere a un indirizzo IPv4 o a un CIDR nel formato indicato di seguito.

```
xxx.xxx.xxx.xxx
xxx.xxx.xxx.xxx/n
```

 **Note:**

- Sono supportati solo gli indirizzi IP di tipo IPv4.
- Utilizzare il formato CIDR anziché singoli indirizzi IP per specificare un intervallo continuo di indirizzi IP.
- Per disabilitare una lista di inclusione in modo da consentire l'accesso da qualsiasi indirizzo IP, utilizzare il comando [getIPAllowlist](#) per scrivere tutti gli indirizzi IP e i CIDR esistenti in un file. Caricare il file nell'ambiente, quindi eseguire questo comando con l'opzione `remove` come mostrato in questo esempio:

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

Esempi

- Aggiunta di alcuni indirizzi IP e CIDR a una lista di inclusione:

```
epmAutomate setIPAllowlist add myAddList.txt
```

- Rimozione di alcuni indirizzi IP da una lista di inclusione:

```
epmAutomate setIPAllowlist remove myRemoveList1.txt
```

setManualDataAccess

Specifica se Oracle può accedere manualmente ai database relazionali e di Oracle Essbase di un ambiente in situazioni di emergenza in cui un ambiente non risponde e il cliente non ha ancora fornito una richiesta di servizio per esaminare e rendere disponibile l'ambiente.

In situazioni di emergenza, Oracle ricorre a un processo interno in cui un dirigente di sviluppo ad alto livello, dopo un processo di verifica indipendente, consente l'accesso manuale ai database relazionali ed Essbase. Utilizzare questo comando per impedire a Oracle di accedere a questi database senza l'approvazione esplicita dell'utente. Inoltre, è possibile proibire a Oracle di accedere manualmente ai database relazionali ed Essbase in situazioni di emergenza anche se è aperta una richiesta di servizio.

L'impostazione specificata mediante questo comando è valida immediatamente.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate setManualDataAccess Allow|Revoke [disableEmergencyAccess=true|false]` dove `disableEmergencyAccess` (facoltativo) consente di specificare se si desidera proibire tutti gli accessi manuali ai database relazionali ed Essbase. Impostando questo valore della proprietà su `true` si evita che Oracle acceda manualmente a questi database anche se è aperta una richiesta di servizio. L'impostazione predefinita è `false`.

Oracle sconsiglia di impostare `disableEmergencyAccess=true` per evitare il rischio di non poter accedere ai database relazionali ed Essbase in caso di necessità per risolvere un problema relativo a un ambiente inattivo. Se l'ambiente è inattivo, non sarà possibile utilizzare questo comando per consentire a Oracle di accedere manualmente a questi database.

Esempi

- Revoca dell'autorizzazione concessa per l'accesso manuale ai database relazionali ed Essbase in situazioni di emergenza senza approvazione esplicita:
`epmautomate setManualDataAccess revoke`
- Autorizzazione per l'accesso manuale ai database relazionali ed Essbase durante le emergenze:
`epmautomate setManualDataAccess allow`
- Divieto di accesso manuale ai database relazionali ed Essbase anche se è aperta una richiesta di servizio:
`epmautomate setManualDataAccess revoke disableEmergencyAccess=true`

setPeriodStatus

Imposta uno stato specifico per un periodo.

Si applica a

Riconciliazione conti

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore
È possibile che gli utenti che dispongono dei ruoli predefiniti Utente avanzato, Utente e Visualizzatore necessitino di ruoli applicazione aggiuntivi.

Utilizzo

`epmautomate setPeriodStatus PERIOD STATUS` dove:

- `PERIOD` è il nome di un periodo
- `STATUS` è OPEN, CLOSED o LOCKED

Esempio

```
epmautomate setPeriodStatus "January 2015" OPEN
```

setSubstVars

Crea o aggiorna le variabili di sostituzione a livello di applicazione o cubo.

Non è possibile utilizzare questo comando per impostare più valori e/o funzioni per le variabili di sostituzione.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate setSubstVars CUBE_NAME SUBSTVAR=VALUE [SUBSTVAR=VALUE]` dove:

- `CUBE_NAME` è il cubo, ad esempio Plan1, Plan2, per il quale creare o aggiornare le variabili di sostituzione. Utilizzare `All` invece di un nome di cubo per impostare o aggiornare la variabile di sostituzione a livello di applicazione.
- `SUBSTVAR` è il nome della variabile di sostituzione per la quale impostare o aggiornare un valore.
- `VALUE` è il nuovo valore della variabile di sostituzione.

Esempi

- Creare o aggiornare una variabile di sostituzione a livello di applicazione: `epmautomate setSubstVars ALL CurYear=2015 CurPeriod=Jan`
- Creare o aggiornare variabili di sostituzione a livello di cubo: `epmautomate setSubstVars Plan2 CurYear=2013 CurPeriod=Jan`

setVirusScanOnFileUploads

Consente all'ambiente OCI (Gen 2) di eseguire la scansione dei file per la scansione antivirus prima che vengano caricati in Oracle Enterprise Performance Management Cloud.

Tutti gli ambienti OCI (Gen 2) sono protetti tramite un programma antivirus. Questo comando offre sicurezza aggiuntiva consentendo di abilitare la scansione antivirus sui file caricati. La scansione dei file prima che vengano caricati impedisce la possibilità di caricare virus nell'ambiente.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate setVirusScanOnFileUploads true|false
```

Per impostazione predefinita, la scansione antivirus non è abilitata (è impostata su `false`). Quando questo valore è impostato su `true`, EPM Cloud esegue la scansione di tutti i file caricati. Se un file è infetto da un virus, non viene caricato nell'ambiente.

Esempio

- **Abilita la scansione antivirus:** `epmautomate setVirusScanOnFileUploads true`
- **Disabilita la scansione antivirus:** `epmautomate setVirusScanOnFileUploads false`

simulateConcurrentUsage

Consente l'esecuzione di diverse operazioni concorrenti in un ambiente simulando gli utenti.

Questo comando consente di convalidare le performance dell'ambiente e verificare che il tempo di risposta sia accettabile quando il servizio è sotto carico durante operazioni specifiche eseguite da un determinato numero di utenti. Ad esempio, il comando può essere utilizzato per misurare le performance quando 50 utenti aprono simultaneamente un modulo utilizzando diversi POV. Consente di eseguire test di carico self-service degli ambienti.

Questo comando esegue la simulazione effettuando le operazioni specificate per un determinato numero di utenti e di iterazioni. Eseguce più iterazioni per calcolare il tempo minimo, il tempo massimo e il tempo medio per una determinata operazione. Le operazioni supportate per l'esecuzione del test di carico durante l'uso simultaneo sono:

- Apertura di form
- Salvataggio di form
- Esecuzione di regole business
- Esecuzione di regole dati
- Apertura di griglie ad hoc
- Esecuzione di report
- Esecuzione di registri



Note:

Questo comando non supporta i report e i registri di Financial Reporting; supporta solo i registri e i report appartenenti a Report (in precedenza Report gestione).

Questo comando accetta come input un file ZIP, che deve essere stato caricato in precedenza nella cartella inbox dell'ambiente. Il file ZIP contiene un file `requirement.csv` e i file di input che supportano i casi d'uso inclusi in `requirement.csv`. Il comando simula quindi i casi d'uso e crea un report che può essere inviato via e-mail a uno o più destinatari.

Scenario d'uso 1: test di accettazione delle performance dell'applicazione per 50 utenti che aprono simultaneamente un form.

Soluzione

1. Creare `requirement.csv` con voci simili alle seguenti, partendo dal presupposto che si voglia aprire un form denominato `Exchange Rates` archiviato nella cartella `Library/Global Assumption/`:


```
# 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. Creare `open_form_input.csv` utilizzando il formato specificato nella sezione [File di input per l'apertura dei form](#). Il file conterrà una voce che verrà utilizzata 50 volte. Per aprire lo stesso form con POV diversi, saranno necessarie tante voci quanti sono i POV da usare.
3. Creare `userVarMemberMapping.csv` utilizzando il formato specificato nella sezione [Creazione del file UserVarMemberMapping.csv](#).
4. Creare un file ZIP contenente i file indicati nei passi precedenti e caricarlo nella cartella inbox.
5. Eseguire il comando `simulateConcurrentUsage` utilizzando come file di input il file ZIP creato nel passo precedente.

Scenario d'uso 2: simulazione delle performance per un aumento nell'utilizzo stagionale, ad esempio alla fine dell'anno finanziario. Presupposto: 100 utenti salvano un form con uno scarto temporale di sei secondi tra ogni utente.

Soluzione

1. Creare `requirement.csv` con voci simili alle seguenti partendo dal presupposto che si voglia salvare un form denominato `Accessories Revenue` archiviato nella cartella `Library/Dashboards/`:


```
# Type of Operation,Artifact Name,Number of Users,Input File,Additional Info
Save Form, Library/Dashboards/Accessories Revenue,100,save_form_input.csv,
```
2. Creare `save_form_input.csv` utilizzando il formato specificato nella sezione [File di input per il salvataggio dei form](#).
3. Creare `userVarMemberMapping.csv` utilizzando il formato specificato nella sezione [Creazione del file UserVarMemberMapping.csv](#).
4. Creare un file ZIP contenente i file indicati nei passi precedenti e caricarlo nella cartella inbox.
5. Eseguire il comando `simulateConcurrentUsage` utilizzando come file di input il file ZIP creato nel passo precedente e i valori di proprietà `iteration=1` e `lagTime=6`.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi. Il ruolo Amministratore del dominio di Identity è necessario anche per utilizzare testModes 0, 1 e 2.

Utilizzo

```
epmautomate simulateConcurrentUsage INPUT_FILE.zip [iterations=COUNT]
[notificationEmails="EMAIL_ADDRESS"] [testMode=0|1|2|3] [lagTime=LAG_TIME]
dove:
```

- *INPUT_FILE.zip* è il nome di un file ZIP che identifica i casi d'uso dell'utente. Utilizzare il comando [uploadFile](#) (esempio di sintassi `epmautomate uploadFile "C:/uploads/INPUT_FILE.zip" inbox`) per caricare il file nella cartella inbox prima di eseguire il comando. Il file ZIP deve contenere i file indicati di seguito.
 - Un file CSV del caso d'uso denominato `requirement.csv`. Ogni riga del file CSV identifica il tipo di operazione da eseguire, il nome dell'artifact, il numero di utenti concorrenti, il file di input che specifica i dettagli dell'operazione e informazioni aggiuntive correlate al singolo caso d'uso. Fare riferimento alla sezione [Creazione del file requirement.csv](#).
 - I file di input contenenti i dettagli di ogni operazione. Fare riferimento agli argomenti sui seguenti comandi:
 - * [File di input per l'apertura dei form](#)
 - * [File di input per il salvataggio dei form](#)
 - * [Esecuzione del file di input di una regola business](#)
 - * [Esecuzione del file di input della regola dati](#)
 - * [File di input della griglia ad hoc](#)
 - * [Esecuzione del file di input del registro](#)
 - * [Esecuzione del file di input del report](#)
 - * [Creazione del file UserVarMemberMapping.csv](#)
- `iterations` è un numero positivo che indica quante volte ogni caso d'uso identificato in `requirement.csv` deve essere eseguito per misurare i tempi di risposta. Se non specificato, l'operazione viene eseguita una sola volta.
- `notificationEmails`, facoltativo, indica gli indirizzi email ai quali inviare i risultati di questo comando. Se si specificano più indirizzi e-mail, separarli con un punto e virgola. Includere anche l'elenco di indirizzi tra virgolette. Se non specificato, i risultati vengono inviati all'utente che ha avviato il comando. Per ulteriori informazioni su questo report, fare riferimento alla sezione [Esempio di report di simulazione dell'utilizzo concorrente](#).
- `[testMode]`, facoltativo, specifica la modalità di simulazione dell'utilizzo concorrente. Il valore predefinito è 0.

Eseguire quindi il comando nella modalità 3 per ripetere la simulazione per il numero di volte richieste. Al termine, eseguire il comando nella modalità 2 per eliminare gli utenti simulati.

Di seguito sono riportati i valori consentiti.

- 0: la modalità di simulazione predefinita che aggiunge utenti simulati all'ambiente e assegna loro il ruolo di Amministratore servizi, esegue la simulazione e poi elimina gli utenti simulati. Questa modalità è utile per eseguire il test una sola volta. Gli utenti simulati possiedono le proprietà indicate di seguito.

Nome: testuser1, testuser2 e così via.

Cognome: testuser1, testuser2 e così via.

Indirizzo e-mail: testuser1@discard.oracle.com, testuser2@discard.oracle.com e così via.

Nome utente: testuser1, testuser2 e così via.

- 1: gli utenti simulati vengono aggiunti all'ambiente e viene loro assegnato il ruolo Amministratore servizi. La simulazione non viene eseguita né vengono eliminati gli utenti simulati.
- 2: gli utenti simulati vengono eliminati. Gli utenti non vengono creati né la simulazione eseguita.
- 3: la simulazione viene eseguita utilizzando utenti simulati già esistenti senza aggiungere né eliminare utenti.

Se si desidera eseguire una sola volta l'utilizzo concorrente, usare `testMode=0`. Per eseguire una serie di test, procedere come segue.

- Innanzi tutto eseguire il comando utilizzando `testMode=1` per aggiungere gli utenti simulati e assegnare loro il ruolo Amministratore servizi.
- Quindi eseguire il comando utilizzando `testMode=3` per ripetere la simulazione il numero di volte richiesto.
- Infine eseguire il comando utilizzando `testMode=2` per eliminare gli utenti simulati.
- `[lagTime]`, facoltativo, specifica il numero di secondi (5 o più) di attesa tra l'esecuzione di ciascun caso d'uso in `requirement.csv`. Il valore predefinito è 5 secondi. Non utilizzare numeri negativi (ad esempio -1), frazioni (ad esempio 1/2) e valori decimali. Dopo l'avvio dell'esecuzione di un caso d'uso in `requirement.csv` da parte di un utente , il comando attende il numero di secondi specificato da questo parametro per avviare l'esecuzione del caso d'uso da parte dell'utente successivo. Poiché le attività dell'utente solitamente non vengono avviate contemporaneamente, l'impostazione di questo parametro consente di creare una simulazione più realistica del carico in un ambiente.

Esempio

```
epmAutomate simulateConcurrentUsage test_simulation.zip iterations=5
notificationEmails="jane.doe@example.com;john.doe@example.com;example@example.com" lagTime=6
```

skipUpdate

Richiede a Oracle di ignorare l'applicazione degli aggiornamenti mensili a un ambiente per un massimo di tre cicli consecutivi oppure rimuove tutte le richieste di differimento degli

aggiornamenti precedentemente impostate utilizzando questo comando in modo che l'ambiente venga aggiornato in base alla codeline principale.

È possibile utilizzare questo comando anche per elencare le richieste di differimento degli aggiornamenti attualmente specificate per un ambiente. Lo stato Salta aggiornamento dell'ambiente viene incluso nel report attività generato (nelle metriche operative) dopo aver utilizzato questo comando per ignorare gli aggiornamenti da apportare a un ambiente. Fare riferimento alla sezione Metriche operative nella Guida *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori*.

Le patch settimanali e di emergenza per il mese corrente, se disponibili, continueranno ad essere applicate all'ambiente. Non verrà applicato alcun aggiornamento per i mesi per i quali è stato richiesto il differimento degli aggiornamenti.

Non è possibile ignorare gli aggiornamenti per un ambiente interessato da una patch singola. Non è inoltre possibile ignorare gli aggiornamenti mensili a più di tre mesi di distanza dall'aggiornamento attualmente implementato nell'ambiente. Ad esempio, se l'aggiornamento corrente dell'ambiente è 23.12, è possibile ignorare gli aggiornamenti 24.01, 24.02 e 24.03, ma non l'aggiornamento 24.04. Per informazioni dettagliate su come funzionano i differimenti degli aggiornamenti, fare riferimento alla sezione Richiesta di differimento di aggiornamento per gli ambienti di produzione nella *Guida operativa di Oracle Enterprise Performance Management Cloud*.

Nota:

Se per tre mesi si ignora l'aggiornamento per uno solo degli ambienti utilizzati, ad esempio non si esegue l'aggiornamento dell'ambiente di produzione mentre lo si esegue dell'ambiente di test, gli ambienti avranno una distanza di tre versioni. In uno scenario di questo tipo, potrebbe non essere possibile migrare gli snapshot tra questi ambienti.

Si supponga, ad esempio, che gli ambienti di produzione e di test abbiano la versione 23.12 e che vengano ignorati gli aggiornamenti alle versioni 24.01, 24.02 e 24.03 per il solo ambiente di produzione. Quando diventa disponibile la versione 24.03, nell'ambiente di test verrà utilizzata la versione 24.03 mentre nell'ambiente di produzione continuerà a essere utilizzata la versione 23.12. In questo caso, la migrazione tra gli ambienti di test e di produzione non è supportata.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

```
epmautomate skipUpdate add|remove|list [version=UPDATE_NUMBER
comment="COMMENT"] dove:
```

- `add` imposta una richiesta di differimento degli aggiornamenti per un aggiornamento mensile specifico. È necessario specificare i parametri descritti di seguito.
 - `version`: indica l'aggiornamento mensile da ignorare. È possibile ignorare uno o entrambi i tre successivi aggiornamenti mensili. Ad esempio, se l'ambiente utilizza l'aggiornamento mensile 23.12, è possibile ignorare gli aggiornamenti 24.01, 24.02, 24.03 o uno qualsiasi di essi. Per ignorare tre aggiornamenti mensili, eseguire il comando tre volte, specificando a ogni esecuzione l'aggiornamento da ignorare, ad esempio `version=24.01`, quindi `version=24.02` e poi `version=24.03`. In questo scenario, l'ambiente verrà aggiornato al codeline principale nel ciclo di aggiornamento mensile 24.04.

Se esiste un divario tra il ciclo mensile per il quale viene specificata una richiesta di differimento degli aggiornamenti e il ciclo mensile corrente, Oracle aggiornerà l'ambiente in base alla richiesta e quindi ignorerà gli aggiornamenti nel ciclo mensile specificato. Si supponga, ad esempio, che l'ambiente utilizzi l'aggiornamento mensile 23.12 e si specifichi di ignorare gli aggiornamenti versione 24.02 e 24.03. In questo caso, l'ambiente verrà aggiornato alla versione 24.01 mentre gli aggiornamenti 24.02 24.03 verranno ignorati. L'ambiente verrà aggiornato al codeline principale nell'aggiornamento 24.04.
 - `comment`: testo che descrive il motivo del differimento degli aggiornamenti. I commenti devono essere racchiusi tra virgolette.
- `remove` rimuove tutte le richieste di differimento degli aggiornamenti specificate per l'ambiente, in modo che l'ambiente possa essere aggiornato in base al codeline principale durante la successiva manutenzione giornaliera. Se per un ambiente sono presenti più richieste di differimento degli aggiornamenti, questo comando le rimuove tutte.
- `list`: visualizza le richieste di differimento degli aggiornamenti (ID di accesso dell'utente che ha effettuato la richiesta di differimento degli aggiornamenti, commento, versione per la quale gli aggiornamenti devono essere ignorati e data di creazione della richiesta) attualmente impostate per l'ambiente, come illustrato nella figura riportata di seguito.

```
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
```

Esempi

- Richiedere un differimento degli aggiornamenti: `epmautomate skipUpdate add version=24.01 comment="We are in the process of closing the quarter"`
- Visualizzare i dettagli del differimento degli aggiornamenti: `epmautomate skipUpdate list`

- Rimuovere tutte le richieste di differimento degli aggiornamenti: `epmAutomate skipUpdate remove`

snapshotCompareReport

Confronta due snapshot e crea il report di confronto degli snapshot, che identifica le differenze nelle regole e nei set di regole di calcolo e nei form dati inclusi negli snapshot. È possibile utilizzare questo report per la risoluzione di problemi come quelli elencati di seguito.

- Recente calo delle performance in un ambiente. È possibile confrontare lo snapshot precedente con quello corrente per controllare le differenze che possono aver causato il calo di performance.
- Riscontro di una differenza nel funzionamento o nelle performance tra due ambienti che dovrebbero avere lo stesso funzionamento o le stesse performance. In questo caso, è possibile confrontare gli snapshot dei due ambienti per individuare le differenze.
- Sospetto che alcune regole o form siano scomparsi da un ambiente. Utilizzare questo report per confrontare gli artifact che erano presenti con quelli correnti.

Si applica a

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmAutomate snapshotCompareReport SOURCE_SNAPSHOT TARGET_SNAPSHOT [reportName=REPORT_NAME.html]`. I parametri sono descritti di seguito.

- `SOURCE_SNAPSHOT` indica il nome dello snapshot rispetto al quale deve essere effettuato il confronto. Il report contiene dati sulle differenze nelle regole, nei form, nelle dimensioni e nei membri dimensione presenti nello snapshot.
- `TARGET_SNAPSHOT` indica il nome dello snapshot che si desidera confrontare.

Note:

- I nomi degli snapshot possono essere specificati con o senza l'estensione .ZIP.
 - Entrambi gli snapshot devono essere disponibili nell'ambiente. Utilizzare il comando [uploadFile](#), [copyFromObjectStorage](#) o [copySnapshotFromInstance](#) per caricarli nell'ambiente.
- `REPORT_NAME`, (facoltativo) indica il nome del file di report. Il nome di report predefinito è `SnapshotCompare.html`. Utilizzare il comando [downloadFile](#) per scaricare il report.

Esempi

- `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

Ordina i membri delle dimensioni Entità, Conto, Scenario e Versioni e di dimensioni customizzate.

Questo comando è utile per ordinare i membri delle dimensioni dopo avere caricato i membri nell'applicazione.



Nota:

Non è possibile utilizzare questo comando per ordinare membri delle dimensioni Periodo, Anni e Valuta.

Si applica a

Planning, Planning Modules, FreeForm, Enterprise Profitability and Cost Management, Sales Planning e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate sortMember Member [type=children|descendants] [order=ascending|descending] dove:`

- *Member* è il nome del membro padre di cui devono essere ordinati i discendenti o figli.
- *type*, facoltativamente, specifica i membri da ordinare. Di seguito sono riportati i valori consentiti.
 - `descendants` ordina tutti i membri secondari (figli e discendenti) del membro padre specificato come valore di *Member*
 - `children`, il valore predefinito, ordina solo i membri nel livello immediatamente inferiore al membro padre specificato come valore di *Member*.
- *order*, facoltativamente, identifica un criterio di ordinamento. Di seguito sono riportati i valori consentiti.
 - `ascending`; il criterio di ordinamento predefinito.
 - `descending`

Esempi

- Ordinare i figli delle dimensioni Entità in ordine crescente: `epmautomate sortMember Entity`

- Ordina tutti i sottomembri della dimensione Entità in ordine decrescente:
`epmautomate sortMember Entity type=descendants order=descending`

unassignRole

Rimuove un ruolo attualmente assegnato agli utenti, incluso l'utente che esegue il comando, i cui ID di accesso sono inclusi nel file CSV con codifica ANSI o UTF-8 utilizzato con questo comando. È possibile utilizzare questo comando per rimuovere l'assegnazione di un ruolo predefinito o di un ruolo applicazione.

Nota:

Non è possibile utilizzare questo comando per rimuovere le assegnazioni dei ruoli applicazione dalle applicazioni Account Reconciliation e Profitability and Cost Management, poiché tali applicazioni non supportano l'assegnazione di ruoli applicazione agli utenti.

Il formato del file CSV è il seguente:

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

Utilizzare il comando [uploadFile](#) per aggiornare il file nell'ambiente.

Nota:

Utilizzare le virgolette doppie per racchiudere i nomi dei ruoli contenenti uno spazio.

Quando termina l'esecuzione del comando, EPM Automate invia in stampa alla console le informazioni su ogni singola voce non riuscita. Esaminare queste informazioni per comprendere il motivo per cui l'esecuzione del comando non è riuscita per alcune voci nel file CSV.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Per rimuovere le assegnazioni di ruoli predefiniti, procedere come segue.

- Ambienti Classic: amministratore del dominio di Identity e qualsiasi ruolo predefinito (amministratore servizi, utente avanzato, Utente o visualizzatore)

- Ambienti OCI: amministratore servizi o amministratore del dominio di Identity e qualsiasi ruolo predefinito (amministratore servizi, utente avanzato, utente o visualizzatore)

Per rimuovere le assegnazioni di ruolo dell'applicazione: Amministratore servizi o Gestione controllo accesso

Utilizzo

`epmautomate unassignRole FILE_NAME ROLE` dove:

- `FILE_NAME` indica il nome di un file CSV contenente gli ID di accesso degli utenti per cui deve essere revocata l'assegnazione ruolo. Specificare l'estensione CSV in caratteri minuscoli.
Per i valori di User Login non è prevista la distinzione tra maiuscole e minuscole. Ad esempio, il valore `jane.doe@example.com` viene considerato uguale a `Jane.Doe@Example.com` o a qualsiasi variante di maiuscole e minuscole.
- `ROLE` identifica uno dei seguenti elementi. Per i nomi di ruoli non è prevista la distinzione tra maiuscole e minuscole.
 - In caso di rimozione delle assegnazioni dei ruoli predefiniti per gli utenti, `ROLE` deve identificare un ruolo predefinito applicabile al servizio. Fare riferimento alla sezione Introduzione ai ruoli predefiniti nella *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori*.
 - Se si sta rimuovendo l'assegnazione di utenti a ruoli applicazione, `ROLE` deve identificare un ruolo appartenente all'applicazione nell'ambiente corrente. I ruoli applicazione sono elencati nella scheda **Ruoli** di Controllo accesso. Per una descrizione dei ruoli applicazione per ciascun processo aziendale, fare riferimento agli argomenti della guida *Amministrazione del controllo dell'accesso per Oracle Enterprise Performance Management Cloud* elencati di seguito.
 - * Planning, Financial Consolidation and Close e Tax Reporting
 - * Enterprise Profitability and Cost Management
 - * Oracle Enterprise Data Management
 - * Profitability and Cost Management
 - * Account Reconciliation

Esempi

- Annullare l'assegnazione degli utenti a un ruolo di dominio di Identity predefinito:
`epmautomate unassignRole remove_roles.csv "Service Administrator"`
- Annullare l'assegnazione degli utenti da un ruolo applicazione:
`epmautomate unassignRole example_file.csv "Task List Access Manager"`

updateUsers

Modifica attributi quali l'e-mail, il nome e il cognome di utenti di Oracle Enterprise Performance Management Cloud utilizzando i nuovi valori identificati in un file con valori separati da virgola (CSV, Comma Separated Value) con codifica ANSI o UTF-8 caricato nell'ambiente.

Utilizzare il comando [uploadFile](#) per caricare file in un ambiente. Tutte le colonne nel file CSV sono obbligatorie. È necessario specificare una voce valida in ciascuna colonna. Questo comando convalida ogni definizione per questi valori obbligatori e visualizza un messaggio di

errore in cui viene identificato ogni valore mancante o non valido. Il formato del file di input è il seguente:

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

Se il valore di accesso utente nel file CSV corrisponde a un account esistente nel dominio di Identity, il comando modifica l'account utente in modo che corrisponda ai valori presenti nel file di input. Poiché gli account utente sono comuni a tutti gli ambienti supportati da un dominio di Identity, le informazioni aggiornate relative agli utenti sono disponibili per tutti gli ambienti che condividono tale dominio. I ruoli predefiniti e specifici dell'applicazione assegnati all'utente non sono interessati da questo comando.

Note:

- Non è possibile utilizzare questo comando per modificare i valori di accesso utente.
- Non si dispone dell'autorizzazione per modificare gli attributi del proprio account.
- Il file di input contenente caratteri multibyte deve utilizzare la codifica di caratteri UTF-8. L'uso della codifica ANSI causa problemi nel modo in cui le informazioni dell'utente vengono visualizzate nelle schermate di Servizi personali.

Si applica a

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, Strategic Workforce Planning e Sales Planning.

Ruoli necessari

Amministratore del dominio di Identity e qualsiasi ruolo predefinito (Amministratore servizi, Utente avanzato, Utente o Visualizzatore)

Utilizzo

`epmautomate updateUsers FILE_NAME` dove `FILE_NAME` indica il nome di un file CSV contenente le informazioni utente da modificare.

Esempio

```
epmautomate updateUsers update_user_info.csv
```

upgrade

Esegue automaticamente lo scaricamento e l'installazione in background della versione di EPM Automate più aggiornata.

Dopo aver eseguito il comando `login` per avviare una sessione, EPM Automate identifica la versione installata corrente. Se la versione installata non è la versione più recente disponibile, verrà visualizzato da EPM Automate un messaggio indicante che è disponibile una versione più aggiornata.



Nota:

Se è stato distribuito da un amministratore di Windows, EPM Automate può essere aggiornato solo se l'utente che ha eseguito l'accesso è un amministratore di Windows.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato, Utente, Visualizzatore

Utilizzo

`epmautomate upgrade`

Esempio

`epmautomate upgrade`

uploadFile

Carica un file dal computer locale nel servizio. Utilizzare questo comando per caricare file contenenti dati, metadati, definizioni di regole, definizioni di dimensioni, transazioni mappate, template e snapshot di backup.

Questo comando non sovrascrive i file esistenti nell'ambiente. EPM Automate visualizza un messaggio di errore se il nome del file che si sta caricando è uguale a quello di un file nella posizione di caricamento.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi, Utente avanzato assegnati al ruolo applicazione Amministratore migrazione

Utilizzo

`epmautomate uploadFile "FILE_NAME" [UPLOAD_LOCATION]`. I parametri sono descritti di seguito.

- `FILE_NAME` indica il nome del file, incluso il percorso assoluto se il file non è contenuto nella directory da cui si esegue EPM Automate.
- `UPLOAD_LOCATION` (facoltativo) indica la posizione Oracle Enterprise Performance Management Cloud in cui si desidera caricare il file. Non specificare una posizione di caricamento se si desidera caricare i file nella posizione di caricamento predefinita. Per informazioni dettagliate, vedere [Posizioni di file predefinite](#). Tra i valori supportati sono inclusi quelli elencati di seguito.
 - `inbox` per caricare i file nella cartella inbox. Ad eccezione di Profitability and Cost Management, i processi aziendali EPM Cloud cercano in questa posizione i file da elaborare.
 - `profitinbox` per caricare i file che devono essere elaborati da Profitability and Cost Management.
 - `to_be_imported` per caricare uno snapshot di Narrative Reporting che deve essere importato durante la successiva manutenzione giornaliera dell'ambiente.
 - `inbox/directory_name` per caricare i file in una directory nella cartella della posta in entrata per l'elaborazione da parte di Data Management.
 - `outbox` per caricare i file nella cartella outbox utilizzata da processi aziendali diversi da Profitability and Cost Management.
 - `profitoutbox` per caricare i file nella cartella outbox utilizzata da Profitability and Cost Management.

Esempi

- Caricamento di uno snapshot nella posizione predefinita:
`epmautomate uploadFile "C:/snapshots/backup_snapshot.zip"`
- Caricamento di un file nella cartella inbox di Data Management:
`epmautomate uploadFile "C:/pbcdata/quarterlydata.csv" inbox`
- Caricamento di un file in una cartella nella cartella inbox (per Data Management):
`epmautomate uploadFile "C:/fdmee_data/data.zip" inbox/dm_folder`
- Caricamento di un file nella cartella profitinbox (Profitability and Cost Management):
`epmautomate uploadFile "C:/profitability_data/data.zip" profitinbox`
- Caricamento dello snapshot di Narrative Reporting dalla directory `C:\temp` nella posizione `to_be_imported`:
`epmautomate uploadFile "C:\temp\EPRCS_Backup.tar.gz" to_be_imported`

userAuditReport

Genera un report Audit utente (file .CSV) e lo memorizza nella posizione di scaricamento predefinita.

Nel report Audit utente sono contenute informazioni sugli utenti che hanno effettuato l'accesso a un ambiente nel periodo di tempo specificato (al massimo negli ultimi 120

giorni). In questo report sono elencati l'ID di accesso utente, l'indirizzo IP del computer da cui l'utente ha eseguito l'accesso e la data e ora in cui l'utente ha avuto accesso all'ambiente, ad esempio 28 luglio 2022 18.43.31 UTC.

 **Nota:**

Nel report Audit utente è elencata una sola voce di accesso per un utente che ha eseguito l'accesso a un ambiente Oracle Enterprise Performance Management Cloud più volte nell'arco di tempo di cinque minuti.

	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

Utilizzare il comando [downloadFile](#) per scaricare il report generato nel computer in uso.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate userAuditReport FROM_DATE TO_DATE REPORT_NAME` dove:

- *FROM_DATE* indica la data di inizio (nel formato YYYY-MM-DD) del periodo per il quale generare il report di audit
- *TO_DATE* indica la data di fine (nel formato YYYY-MM-DD) del periodo per il quale generare il report di audit
- *REPORT_NAME* è il nome del file di report

 **Nota:**

Questo report può essere generato solo per gli ultimi 120 giorni.

Esempio

`epmautomate userAuditReport 2016-10-15 2016-12-15 myAuditReport.CSV`

userGroupReport

Genera un report (file CSV) contenente l'elenco dei gruppi a cui gli utenti sono assegnati in Controllo accesso e lo memorizza nella posizione di scaricamento predefinita.

Il report indica se l'assegnazione utente al gruppo è diretta (come membro del gruppo) oppure indiretta (come membro di un gruppo che è figlio di un gruppo nidificato).

Nel report sono riportati il nome di accesso dell'utente, il nome, il cognome, l'indirizzo e-mail, il gruppo assegnato e il tipo di assegnazione nel formato seguente. È identico alla versione CSV del report creato nella scheda Report gruppo utenti in Controllo accesso. Si supponga, ad esempio, che l'utente `jdoo` sia membro del gruppo `Test1`, che è un gruppo figlio del gruppo nidificato `Test2`. In questo scenario, nel report verranno visualizzate le seguenti informazioni per `jdoo`:

```
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
```

Utilizzare il comando [downloadFile](#) per scaricare il report generato nel computer in uso.

Si applica a

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 e Strategic Workforce Planning.

Ruoli necessari

Amministratore servizi o Gestione controllo accesso

Utilizzo

`epmautomate userGroupReport REPORT_NAME dove REPORT_NAME` è il nome del file di report.

Esempio

```
epmautomate userGroupReport UsrGrpReport.CSV
```

validateConsolidationMetadata

Convalida i metadati dell'ambiente per garantire un aggiornamento e un consolidamento del database senza errori.

Dopo l'importazione dei metadati tramite il comando [importMetadata](#), eseguire questo comando per convalidare i metadati e garantire un aggiornamento del database senza errori quando si esegue il comando [refreshCube](#). Se i metadati di consolidamento non sono corretti, è possibile che anche il consolidamento abbia esito negativo.

Questo comando visualizza 0 (zero) o un conteggio del numero di errori di convalida nella console da cui viene eseguito. Scrive gli errori di convalida in un file CSV che

può essere utilizzato per correggere gli errori di metadati. Utilizzare il comando [downloadFile](#) per scaricare in un server locale il file CSV risultante.

Si applica a

Consolidamento finanziario e chiusura

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate validateConsolidationMetadata LOG_FILE_NAME dove LOG_FILE_NAME` identifica il nome del file in cui saranno contenute le informazioni sugli errori individuati da questo comando.

Esempi

```
epmautomate validateConsolidationMetadata validation_error.csv
```

validateModel

Convalida un modello Enterprise Profitability and Cost Management e scrive l'output della convalida in un file.

Si applica a

Enterprise Profitability and Cost Management

Ruoli necessari

Amministratore servizi

Utilizzo

`epmautomate validateModel "modelName" FILE_NAME.txt [messageType=All|Warning|Error]`. I parametri sono descritti di seguito.

- `modelName` indica il nome del modello Enterprise Profitability and Cost Management da convalidare. Questo valore deve essere racchiuso tra virgolette.
- `FILE_NAME` indica un nome univoco di un file di testo in cui EPM Automate deve scrivere l'output della convalida del modello. Il file, creato nella cartella outbox, può essere scaricato utilizzando il comando [downloadFile](#).
- `messageType`, facoltativo, è lo stato delle informazioni da includere nell'output di convalida del modello. Di seguito sono riportati i valori di parametro possibili.
 - `All`: scrive sia gli errori sia gli avvisi nel file di output di convalida.
 - `Error`: registra solo gli errori nel file di output di convalida. Questo è il valore predefinito.
 - `Warning`: registra solo gli avvisi di convalida del modello nel file di output di convalida.

Esempio

```
epmautomate validateModel "10 Actuals Allocation Process" results.txt
messageType=All
```

Codici di uscita

EPM Automate restituisce un codice e un messaggio di uscita per indicare lo stato dell'operazione. I codici di uscita sono raggruppati in base a 5 numeri di codice. Ogni codice può indicare numerose condizioni di errore. Analizzare il messaggio corrispondente per identificare la condizioni specifica che ha causato l'errore.

Inoltre, EPM Automate crea un file di log (*COMMANDNAME_TIMESTAMP.log*, ad esempio *uploadfile_16_11_2016_11_27_10.log*) per ogni esecuzione non riuscita di un comando. I file di log vengono creati nel computer da cui si esegue EPM Automate.

Errori relativi al codice di uscita 1

Command failed to execute EPM Automate utilizza questo codice di uscita per visualizzare i messaggi correlati al codice stato HTTP 200 e 400. Questi codici vengono restituiti dalle API REST utilizzate da EPM Automate.

Insufficient privileges to perform the operation Questo errore viene visualizzato se l'utente le cui credenziali vengono utilizzate per eseguire l'accesso al servizio non dispone di privilegi sufficienti per eseguire l'operazione tentata.

Eseguire l'accesso utilizzando un account con privilegi sufficienti per eseguire l'operazione. In genere, solo gli amministratori del servizio sono autorizzati a eseguire operazioni nel servizio.

Resource does not exist Questo errore viene visualizzato se il file o lo snapshot che si desidera eliminare o scaricare non esiste nel servizio.

Utilizzare il comando `listfiles` per verificare il nome del file e la relativa posizione.

Invalid snapshot SNAPSHOT Questo errore viene visualizzato quando il servizio non è in grado di convalidare lo snapshot specificato per l'operazione di esportazione o importazione.

Verificare se si sta utilizzando uno snapshot valido.

Internal server error. Unable to delete file: FILE_NAME Please issue "Provide Feedback" with details Questo errore viene visualizzato se risulta impossibile eliminare il file o lo snapshot dal servizio a causa di un errore del server.

Segnalare questo problema a Oracle utilizzando il comando o la funzionalità Feedback.

Invalid file: FILE_NAME Questo errore viene visualizzato se il file o lo snapshot che si desidera eliminare o scaricare non esiste nel servizio oppure se il nome di file non è nel formato richiesto.

Utilizzare il comando `listfiles` per verificare il nome del file e la relativa posizione.

Recreate is running for a long time. Please contact support Questo errore viene visualizzato se l'operazione di ricreazione avviata non è stata completata entro un'ora.

Segnalare questo problema a Oracle utilizzando il comando o la funzionalità Feedback.

Reset service is running for a long time. Please contact support

Questo errore viene visualizzato se l'operazione di ripristino del servizio avviata non è stata completata entro un'ora.

Segnalare questo problema a Oracle utilizzando il comando o la funzionalità `Feedback`.

Cannot perform operation. Another instance is in progress. Please

try after some time Questo errore viene visualizzato se si cerca di eseguire il comando `copysnapshotfrominstance` quando è attiva un'altra istanza dello stesso comando.

Attendere il termine dell'esecuzione del comando `copysnapshotfrominstance` prima di cercare di eseguire di nuovo il comando.

Cannot perform operation. Another maintenance script is in progress.

Please try after some time Questo errore viene visualizzato se si cerca di eseguire il comando `copysnapshotfrominstance`, `recreate` o `resetservice` quando è in esecuzione il processo di manutenzione giornaliera o di ripristino del servizio.

Rieseguire l'operazione al termine del processo di manutenzione o ripristino.

Login to source instance failed: SOURCE_URL Questo errore viene visualizzato

se EPM Automate non è in grado di eseguire l'accesso all'ambiente di origine per avviare il comando `copysnapshotfrominstance`.

Verificare che le credenziali, il dominio di Identity e l'URL utilizzati per accedere all'ambiente di origine siano validi.

Internal server error. Copy snapshot from source instance failed.

Please issue "Provide Feedback" with details Questo errore viene visualizzato quando EPM Automate rileva un errore imprevisto durante l'esecuzione del comando `copysnapshotfrominstance`.

Segnalare questo problema a Oracle utilizzando il comando o la funzionalità `feedback`.

Internal server error. Please issue "Provide Feedback" with details

Questo errore viene visualizzato per indicare numerose condizioni interne del server che richiedono un intervento correttivo da parte di Oracle.

Segnalare questo problema a Oracle utilizzando il comando o la funzionalità `feedback`.

Snapshot SNAPSHOT_NAME already exists. Please delete the snapshot and rerun the command Questo errore viene visualizzato quando si scarica o si carica uno snapshot in una posizione in cui è presente un altro snapshot con lo stesso nome.

Eliminare o rimuovere lo snapshot esistente, quindi rieseguire il comando.

Error in extracting the snapshot. Please retry with a proper

snapshot Questo errore viene visualizzato se EPM Automate non è in grado di estrarre il contenuto dello snapshot durante l'esecuzione del comando `importsnapshot`.

Verificare che lo snapshot sia valido.

Internal server error. Unable to open file for write. Please issue

"Provide Feedback" with details Questo errore viene visualizzato se gli errori causano la creazione o l'aggiornamento dei file CSV, ad esempio durante la generazione del report di audit.

Segnalare questo problema a Oracle utilizzando il comando o la funzionalità `feedback`.

No matching records found, please select a different date range
Questo errore viene visualizzato se si esegue il comando `userauditreport` per generare il report di audit per un intervallo di date per il quale non sono disponibili dati di audit.

Specificare un intervallo di date valido, quindi rieseguire il comando `userauditreport`. Si noti che il servizio conserva la cronologia di audit solo per gli ultimi 365 giorni.

File with same name exists: FILE_NAME, please choose a different filename Questo errore viene visualizzato se nel servizio esiste già un report con il nome del report di audit specificato.

Eliminare il file esistente dal servizio oppure specificare un nome di file diverso, quindi rieseguire il comando `userauditreport`.

Operation failed with status \$1. Please issue "Provide Feedback" Questo messaggio indica un errore interno del server che ha causato la mancata esecuzione del processo di ripristino o ricreazione del servizio.

Segnalare questo problema a Oracle utilizzando il comando o la funzionalità `feedback`.

EPMCSS-20643: Failed to add users. File FILE_NAME.csv is not found. Please provide a valid file name Questo errore viene visualizzato se il file CSV specificato che contiene le informazioni sugli utenti da aggiungere non è disponibile nella cartella Posta in entrata.

Utilizzare il comando `listfiles` per verificare il nome del file e la relativa posizione. Se il file non si trova nella cartella inbox, utilizzare il comando `uploadFile` per caricarlo.

EPMCSS-20644: Failed to remove users. File FILE_NAME.csv is not found. Please provide a valid file name Questo errore viene visualizzato se il file CSV specificato che contiene le informazioni sugli utenti da eliminare non è disponibile nella cartella Posta in entrata.

Utilizzare il comando `listfiles` per verificare il nome del file e la relativa posizione. Se il file non si trova nella cartella inbox, utilizzare il comando `uploadFile` per caricarlo.

20645: Failed to assign role for users. Invalid role name role. Please provide a valid role name Questo errore viene visualizzato se il ruolo specificato nel file CSV non è supportato.

Verificare che il nome ruolo specificato nel file sia `Service Administrator`, `Power User`, `User 0 Viewer`.

Utilizzare il comando `listfiles` per verificare il nome del file e la relativa posizione. Se il file non si trova nella cartella inbox, utilizzare il comando `uploadFile` per caricarlo.

Errori relativi al codice di uscita 6

Service Unavailable Il servizio non è disponibile a causa di un errore HTTP 404.

Per verificare la disponibilità del servizio, accedere al servizio da un browser sul computer da cui si sta eseguendo EPM Automate. Se per qualche motivo il servizio non è attivo, attendere, quindi riprovare oppure contattare il Supporto Oracle.

Read/Write timeout Questo errore viene visualizzato se si verifica il timeout del socket del client (il timeout del socket è pari a 15 minuti) durante qualsiasi operazione di lettura/ scrittura a causa di problemi di rete lenta o firewall.

Rieseguire il comando non eseguito quando il throughput di rete è elevato. Se l'errore è causato dalle impostazioni del firewall, contattare l'amministratore della rete.

Errori relativi al codice di uscita 7

EPM Automate visualizza questo errore se non è in grado di eseguire un comando. Il messaggio di errore, ad esempio `Invalid command`, specifica il motivo dell'errore.

Unable to open password file *FILE_NAME* Invalid encrypted password file, for example, `PASSWORD_FILE.EPW`. EPM Automate non ha trovato il file nella posizione specificata oppure il file non è nel formato richiesto.

Verificare il nome e il percorso del file. Se il file non può essere analizzato a causa di un formato non valido, utilizzare il comando `encrypt` per ricreare il file.

Unable to parse password file *FILE_NAME* EPM Automate non è stato in grado di analizzare il file delle password cifrate a causa del formato non valido oppure perché il file è stato danneggiato.

Utilizzare il comando `encrypt` per ricreare il file.

Unable to connect to *URL*. Root cause *MESSAGE* Questo errore viene visualizzato se risulta impossibile stabilire una connessione a causa di un URL non valido. Nel messaggio visualizzato come causa principale è dettagliato l'errore di base generato dall'uso di un URL errato.

- Verificare se si sta utilizzando un URL valido.
- Se l'impostazione proxy richiede l'autenticazione al server proxy per la connessione a Internet, specificare il nome utente, il dominio e la password di un server proxy (oppure utilizzare un password file cifrato contenente la password del server proxy) per eseguire l'accesso. Per assistenza, contattare l'amministratore della rete.

Unable to connect to *URL* Unsupported Protocol Il comando di login non è riuscito perché l'URL specificato utilizza un protocollo non supportato. Il messaggio di errore corrispondente indica il protocollo non supportato.

Assicurarsi che l'URL usato con il comando di login utilizzi un protocollo sicuro (HTTPS).

Session is not authenticated. Please execute the "login" command before executing any other command Si è cercato di eseguire un comando prima di stabilire una sessione utilizzando il comando `login`.

Eseguire il comando `login` per stabilire una connessione sicura all'ambiente prima di cercare di eseguire altri comandi.

Invalid parameter Questo messaggio fa riferimento a un errore di utilizzo in un comando causato da una sequenza errata di parametri del comando oppure dall'assenza di alcuni valori di parametro obbligatori per il comando.

Rivedere e correggere i parametri del comando e la sequenza in base alla quale sono stati specificati.

COMMAND_NAME* command is not supported by *SERVICE_TYPE EPM Automate non è stato in grado di eseguire il comando per l'ambiente a cui si è connessi perché il processo aziendale non supporta il comando.

Fare riferimento alla sezione [Guida comandi](#) per gli elenchi dei comandi supportati da ciascun processo aziendale.

File does not exist in location: *PATH* EPM Automate non è stato in grado di trovare il file che si desidera elaborare, ad esempio, utilizzando il comando `upload o replay`.

Verificare che il nome e il percorso del file siano corretti.

Unable to open file for read: *PATH* EPM Automate non è stato in grado di leggere il file specificato.

Verificare che il file sia nel formato richiesto. Verificare che l'utente che sta eseguendo EPM Automate abbia l'accesso in lettura al file.

Unable to open file for write: *PATH* EPM Automate non è stato in grado di scrivere nel file specificato.

Assicurarsi che il file non sia bloccato da un altro processo. Verificare che l'utente che sta eseguendo EPM Automate abbia l'accesso in scrittura al file.

Invalid command EPM Automate ha rilevato un comando non supportato.

Verificare che EPM Automate supporti il comando. Assicurarsi inoltre che il nome del comando non contenga errori di ortografia.

Invalid date format Lo strumento ha rilevato un formato di data non valido.

Specificare le date di generazione del report in un formato di data supportato.

FROMDATE *DATE* cannot be greater than TODATE *DATE* EPM Automate ha rilevato una data di fine anteriore alla data di inizio.

Assicurarsi che la data specificata da `to date` nell'intervallo di date definito sia successivo alla data specificata da `from date`.

Exceeded maximum number of feedbacks (6) for a day Questo errore viene visualizzato quando è stato superato il numero di feedback che è possibile sottomettere utilizzando il comando `feedback`.

File with the same name already exists in the download path *PATH*. Please delete the file and rerun the command Questo errore viene visualizzato quando si tenta di scaricare un file in una posizione in cui è presente un altro file con lo stesso nome.

Eliminare, rinominare o rimuovere il file esistente, quindi eseguire di nuovo il comando.

File is empty: *PATH* Questo errore viene visualizzato se il file di ripetizione non include contenuti.

Assicurarsi che il file di ripetizione (file CSV) includa l'elenco delle credenziali (nome utente e password) e il nome dei file HAR da utilizzare per eseguire il comando `replay`.

Unable to encrypt the password as localhost cannot be resolved. Ensure that hostnames are mapped properly to IP addresses Questo errore viene visualizzato se EPM Automate non è in grado di risolvere la definizione `localhost` in un indirizzo MAC perché il file `host` nel computer in uso contiene un nome server anziché `localhost` per l'indirizzo `127.0.0.1`.

Assicurarsi che nel file host sia specificato `localhost` come nome server per `127.0.0.1`

Snapshot Name is invalid Questo errore viene visualizzato se non si specifica il nome dello snapshot da rinominare.

Specificare il nome di uno snapshot disponibile nell'ambiente.

New Snapshot Name is invalid Questo errore viene visualizzato se non si specifica un nuovo nome per lo snapshot.

Specificare il nuovo nome per lo snapshot.

Invalid snapshot name: {0}. Characters \\/*?"<>| are not allowed
Questo errore viene visualizzato se il nome dello snapshot contiene caratteri speciali quali, ad esempio, spazi, \ (barra rovesciata), / (barra), * (asterisco), ? (punto interrogativo), " (virgolette), < (minore) e > (maggiore).

Specificare un nuovo nome di snapshot che non contenga questi caratteri speciali.

Unable to rename snapshot : {0}. There could be another process accessing it. Please try after sometime Questo errore viene visualizzato se EPM Automate non è in grado di ottenere un lock esclusivo sullo snapshot perché è utilizzato da un altro processo.

Attendere il completamento dell'operazione che utilizza lo snapshot, quindi riprovare.

Snapshot {0} already exist. Please delete the snapshot and re-run the command Questo errore viene visualizzato se il nome del nuovo snapshot è uguale a quello di uno snapshot già presente nell'ambiente.

Utilizzare un nome diverso oppure eliminare lo snapshot esistente utilizzando il comando `deletefile`.

Errori relativi al codice di uscita 9

Invalid credentials Questo errore viene visualizzato quando il nome utente o la password utilizzata con il comando `login` non è corretta.

Specificare credenziali valide per l'ambiente a cui si sta cercando di connettersi.

Authentication failed while executing command. Riprovare Questo errore viene visualizzato quando l'autenticazione di base non riesce durante l'esecuzione di un comando diverso da `login`. Questo errore può inoltre verificarsi per le chiamate HTTP quando viene tentata di nuovo l'esecuzione di un comando (per un massimo di tre volte).

Errori relativi al codice di uscita 11

Errore server interno. A causa del servizio di reimpostazione manuale, l'ambiente Oracle EPM Cloud Service non è attualmente disponibile. Questo errore viene visualizzato se i comandi di EPM Automate vengono eseguiti mentre è in corso una reimpostazione dell'ambiente.

Internal server error MESSAGE Questo errore viene visualizzato se EPM Automate rileva eccezioni sconosciute non associate alle connessioni HTTP. Include gli errori server 503 e 500.

Unable to connect to URL: MESSAGE Questo errore viene visualizzato quando il server non è disponibile. Il messaggio di errore fa riferimento all'eccezione che ha causato la mancata esecuzione del comando.

Se il server non è disponibile, contattare il Supporto Oracle. Se il messaggio fa riferimento a errori associati all'URL, verificare che l'URL in uso sia valido.

3

Scenari di esempio per l'esecuzione dei comandi

È possibile utilizzare EPM Automate per automatizzare molti task amministrativi comuni di Oracle Enterprise Performance Management Cloud.

- [Scenari di esempio per tutti i servizi](#)
- [Scenari campione per Planning, Consolidation, Tax Reporting ed Enterprise Profitability and Cost Management](#)
- [Scenari di esempio per Account Reconciliation](#)
- [Scenari di esempio per Profitability and Cost Management](#)
- [Scenari di esempio per Oracle Enterprise Data Management Cloud](#)

Informazioni sulla copia di script campione

Non copiare gli script campione dalla versione PDF di questo documento. Per evitare interruzioni di riga e informazioni a piè di pagina che renderebbero gli script inutilizzabili, Oracle consiglia di copiare gli script campione dalla versione HTML della guida [Utilizzo della utility Automazione EPM per Oracle Enterprise Performance Management Cloud](#).

Scenari di esempio per tutti i servizi

Gli scenari seguenti descrivono una tipica sequenza di comandi che è possibile utilizzare per eseguire specifiche operazioni nell'ambiente Oracle Enterprise Performance Management Cloud.

Vedere anche:

- [Creazione del backup dello snapshot dell'applicazione in un computer](#)
Questo scenario illustra come automatizzare il processo di creazione del backup dello snapshot creato durante la manutenzione giornaliera del servizio in un computer locale.
- [Informazioni agli utenti sul completamento della manutenzione giornaliera](#)
La manutenzione giornaliera dell'ambiente Oracle Enterprise Performance Management Cloud richiede in genere molto meno tempo dell'ora formalmente indicata.
- [Copia di uno snapshot a oppure da Oracle Object Storage](#)
- [Creazione di utenti e assegnazione a ruoli predefiniti](#)
Utilizzare gli script in questa sezione per creare utenti e assegnarli a ruoli predefiniti nel dominio di Identity.
- [Conteggio del numero di utenti con licenza \(utenti assegnati a ruoli\)](#)
Utilizzare lo script descritto in questa sezione per generare il Report assegnazioni ruoli per eseguire il conteggio del numero di utenti per un ambiente.

- [Creazione di report di audit di utenti assegnati a ruoli](#)
Utilizzare gli script descritti in questa sezione per automatizzare il processo di creazione di un report di audit per gli utenti assegnati a ruoli predefiniti in un ambiente e, facoltativamente, inviare via e-mail tale report a un destinatario.
- [Creazione del report di audit sull'assegnazione e revoca dei ruoli](#)
Utilizzare lo script di PowerShell descritto in questa sezione per automatizzare il processo di creazione di un report di audit contenente i dettagli dell'assegnazione e della revoca dei ruoli in un ambiente.
- [Mascheramento dei log di accesso e del report attività per conformità con le leggi sulla privacy](#)
Utilizzare gli script descritti in questa sezione per automatizzare il processo di mascheramento delle informazioni nel report attività o nei log degli accessi in modo conforme alle normative sulla privacy in vigore e, facoltativamente, inviare via e-mail il report a un destinatario.
- [Automazione del download dei report attività in un computer locale](#)
Utilizzare lo script descritto in questa sezione per automatizzare lo scaricamento dei report attività da un ambiente in un computer locale.
- [Scaricamento dei log di accesso da un ambiente](#)
Utilizzare lo script descritto in questa sezione per automatizzare il processo di scaricamento dei log di accesso da un ambiente in un computer locale.
- [Automazione della clonazione degli ambienti](#)
Utilizzare lo script descritto in questa sezione per automatizzare la clonazione degli ambienti.
- [Rimozione dei file superflui da un ambiente](#)
Utilizzare questi script per rimuovere da un ambiente i file superflui.
- [Ricerca e scaricamento di file da un ambiente](#)
Utilizzare lo script di esempio descritto in questa sezione per automatizzare il processo di scaricamento di uno o più file da un ambiente Oracle Enterprise Performance Management Cloud utilizzando una stringa di testo come carattere jolly.
- [Come ricreare un vecchio ambiente EPM Cloud per gli audit](#)
Utilizzare lo script descritto in questa sezione per creare una soluzione self-service per mantenere aggiornata una libreria di snapshot per l'ambiente Oracle Enterprise Performance Management Cloud in uso. È necessario un ambiente dedicato ai fini dell'aggiornamento e della gestione di una libreria di snapshot aggiornati.
- [Automazione dell'audit e della conformità degli accessi al database](#)
Utilizzare gli script shell di PowerShell e Bash descritti in questa sezione per usare i comandi EPM Automate per raccogliere i dati di audit e conformità relativi all'accesso manuale al database.
- [Replica degli utenti e delle assegnazioni dei ruoli predefiniti](#)
Gli script descritti in questa sezione consentono di migrare gli utenti e le assegnazioni dei ruoli predefiniti da un ambiente a un altro.
- [Creazione di una cadenza di aggiornamento trimestrale di EPM Cloud](#)
Utilizzare questi script per creare una soluzione self-service per ignorare gli aggiornamenti in modo che gli ambienti Oracle Enterprise Performance Management Cloud vengano aggiornati con frequenza trimestrale con un ciclo di test quindicinale. In questo caso gli ambienti di produzione vengono aggiornati due settimane dopo gli ambienti di test.

- [Creazione di una cadenza di aggiornamento trimestrale di EPM Cloud con cicli di test di sei settimane](#)
 Utilizzare lo script di questa sezione per creare una soluzione self-service per ignorare gli aggiornamenti in modo che gli ambienti Oracle Enterprise Performance Management Cloud vengano aggiornati con frequenza trimestrale con un ciclo di test di sei settimane. In questo caso gli ambienti di produzione vengono aggiornati sei settimane dopo gli ambienti di test.

Creazione del backup dello snapshot dell'applicazione in un computer

Questo scenario illustra come automatizzare il processo di creazione del backup dello snapshot creato durante la manutenzione giornaliera del servizio in un computer locale.

- esegue il download dello snapshot dell'applicazione (`Artifact Snapshot`) creato durante la finestra di manutenzione;
- rinomina lo snapshot scaricato aggiungendovi l'indicatore orario;
- conserva 10 backup eliminando il più vecchio, se necessario.

Note:

- Questo script non può essere utilizzato per eseguire il backup di Narrative Reporting
- Se si assegna un nuovo scopo a questo script per un uso personale, modificare in modo adeguato i valori dei parametri di runtime (`url`, `user`, `password` e `NumberOfBackups`).

Vedere [Automazione dell'esecuzione degli script](#) per informazioni su come pianificare lo script mediante Utilità di pianificazione di Windows.

Script di esempio per Windows

Creare un file batch (`.bat`) o shell (`.sh`) contenente uno script simile a quello riportato di seguito per rendere automatico lo scaricamento degli snapshot.

```
@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%

```

Script di esempio di Linux/UNIX

Creare un file shell (.sh) contenente uno script simile a quello riportato di seguito per rendere automatici gli scaricamenti dello snapshot. Se la password contiene caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#).

```

#!/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} ${op}"
    operationoutput=`eval "$sepmautomatescript $op"`
    printResult "$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 --

```

Informazioni agli utenti sul completamento della manutenzione giornaliera

La manutenzione giornaliera dell'ambiente Oracle Enterprise Performance Management Cloud richiede in genere molto meno tempo dell'ora formalmente indicata.

La durata della manutenzione giornaliera effettiva dell'ambiente è registrata come valore della metrica "Manutenzione giornaliera - Durata (minuti)" nella sezione "Metriche operazioni" del report attività. Se non si desidera attendere un'intera ora prima di utilizzare l'ambiente, usare una versione customizzata di questo script per informare gli utenti che la manutenzione giornaliera è completa e che possono riprendere le attività.

Script per Windows

Creare `daily_maintenance_completed.ps1` copiando lo script PowerShell indicato di seguito. Per informazioni sull'aggiornamento dello script da utilizzare, fare riferimento alla sezione [Esecuzione dello script](#).

```

# 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."

```

Script Linux/UNIX

Creare `daily_maintenance_completed.sh` copiando lo script indicato di seguito. Per informazioni sull'aggiornamento dello script da utilizzare, fare riferimento alla sezione [Esecuzione dello script](#).

```

#!/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."

```

Script Groovy lato server

Creare lo script Groovy `daily_maintenance_completed` copiando il codice indicato di seguito. Per informazioni sull'aggiornamento dello script da utilizzare, fare riferimento alla sezione [Esecuzione dello script](#).

```

// 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.")
```

Esecuzione dello script

Windows e Linux/UNIX

1. Creare `daily_maintenance_completed.ps1` o `daily_maintenance_completed.sh` copiando lo script da una sezione precedente.
2. Aggiornare lo script:
 - **Windows:** aggiornare il valore di `emailaddresses` con un elenco di indirizzi e-mail separati da virgole a cui inviare la notifica al termine della manutenzione giornaliera.
 - **Linux/UNIX:** aggiornare le variabili riportate di seguito.
 - `epmautomatescript` con la posizione dell'eseguibile di EPM Automate.
Esempio: `epmautomatescript="/home/utils/EPMAutomate/bin/epmautomate.sh"`
 - `javahome` con la directory in cui è installata la versione di JDK utilizzata da EPM Automate. Ad esempio: `"/home/user1/jdk1.8.0_191"`
 - `emailaddresses` con un elenco di indirizzi e-mail separati da virgole a cui inviare la notifica al termine della manutenzione giornaliera. Ad esempio: `jdoh@example.com,jane_doh@example.com`
3. In una finestra o console di comandi, passare alla cartella in cui è memorizzato lo script `daily_maintenance_completed`.
4. Eseguire il comando:
 - **Windows:** `./daily_maintenance_completed.ps1 USERNAME PASSWORD URL`
 - **Linux/UNIX:** `./daily_maintenance_completed.sh USERNAME PASSWORD URL, dove:`
 - `USERNAME` è il nome utente di un amministratore del servizio
 - `PASSWORD` è la password dell'amministratore del servizio
 - `URL` è l'URL dell'ambiente EPM Cloud.

Groovy lato server:

1. Creare lo script Groovy `daily_maintenance_completed.groovy` copiandolo da una sezione precedente.
2. Aggiornare i valori indicati di seguito.
 - `username` con il nome utente di un Amministratore servizi.
 - `password` con la password dell'Amministratore servizi.
 - `url` con l'URL dell'ambiente EPM Cloud per il quale effettuare la notifica del completamento della manutenzione giornaliera. Ad esempio: Esempio: `https://testExample-idDomain.pbcs.us1.oraclecloud.com`
 - `emailaddresses` con un elenco di indirizzi e-mail separati da virgole a cui inviare la notifica al termine della manutenzione giornaliera.
3. Utilizzare la schermata Groovy in un processo aziendale EPM Cloud oppure rendere automatica l'esecuzione dello script come indicato nella sezione

[runBusinessRule](#). Per ulteriori informazioni, fare riferimento alle fonti di informazioni riportate di seguito.

- [Running Commands without Installing EPM Automate \(in lingua inglese\)](#)
- Utilizzo delle regole Groovy nella guida *Amministrazione di Planning*.

Copia di uno snapshot a oppure da Oracle Object Storage

Questo argomento contiene script campione per completare i seguenti task:

- Copiare `Artifact Snapshot` (lo snapshot di manutenzione) da Oracle Enterprise Performance Management Cloud a un gruppo di Oracle Object Storage e ridenominarlo aggiungendo la data in cui è stato copiato lo snapshot.
- Copiare uno snapshot di backup da un gruppo di Oracle Object Storage in EPM Cloud.

Gli script in questa sezione prevedono che l'utente abbia già creato un gruppo in Oracle Object Storage dove conservare lo snapshot. Prima di eseguire questi script, customizzarli per poterli utilizzare aggiornando i seguenti segnaposto:

Table 3-1 Parametri e relativi valori

Segnaposto	Valore previsto
<code>JAVA_HOME</code>	Directory dove è installato il JDK utilizzato da EPM Automate. Esempio: <code>./home/JDK/bin</code>
<code>epmautomateExe</code>	Directory dove è installato EPM Automate. Esempio: <code>./home/utills/EPMAutomate/bin</code>
<code>cloudServiceUser</code>	ID utente di chi in EPM Cloud ha il ruolo di amministratore dei servizi. Esempio: <code>John.doe@example.com</code>
<code>cloudServicePassword</code>	Password dell'utente con ruolo Amministratore servizi o posizione in cui è disponibile il file delle password. Se la password contiene caratteri speciali, fare riferimento alla sezione Gestione dei caratteri speciali . Esempio: <code>ex_PWD_213</code>
<code>cloudServiceUrl</code>	URL dell'ambiente EPM Cloud da cui <code>Artifact Snapshot</code> deve essere copiato. Esempio: <code>https://test-cloud-id_Dom.pbcs.us1.oraclecloud.com</code>
<code>objectStorageUser</code>	ID utente di un utente in Oracle Object Storage. Per copiare uno snapshot in Object Storage, è necessario che l'utente disponga delle autorizzazioni di accesso in scrittura per il gruppo in cui viene copiato lo snapshot. Per copiare uno snapshot da Object Storage, è necessario che l'utente disponga delle autorizzazioni di accesso in lettura per il gruppo da cui viene copiato lo snapshot. Esempio: <code>jDoe</code>
<code>objectStoragePassword</code>	Password di <code>objectStorageUser</code> . Esempio: <code>example_PWD</code>

Table 3-1 (Cont.) Parametri e relativi valori

Segnaposto	Valore previsto
objectStorageBucketUrl	URL del gruppo di Oracle Object Storage dove lo snapshot deve essere copiato. Verificare le seguenti informazioni per il formato URL: <ul style="list-style-type: none"> • copyToObjectStorage • copyFromObjectStorage Esempio: https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/axaxnpcrorw5/bucket-20210301-1359
snapshot	Nome dello snapshot da copiare dal gruppo di Oracle Object Storage. Esempio: Artifact Snapshot20210429.zip

Script di EPM Automate campione per copiare uno snapshot da EPM Cloud a Oracle Object Storage

Customizzare ed eseguire lo script per ridenominare e copiare Artifact Snapshot da EPM Cloud a un gruppo di Oracle Object Storage.

```
#!/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
```

Script di EPM Automate campione per copiare uno snapshot da Oracle Object Storage a EPM Cloud

Customizzare ed eseguire lo script per copiare Artifact Snapshot di una data specifica da un gruppo di Oracle Object Storage a EPM Cloud.

```
#!/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

```

Creazione di utenti e assegnazione a ruoli predefiniti

Utilizzare gli script in questa sezione per creare utenti e assegnarli a ruoli predefiniti nel dominio di Identity.

Questi script usano i comandi di EPM Automate per completare le attività riportate di seguito.

- Accedere all'ambiente come Amministratore servizi con il ruolo Amministratore del dominio di Identity.
- Esportazione delle informazioni relative a gruppi e appartenenza dall'ambiente per rigenerare uno snapshot specificato, ad esempio `example_snapshot.zip`. Si presume che per creare lo snapshot siano già stati esportati gli artifact di Gruppi e appartenenza utilizzando l'opzione Migrazione.
- Scaricare lo snapshot (`example_snapshot.zip`) in una directory locale.
- Eliminare lo snapshot (`example_snapshot.zip`) dall'ambiente.
- Scollegarsi dall'ambiente.
- Eseguire operazioni per le quali è stato aggiunto il codice customizzato. Tali operazioni possono includere le attività indicate di seguito.
 - Estrazione del contenuto di `example_snapshot.zip`
 - Aggiunta di informazioni sui nuovi utenti al file `HSS-Shared Services\resource\External Directory\Users.csv` nel formato Nome, Cognome, E-mail, Login utente. Per informazioni dettagliate, fare riferimento a Importazione di un batch di account utente in *Guida introduttiva a Oracle Cloud*.
 - Aggiunta di informazioni sulle assegnazioni dei ruoli ai nuovi utenti (nel formato Nome, Cognome, E-mail, Login utente) ai file dei ruoli appropriati in `HSS-Shared Services\resource\External Directory\Roles\`. L'assegnazione del ruolo Amministratore servizi, ad esempio, deve essere aggiunta a `<service_name> Service Administrator.csv`, mentre le assegnazioni del ruolo Visualizzatore devono essere aggiunte a `HSS-Shared Services\resource\External Directory\Roles\<service_name> Viewer.csv`. Per informazioni dettagliate, fare riferimento a Assegnazione di un ruolo a più utenti in *Guida introduttiva a Oracle Cloud*.
 - Creazione di un nuovo snapshot con i file aggiornati comprimendo la directory `HSS-Shared Services` con i relativi contenuti.

- Accedere all'ambiente come Amministratore servizi, a cui è assegnato anche il ruolo di Amministratore del dominio di Identity.
- Caricare nell'ambiente il file `example_snapshot.zip` modificato.
- Importare `example_snapshot.zip` nell'ambiente.
- Eliminare dall'ambiente il file `example_snapshot.zip` caricato.
- Scollegarsi.

Script di esempio per Windows

Creare un file denominato `createUsersAndAssignRoles.ps1` copiando lo script seguente. Archivarlo in una directory locale.

```
$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
```

Script di esempio di Linux/UNIX

Creare un file denominato `createUsersAndAssignRoles.sh` copiando lo script seguente. Archivarlo in una directory locale.

```
#!/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
```

Esempio di file input.properties

Per eseguire gli script `createUsersAndAssignRoles`, creare il file `input.properties` e aggiornarlo con le informazioni relative all'ambiente. Salvare il file nella directory in cui è archiviato `createUsersAndAssignRoles.ps1` o `createUsersAndAssignRoles.sh`.

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
```

Tabella 3-2 Parametri di input.properties

Parametro	Descrizione
javahome	Posizione di <code>JAVA_HOME</code> . Solo per Linux/UNIX.
epmautomatescript	Percorso assoluto dell'eseguibile di EPM Automate (<code>epmautomate.sh</code>). Solo per Linux/UNIX.

Tabella 3-2 (Cont.) Parametri di input.properties

Parametro	Descrizione
username	Nome utente di un Amministratore servizi a cui è assegnato anche il ruolo Amministratore del dominio di Identity.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
serviceURL	URL dell'ambiente dal quale generare lo snapshot.
snapshotName	Il nome dello snapshot da creare. Si presume che per creare lo snapshot siano già stati esportati gli artifact di Gruppi e appartenenza utilizzando l'opzione Migrazione.
userspassword	La password iniziale per i nuovi utenti.
resetPassword	Se i nuovi utenti devono reimpostare la password al primo accesso. Impostare <code>true</code> per obbligare i nuovi utenti a modificare la password al primo accesso.

Esecuzione dello script

1. Creare `createUsersAndAssignRoles.ps1` o `createUsersAndAssignRoles.sh` copiando lo script da una sezione precedente.
2. Aggiungere un codice customizzato per eseguire le operazioni indicate di seguito.
 - Estrazione del contenuto dello snapshot.
 - Aggiunta di informazioni sui nuovi utenti al file `HHSS-Shared Services\resource\External Directory\Users.csv`.
 - È inoltre possibile aggiungere informazioni sulle assegnazioni dei ruoli ai nuovi utenti (nel formato Nome, Cognome, E-mail, Login utente) ai file dei ruoli appropriati in `HSS-Shared Services\resource\External Directory\Roles\`.
 - Creazione di un nuovo snapshot con i file aggiornati.
3. Creazione del file `input.properties` e suo salvataggio nella directory in cui è memorizzato lo script `createUsersAndAssignRoles`. Il contenuto di questo file varia a seconda del sistema operativo. Fare riferimento alla sezione [Esempio di file input.properties](#).
Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
4. Avviare lo script.
 - **Windows PowerShell:** eseguire `createUsersAndAssignRoles.ps1`.
 - **Linux/UNIX:** eseguire `./createUsersAndAssignRoles.sh`.

Conteggio del numero di utenti con licenza (utenti assegnati a ruoli)

Utilizzare lo script descritto in questa sezione per generare il Report assegnazioni ruoli per eseguire il conteggio del numero di utenti per un ambiente.

Creare `provisionedUsersCount.bat` copiando lo script seguente.

 **Nota:**

- I parametri di input per l'esecuzione di `provisionedUsersCount.bat` sono `username`, `password/password_file`, `service_url` e `report_file_name`. Ad esempio, al prompt dei comandi, immettere un comando simile al seguente:
`provisionedUsersCount.bat jdoe password.epw https://example.oraclecloud.com myRole_assign.CSV`
- Se la password contiene caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#).

```
@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%.
)

```

Creazione di report di audit di utenti assegnati a ruoli

Utilizzare gli script descritti in questa sezione per automatizzare il processo di creazione di un report di audit per gli utenti assegnati a ruoli predefiniti in un ambiente e, facoltativamente, inviare via e-mail tale report a un destinatario.

Questo report di audit mostra gli utenti assegnati a ruoli o gruppi predefiniti che sono cambiati dopo l'ultima generazione del report. Per creare un report di audit giornaliero, eseguire questo script a cadenza giornaliera.

Creare `provisioningAuditReport.bat` copiando lo script seguente. Questo script batch del wrapper chiama lo script di PowerShell `provisioningAuditReport.ps1`, il codice sorgente per cui viene fornito più avanti in questo scenario.

Nota:

- I parametri di input per l'esecuzione di `provisioningAuditReport.bat` sono: `username`, `password` o `password_file`, `service_url` e `report_email_to_address` (facoltativo, obbligatorio solo se si desidera inviare il report a un indirizzo e-mail).
- Se la password contiene caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#).

```
@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` chiama `provisioningAuditReport.ps1`, che può essere creato copiando lo script seguente.

provisioningAuditReport.ps1 crea il report di audit. Posizionarlo nella stessa directory in cui si trova provisioningAuditReport.bat.

```

$username=$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"

```

Creazione del report di audit sull'assegnazione e revoca dei ruoli

Utilizzare lo script di PowerShell descritto in questa sezione per automatizzare il processo di creazione di un report di audit contenente i dettagli dell'assegnazione e della revoca dei ruoli in un ambiente.

Creare `AuditReportRoleAssignment.bat` copiando lo script seguente. Questo script batch del wrapper richiama lo script di PowerShell `AuditReportRoleAssignment.ps1`, per il quale più avanti in questo scenario viene fornito il codice sorgente.



Nota:

- I parametri di input per l'esecuzione di `AuditReportRoleAssignment.bat` sono: `username`, `password` **or** `password_file` e `service_url`.
- Se la password contiene caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#).

Script: `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 %*
```

Script: 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-auditte-" + $date + ".csv"
$provisionreporttemp="./provreport-auditte-temp.csv"
$provisionreportunique="./provreport-auditte-unique.csv"
$provisionreportbaselineunique="./provreport-auditte-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"

```

Mascheramento dei log di accesso e del report attività per conformità con le leggi sulla privacy

Utilizzare gli script descritti in questa sezione per automatizzare il processo di mascheramento delle informazioni nel report attività o nei log degli accessi in modo conforme alle normative sulla privacy in vigore e, facoltativamente, inviare via e-mail il report a un destinatario.

A causa delle leggi restrittive sulla privacy di alcuni paesi, le informazioni disponibili nei report attività e nei log di accesso potrebbero venire nascoste dagli amministratori del servizio per proteggere la privacy degli utenti.

Utilizzare `anonymizeData.bat` per mascherare le informazioni nei report attività e nei log degli accessi al fine di garantire la conformità con le leggi sulla privacy e,

facoltativamente, effettuare l'invio via e-mail. Per mascherare le informazioni, programmare questo script o una sua variazione utilizzando la utility di pianificazione di Windows in modo che venga eseguito ogni giorno subito dopo il completamento del processo di manutenzione giornaliera per ciascun ambiente.

Utilizzare le fonti di informazione seguenti.

- Utilizzo di report attività e log di accesso per il monitoraggio dell'utilizzo in *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori*
- [Automazione dell'esecuzione degli script](#)

Creare `anonymizeData.bat` manualmente. A tale scopo, copiare lo script Windows fornito nella seguente procedura e programmarlo utilizzando la utility di pianificazione di Windows. È possibile creare ed eseguire script simili specifici della piattaforma se non si utilizza Windows per programmarne l'esecuzione.

`anonymizeData.bat` è uno script wrapper che esegue lo script `anonymizeData.ps1` creato e aggiornato come descritto nella seguente procedura.

Se la password contiene caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#)

1. Creare un file batch (BAT) denominato `anonymizeData.bat` e contenente lo script riportato di seguito, quindi salvarlo in una posizione significativa, ad esempio

`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 %*`

2. Creare uno script PowerShell (PS1) denominato `anonymizeData.ps1` e contenente lo script riportato di seguito, quindi salvarlo in una posizione significativa, ad esempio

`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/$activityreportfile"
"$element"
}
}

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"
$datestampdirexists=Test-Path "$aprrdir/$datestampdir"
$accesslog="$aprrdir/$datestampdir/$accesslogfile"
$activityreport="$aprrdir/$datestampdir/$activityreportfile"

echo "fullpath: $fullpath" >> $logfile
echo "aprrdir: $aprrdir, datestampdir: $datestampdir" >> $logfile
if (!$datestampdirexists) {
    mkdir "$aprrdir/$datestampdir" -ea 0 2>&1 | out-null
    ProcessCommand downloadfile "$accesslog"
    ProcessCommand downloadfile "$activityreport"
    mv "$accesslogfile" "$aprrdir/$datestampdir"
    mv "$activityreportfile" "$aprrdir/$datestampdir"
    AnonymizeData "$aprrdir" "$datestampdir"
    ProcessCommand deletefile "$accesslog"
    ProcessCommand deletefile "$activityreport"
    ProcessCommand uploadfile "$accesslog" "$aprrdir/$datestampdir"
    ProcessCommand uploadfile "$activityreport"
"$aprrdir/$datestampdir"
    } else {
        EchoAndLogMessage "Files in directory $aprrdir/$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. Mediante la utility di pianificazione di Windows, programmare `anonymizeData.bat`. Vedere [Automazione dell'esecuzione degli script](#) per informazioni sulla procedura dettagliata.

È necessario specificare i seguenti valori di parametro per eseguire `anonymizeData.bat`

- Nome utente di un amministratore del servizio

- Password dell'amministratore del servizio o la posizione in cui è disponibile il password file cifrato
- URL dell'ambiente del servizio in cui devono essere mascherati i log di accesso e i report attività
- **Facoltativo:** indirizzo e-mail a cui deve essere inviato il report. Il report viene inviato via e-mail solo se viene specificato questo valore.

Automazione del download dei report attività in un computer locale

Utilizzare lo script descritto in questa sezione per automatizzare lo scaricamento dei report attività da un ambiente in un computer locale.

Utilizzare `syncAprReports.bat` per scaricare i report attività. È possibile programmare il file batch utilizzando la utility di pianificazione di Windows per automatizzare il download dei report attività. Vedere Utilizzo di report attività e log di accesso per il monitoraggio dell'utilizzo in *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori* per informazioni dettagliate sul report attività.

Creare manualmente `syncAprReports.bat` copiando lo script fornito nella seguente procedura e quindi aggiornando i parametri di connessione. Questo script controlla l'ambiente ed esegue il download solo dei report più recenti rispetto a quelli disponibili nella directory di download nel computer locale.

Nota:

- Lo script deve essere eseguito solo in un computer Windows.
- Questo script non scarica il report attività di feedback, che viene generato quando gli utenti sottomettono il feedback.
- Se la password contiene caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#)

1. Creare un file batch (.BAT) denominato `syncAprReports.bat` e contenente lo script riportato di seguito, quindi salvarlo in una posizione significativa, ad esempio `C:\automate_scripts`.

```
@echo off
title APR
setlocal DisableDelayedExpansion

REM To hardcode the values in the script replace %1, %2, %3, and
REM %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

```

2. Modificare `syncAprReports.bat` per impostare i valori per i parametri nella seguente tabella. Questi valori vengono utilizzati per accedere all'ambiente per eseguire il download dei report attività.

Tabella 3-3 Valori di parametri da includere in `syncAprReports.bat`

Parametro	Valore previsto
<code>set apr_dir=%1</code>	Specificare una directory esistente in cui devono essere scaricati i report attività. Esempio: <code>set apr_dir="C:\Oracle\apr"</code>
<code>set username=%2</code>	Nome utente di Oracle Enterprise Performance Management Cloud da utilizzare per eseguire l'accesso all'ambiente per eseguire il download dei report attività. Esempio: <code>set username="ServiceAdmin"</code>
<code>set password=%3</code>	Nome e posizione del file in cui è memorizzata la password cifrata dell'utente specificato dalla variabile <code>username</code> . È inoltre possibile specificare la password dell'utente in testo normale (operazione sconsigliata). Fare riferimento al comando encrypt per informazioni sulla creazione di un password file cifrato. Esempi: <code>set password="C:\mySecuredir\password.epw"</code> <code>set password="Ex@mple1"</code>
<code>set url=%4</code>	URL dell'ambiente. Esempio: <code>set url="https://test-example.stg-pbcs.us1.oraclecloud.com"</code>

3. Utilizzando la utility di pianificazione di Windows programmare `syncAprReports.bat`. Vedere [Automazione dell'esecuzione degli script](#) per informazioni sulla procedura dettagliata.

Scaricamento dei log di accesso da un ambiente

Utilizzare lo script descritto in questa sezione per automatizzare il processo di scaricamento dei log di accesso da un ambiente in un computer locale.

È possibile programmare `syncAccessLog.bat` mediante la utility di pianificazione di Windows per automatizzare lo scaricamento dei file di log. Fare riferimento alla sezione Visualizzazione e scaricamento di report attività e log di accesso nella *Guida introduttiva a*

Oracle Enterprise Performance Management Cloud per gli amministratori per le procedure che consentono di scaricare i log di accesso mediante Gestione applicazioni.

Lo script seguente verifica l'ambiente e scarica solo i file di log più recenti rispetto a quelli disponibili nella directory di scaricamento nel computer locale. Questo è uno script Windows. È possibile creare uno script shell simile per ambienti Linux/UNIX.

1. Creare un file batch (.BAT) denominato `syncAccessLog.bat` contenente lo script riportato di seguito, quindi salvarlo in una posizione significativa, ad esempio `C:\automate_scripts`.

 **Nota:**

Se la password contiene caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#).

```
@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. Modificare `syncAccessLog.bat` per impostare i valori per i parametri nella tabella riportata di seguito. Questi valori vengono utilizzati per accedere all'ambiente allo scopo di scaricare i log di accesso.

Tabella 3-4 Valori di variabili da includere in `syncAccessLog.bat`

Variabile	Valore previsto
<code>set apr_dir=%1</code>	Specificare la directory in cui devono essere scaricati i log di accesso. Esempio: <code>set apr_dir="C:\Oracle\apr"</code>
<code>set username=%2</code>	Nome utente di Oracle Enterprise Performance Management Cloud da utilizzare per eseguire l'accesso all'ambiente per scaricare i log di accesso. Esempio: <code>set username="ServiceAdmin"</code>

Tabella 3-4 (Cont.) Valori di variabili da includere in syncAccessLog.bat

Variabile	Valore previsto
set password=%3	Nome e posizione del file in cui è memorizzata la password cifrata dell'utente specificato dalla variabile username. È inoltre possibile specificare la password dell'utente in testo normale (operazione sconsigliata). Fare riferimento al comando encrypt per informazioni sulla creazione di un password file cifrato. Esempi: set password="C:\mySecuredir\password.epw" set password="P@ssword1"
set url=%4	URL dell'ambiente. Esempio: set url="https://test-cloudpln.pbcs.us1.oraclecloud.com"

- Mediante l'Utilità di pianificazione Windows, programmare syncAccessLog.bat. Vedere [Automazione dell'esecuzione degli script](#) per informazioni sulla procedura dettagliata.

Automazione della clonazione degli ambienti

Utilizzare lo script descritto in questa sezione per automatizzare la clonazione degli ambienti.

Creare un file batch (.bat) o shell (.sh) contenente uno script simile a quello riportato di seguito per clonare un ambiente. I seguenti script di esempio gestiscono le attività indicate:

- Accedere all'ambiente di origine.
- Utilizzare `Artifact Snapshot` (lo snapshot creato durante l'ultima manutenzione giornaliera dell'ambiente di origine) o un altro snapshot disponibile nell'ambiente di origine per convertire l'ambiente target come clone dell'origine.
- Creare facoltativamente gli utenti e le relative assegnazioni di ruoli predefiniti e applicazione corrispondenti a quelli dell'ambiente di origine.
- Modificare facoltativamente l'ora di inizio della manutenzione giornaliera in modo che corrisponda a quella dell'ambiente di origine.
- Scollegarsi.

Per informazioni dettagliate sul processo di clonazione, fare riferimento alla sezione [Clonazione degli ambienti di EPM Cloud](#) nella Guida *Administering Migration for Oracle Enterprise Performance Management Cloud (in lingua inglese)*.

Vedere [Automazione dell'esecuzione degli script](#) per informazioni su come pianificare lo script mediante Utilità di pianificazione di Windows.

Windows

- Creare un file batch (.BAT) denominato `cloneEnvironment.bat` contenente lo script riportato di seguito e salvarlo in una posizione appropriata, ad esempio `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. Modificare `cloneEnvironment.bat` per impostare i valori per i parametri indicati di seguito.

Tabella 3-5 Parametri da impostare in cloneEnvironment.bat

Parametro	Descrizione
usersandpredefinedroles	Impostare il valore di questo parametro su <code>true</code> per clonare gli utenti e le relative assegnazioni dei ruoli predefiniti e applicazione. Per clonare gli utenti e le assegnazioni dei ruoli, l'utente che esegue lo script deve disporre del ruolo Amministratore servizi e Amministratore del dominio di Identity nell'ambiente target.
snapshotname	Nome dello snapshot nell'ambiente di origine che deve essere utilizzato per la clonazione.
dailymaintenancestarttime	Impostare il valore di questo parametro su <code>true</code> per modificare l'ora di inizio della manutenzione giornaliera dell'ambiente clonato in modo che corrisponda a quella dell'ambiente di origine. Impostare questo valore su <code>false</code> per mantenere l'ora di inizio della manutenzione giornaliera corrente dell'ambiente clonato.

3. Creare uno script PowerShell denominato `cloneEnvironment.ps1` contenente lo script riportato di seguito e salvarlo nella directory in cui è stato salvato `cloneEnvironment.bat`, ad esempio `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. Eseguire `cloneEnvironment.bat` utilizzando il comando riportato di seguito.

```
cloneEnvironment.bat "SOURCE USERNAME" "SOURCE PASSWORD FILE" "SOURCE URL" "TARGET USERNAME" "TARGET PASSWORD FILE" "TARGET URL"
```

Ad esempio:

```
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. Creare uno script della shell denominato `cloneEnvironment.sh` contenente lo script riportato di seguito e salvarlo in una posizione appropriata.

```
#!/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
```

2. Modificare `cloneEnvironment.sh` per impostare i valori dei parametri indicati di seguito.

Tabella 3-6 Parametri da impostare in cloneEnvironment.sh

Parametro	Descrizione
epmautomatescript	Percorso assoluto dell'eseguibile di EPM Automate (epmautomate.sh).
javahome	Posizione JAVA_HOME.
usersandpredefinedroles	Impostare il valore di questo parametro su <i>true</i> per clonare gli utenti e le relative assegnazioni dei ruoli predefiniti e applicazione. Per clonare gli utenti e le assegnazioni dei ruoli, l'utente che esegue lo script deve disporre del ruolo Amministratore servizi e Amministratore del dominio di Identity nell'ambiente target.
snapshotname	Nome dello snapshot nell'ambiente di origine che deve essere utilizzato per la clonazione.
dailymaintenancestarttime	Impostare il valore di questo parametro su <i>true</i> per modificare l'ora di inizio della manutenzione giornaliera dell'ambiente clonato in modo che corrisponda a quella dell'ambiente di origine. Impostare questo valore su <i>false</i> per mantenere l'ora di inizio della manutenzione giornaliera corrente dell'ambiente clonato.

3. Eseguire cloneEnvironment.sh.

```
./cloneEnvironment.sh "SOURCE USERNAME" "SOURCE PASSWORD FILE" "SOURCE URL" "TARGET USERNAME" "TARGET PASSWORD FILE" "TARGET URL"
```

Ad esempio:

```
./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.
```

Rimozione dei file superflui da un ambiente

Utilizzare questi script per rimuovere da un ambiente i file superflui.

Questi script consentono di eseguire i passi indicati di seguito.

- Collegamento all'ambiente.
- Elenco dei file e degli snapshot nell'ambiente
- Eliminazione dei file specificati in `input.properties`.
- Disconnessione.

Script di esempio per Windows

Creare un file denominato `removeUnnecessaryFiles.ps1` copiando lo script seguente. Archivarlo in una directory locale.

```
$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
  
```

Script di esempio di Linux/UNIX

Creare un file denominato `removeUnnecessaryFiles.sh` copiando lo script seguente. Archivarlo in una directory locale.

```

#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "${serviceURL}"
${epmautomatescript} listfiles
${epmautomatescript} deletefile "${file1}"
${epmautomatescript} deletefile "${file2}"
${epmautomatescript} logout
  
```

Creazione del file `input.properties`

Per eseguire gli script `removeUnnecessaryFiles`, creare il file `input.properties` e aggiornarlo con le informazioni relative all'ambiente. Salvare il file nella directory in cui è archiviato `removeUnnecessaryFiles.ps1` o `removeUnnecessaryFiles.sh`.

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
  
```

Tabella 3-7 Parametri di input.properties

Parametro	Descrizione
javahome	Posizione di JAVA_HOME. Solo per Linux/UNIX.
epmautomatescript	Percorso assoluto dell'eseguibile di EPM Automate (epmautomate.sh). Solo per Linux/UNIX.
username	Nome utente di un Amministratore servizi a cui è assegnato anche il ruolo Amministratore del dominio di Identity.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
serviceURL	URL dell'ambiente dal quale generare lo snapshot.
file1 e file2	Il nome di un file o di uno snapshot da eliminare dall'ambiente. Se il file non è presente nella cartella outbox, specificarne percorso e nome.

Esecuzione dello script

1. Creare `removeUnnecessaryFiles.ps1` o `removeUnnecessaryFiles.sh` copiando lo script da una sezione precedente.
2. Creare il file `input.properties` e salvarlo nella directory in cui è memorizzato lo script `removeUnnecessaryFiles`. Il contenuto di questo file varia a seconda del sistema operativo. Fare riferimento alla sezione [Creazione del file input.properties](#). Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
3. Avviare lo script.
 - **Windows PowerShell:** eseguire `removeUnnecessaryFiles.ps1`.
 - **Linux/UNIX:** eseguire `./removeUnnecessaryFiles.sh`.

Ricerca e scaricamento di file da un ambiente

Utilizzare lo script di esempio descritto in questa sezione per automatizzare il processo di scaricamento di uno o più file da un ambiente Oracle Enterprise Performance Management Cloud utilizzando una stringa di testo come carattere jolly.

Lo script seguente consente di trovare una corrispondenza tra la stringa specificata come valore del parametro `FILENAME` e i nomi file visualizzati utilizzando il comando `listfiles` e quindi di scaricare automaticamente i file corrispondenti alla stringa.

Accertarsi di assegnare al parametro `FILENAME` la stringa di ricerca appropriata. Ad esempio, `FILENAME="Scheduler Output/epm"` assocerà la stringa `Scheduler Output/epm` ai nomi di file visualizzati nell'output del comando `listfiles` eseguito nell'ambiente per identificare i file da scaricare.

I parametri di input per l'esecuzione di questo script sono: `username`, `password` o `password_file` e `service_url`.



Nota:

Se la password contiene caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#).

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
```

Come ricreare un vecchio ambiente EPM Cloud per gli audit

Utilizzare lo script descritto in questa sezione per creare una soluzione self-service per mantenere aggiornata una libreria di snapshot per l'ambiente Oracle Enterprise Performance Management Cloud in uso. È necessario un ambiente dedicato ai fini dell'aggiornamento e della gestione di una libreria di snapshot aggiornati.

EPM Cloud supporta la compatibilità degli snapshot solo per un ciclo mensile; è possibile migrare gli snapshot di manutenzione dall'ambiente di test all'ambiente di produzione e viceversa. Tuttavia, i requisiti di auditing di alcuni clienti potrebbero

richiedere il ripristino di snapshot di più anni nell'ambiente più recente e l'accesso all'applicazione in un breve periodo di tempo.

È consigliabile programmare l'esecuzione di questo script una volta al mese allo scopo di convertire gli snapshot disponibili e renderli compatibili con l'ultimo livello di patch di EPM Cloud. Oracle consiglia di eseguire lo script dopo il terzo venerdì del mese per assicurarsi di aver risolto tutti i problemi nell'ambiente di produzione.



Nota:

Non è possibile utilizzare questo script per aggiornare gli snapshot di Narrative Reporting, Account Reconciliation e Oracle Enterprise Data Management Cloud.

Funzionamento dello script

Per ogni snapshot memorizzato dal cliente, lo script di aggiornamento completa i seguenti task utilizzando EPM Automate.

1. Accede all'ambiente utilizzando le informazioni contenute nel file `input.properties`.
2. Utilizza il comando `recreate` per ricreare l'ambiente.
3. Importa lo snapshot nell'ambiente.
4. Esegue la manutenzione giornaliera sull'ambiente, con la conseguente conversione dello snapshot nel formato compatibile con il livello patch corrente di EPM Cloud.
5. Scarica `Artifact Snapshot` (lo snapshot di manutenzione) in una cartella. Se è stato ricreato un ambiente 18.05 mediante il caricamento di snapshot da `snapshots/18.05`, `Artifact Snapshot` viene scaricato in `snapshots/18.06`.
6. Invia tramite e-mail i risultati ottenuti ricreando vecchi ambienti a un indirizzo e-mail, se specificato.

Esecuzione dello script

1. Creare il file `input.properties` e aggiornarlo con le informazioni relative all'ambiente in uso. Salvare il file in una directory locale. La directory viene indicata come `parentsnapshotdirectory` nel corso di questa descrizione. Il contenuto di questo file varia a seconda del sistema operativo.
Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire gli script.
2. Creare lo script `upgradeSnapshots.ps1` (Windows PowerShell) o `upgradeSnapshots.sh` (Linux/UNIX) e salvarlo in `parentsnapshotdirectory` dove si trova il file `input.properties`.
3. Creare una sottodirectory, ad esempio, `snapshots`, in `parentsnapshotdirectory`.
4. Nella directory creata nel passo precedente (`snapshots`), creare una sottodirectory per lo snapshot mensile da convertire in modo da renderlo compatibile con il livello patch corrente di EPM Cloud. Denominare la directory utilizzando il formato `AA.MM`; ad esempio, `18.05` per la directory in cui memorizzare gli snapshot di maggio 2018.
5. Copiare gli snapshot nella sottodirectory appropriata. Ad esempio, copiare gli snapshot per maggio 2018 in `snapshots/18.05`.

6. Avviare lo script.

- Linux/UNIX: eseguire `./upgradeSnapshots.sh`.
- Windows PowerShell: eseguire `upgradeSnapshots.ps1`.

Windows

Creare `input.properties` e lo script `upgradeSnapshots.ps1` copiando gli script in questa sezione.

Creazione di `input.properties`

```
username=exampleAdmin
userpassword=examplePassword
serviceurl=exapleURL
proxyserverusername=proxyServerUserName
proxyserverpassword=proxyPassword
proxyserverdomain=proxyDoamin
parentsnapshotdirectory=C:/some_directory/snapshots
emailtoaddress=exampleAdmin@oracle.com
```

Aggiornamento di `input.properties`



Nota:

Se l'autenticazione a livello di server proxy non è abilitata per l'ambiente di rete Windows, rimuovere le proprietà `proxyserverusername`, `proxyserverpassword` e `proxyserverdomain` dal file `input.properties`.

Tabella 3-8 Parametri di `input.properties`

Parametro	Descrizione
<code>username</code>	Nome utente di un amministratore del servizio.
<code>userpassword</code>	Password dell'amministratore del servizio .
<code>serviceurl</code>	URL dell'ambiente utilizzato per questa attività.
<code>proxyserverusername</code>	Il nome utente per l'autenticazione di una sessione sicura con il server proxy che controlla l'accesso a Internet.
<code>proxyserverpassword</code>	La password per autenticare l'utente con il server proxy.
<code>proxyserverdomain</code>	Il nome del dominio definito per il server proxy.
<code>parentsnapshotdirectory</code>	Percorso assoluto della directory da utilizzare come directory padre della directory in cui sono memorizzati gli snapshot da elaborare. Utilizzare le barre (/) come separatori di directory.
<code>emailtoaddress</code>	(Facoltativo) Indirizzo e-mail a cui devono essere inviati i risultati ottenuti ricreando vecchi ambienti. I risultati vengono inviati vie e-mail solo se viene specificato questo valore. Esempio: <code>john.doe@example.com</code>



Nota:

Se la password contiene caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#).

Creazione di upgradeSnapshots.ps1

Utilizzare questo script di esempio per creare 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"
    $intlatelyear=[int]$latestyear
    $intlatelyearmonth=[int]$latestmonth

    if ($intlatelyearmonth -eq 12) {
        $intnextmonth=1
        $intnextyear=$intlatelyear+1
    } else {
        $intnextmonth=$intlatelyearmonth+1
        $intnextyear=$intlatelyear
    }

    $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('/')[0]
    LogMessage "snapshotfilename: $snapshotfilename"
    $snapshotname=$snapshotfilename.split('.')[0]
    LogMessage "snapshotname: $snapshotname"

    ProcessCommand
    "login $username $userpassword $serviceurl $proxyserverusername $proxyserverp
    assword $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('/')
    $logfilefilename=$elements[-1]

    if (${emailtoaddress} -match "@") {
        epmautomate.bat login ${username} ${userpassword} ${serviceurl}
        epmautomate.bat uploadFile "$logfile"
        epmautomate.bat sendMail $emailtoaddress "Recreating An Old
EPM Cloud Environment results" Body="The results of recreating an old
EPM Cloud Environment are attached." Attachments=$logfilefilename
        epmautomate.bat deleteFile "$logfilefilename"
        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

Creare `upgradeSnapshots.sh` e `input.properties` copiando i seguenti script.

Creazione di `input.properties` per Linux/UNIX

 **Nota:**

Se la rete non è configurata per l'accesso a Internet tramite un server proxy, rimuovere le proprietà `proxyserverusername`, `proxyserverpassword` e `proxyserverdomain` dal file `input.properties`.

```
username=exampleAdmin
userpassword=examplePassword
serviceurl=exapleURL
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
```

Aggiornamento di `input.properties`

Tabella 3-9 Parametri di `input.properties`

Parametro	Descrizione
<code>username</code>	Nome utente di un amministratore del servizio.
<code>userpassword</code>	Password dell'amministratore del servizio .
<code>serviceurl</code>	URL dell'ambiente utilizzato per questa attività.
<code>proxyserverusername</code>	Il nome utente per l'autenticazione di una sessione sicura con il server proxy che controlla l'accesso a Internet.
<code>proxyserverpassword</code>	La password per autenticare l'utente con il server proxy.
<code>proxyserverdomain</code>	Il nome del dominio definito per il server proxy.
<code>jdkdir</code>	Posizione di <code>JAVA_HOME</code> .
<code>epmautomatescript</code>	Percorso assoluto dell'eseguibile di EPM Automate (<code>epmautomate.sh</code>).
<code>parentsnapshotdirectory</code>	Percorso assoluto della directory da utilizzare come directory padre della directory in cui è memorizzato lo snapshot da elaborare.
<code>emailtoaddress</code>	(Facoltativo) Indirizzo e-mail a cui devono essere inviati i risultati ottenuti ricreando vecchi ambienti.

 **Nota:**

Se la password contiene caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#).

Creazione di `upgradeSnapshots.sh`

Utilizzare questo script di esempio per creare `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
}

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

```

Automazione dell'audit e della conformità degli accessi al database

Utilizzare gli script shell di PowerShell e Bash descritti in questa sezione per usare i comandi EPM Automate per raccogliere i dati di audit e conformità relativi all'accesso manuale al database.

È possibile utilizzare i seguenti script per completare i task elencati di seguito:

- Scaricare il report attività per il giorno corrente
- Analizzare il report per determinare se l'accesso manuale al database è stato registrato per l'ambiente in uso
- Creare `./reports/dataAccessAuditReport.txt` in relazione alla directory da cui si esegue lo script. Nel report sono riportati l'ora degli accessi al database e il comando SQL eseguito. Si tratta di un file cumulativo, che riporta le informazioni più aggiornate nella parte superiore dell'elenco. Le informazioni disponibili includono:
 - Data e ora di generazione del report
 - Dettagli relativi agli accessi al database, se disponibili. Gli accessi al database con una richiesta di servizio e gli accessi al database senza una richiesta di servizio sono riportati in sezioni distinte. Se gli accessi manuali al database non sono registrati nel report attività, nel report è contenuto il messaggio `No SQL statements executed`.
 - Inviare facoltativamente il report a un indirizzo e-mail specificato.

Per automatizzare l'audit e la conformità dell'accesso ai dati, procedere come segue.

1. Copiare uno degli script riportati nelle seguenti sezioni in un file e salvare il file nel file system in uso. Denominare il file `parseActivityReport.ps1` (per Windows, fare riferimento alla sezione [Script di PowerShell \(parseActivityReport.ps1\)](#)) o `parseActivityReport.sh` (per Linux/UNIX, fare riferimento alla sezione [Script shell Bash \(parseActivityReport.sh\)](#)).
2. **Solo Windows:** creare un file batch denominato `parseActivityReport.bat` copiando lo script seguente in un file. Salvare il file nella stessa directory in cui è memorizzato il file `parseActivityReport.ps1`.

```
@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. Modificare `parseActivityReport.bat` (Windows) o `parseActivityReport.sh` (Linux/UNIX) per impostare i valori dei parametri descritti nella tabella riportata di seguito.

Tabella 3-10 Valori di variabili da includere negli script

Variabile	Descrizione
<code>epmuser</code>	Nome utente di un amministratore del servizio Esempi: Windows: <code>set epmuser="jDoe"</code> Linux/UNIX: <code>epmuser="jDoe"</code>
<code>epmpassword</code>	Password dell'utente con ruolo Amministratore servizi o posizione in cui è disponibile il password file cifrato. Fare riferimento al comando encrypt per informazioni sulla creazione di un password file cifrato. Se la password contiene caratteri speciali, fare riferimento alla sezione Gestione dei caratteri speciali . Esempi: Windows: <code>set epmpassword = "Example"</code> Linux/UNIX: <code>epmpassword="Example"</code>
<code>epmurl</code>	URL dell'ambiente Oracle Enterprise Performance Management Cloud. Esempi: Windows: <code>set epmurl="https://example.oraclecloud.com"</code> Linux/UNIX: <code>epmurl="https://example.oraclecloud.com"</code>
<code>report_email_to_address</code>	(Facoltativo) Indirizzo e-mail a cui deve essere inviato il report. Il report viene inviato via e-mail solo se viene specificato questo valore. Esempio: <code>john.doe@example.com</code>

4. **Solo per `parseActivityReport.sh`:** assicurarsi che i seguenti valori siano impostati correttamente per il sistema in uso:
 - `JAVA_HOME`
 - Posizione di `epmautomatescript.sh` mediante l'aggiornamento del valore della direttiva `epmautomatescript`
5. Utilizzare una utility di programmazione disponibile nel sistema operativo in uso per programmare l'esecuzione di `parseActivityReport.bat` (che esegue `parseActivityReport.ps1`) o di `parseActivityReport.sh` una volta al giorno. Fare riferimento alla sezione [Automazione dell'esecuzione degli script](#).

Script di PowerShell (`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

```

Script shell Bash (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="<EPM USER>"
epmpwd="<EPM PASSWORD>"
epmurl="<EPM URL>"
reportemailtoaddress="<EMAIL ADDRESS>"

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}
    fi
}
```

```

        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
}

```

```
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
```

Replica degli utenti e delle assegnazioni dei ruoli predefiniti

Gli script descritti in questa sezione consentono di migrare gli utenti e le assegnazioni dei ruoli predefiniti da un ambiente a un altro.

Informazioni sugli script

Utilizzare due script distinti: uno per replicare gli utenti nei domini di Identity e l'altro per replicare le assegnazioni dei ruoli predefiniti degli utenti. L'ordine di esecuzione degli script è riportato di seguito.

- Eseguire lo script per replicare gli utenti (`replicateusers`) e verificare che tutti gli utenti siano stati creati nel dominio di Identity target. L'utente che esegue questo script deve disporre dei ruoli di Amministratore dominio di Identity e Amministratore servizi in entrambi gli ambienti.
- Eseguire lo script per replicare le assegnazioni dei ruoli (`replicatepredefinedroles`).

Nota:

- Se le password contengono caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#).
- Gli script in questa sezione funzionano solo per i ruoli predefiniti: Amministratore servizi, Utente avanzato, Utente e Visualizzatore.

Esecuzione degli script

Per informazioni sulla creazione degli script e dei file batch richiesti, fare riferimento alle sezioni riportate di seguito.

- [Replica degli utenti da un dominio di Identity a un altro](#)
- [Replica delle assegnazioni dei ruoli predefiniti da un ambiente a un altro](#)

Procedura per Windows

1. Creare `replicateusers.bat`, `replicateusers.ps1`, `replicatepredefinedroles.bat` e `replicatepredefinedroles.ps1` e salvarli in una directory locale nella quale si dispone di privilegi di scrittura ed esecuzione.
2. Aggiornare i file batch con informazioni per gli ambienti di origine e target e il server proxy Internet, se necessario.
3. Eseguire `replicateusers.bat` che esegue `replicateusers.ps1`. È necessario specificare la password predefinita da assegnare agli utenti replicati come parametro della riga comandi, come riportato di seguito.
`replicateusers.bat Pwd_for_users`
 Se la password contiene caratteri speciali, utilizzare il carattere di escape appropriato. Fare riferimento alla sezione [Gestione dei caratteri speciali](#).
4. Eseguire `replicatepredefinedroles.bat` per creare assegnazioni dei ruoli identiche a quelle che esistono nell'ambiente di origine.

Procedura per Linux/UNIX

1. Creare gli script `replicateusers.sh` e `replicatepredefinedroles.sh` e salvarli in una directory locale nella quale si dispone di privilegi di scrittura ed esecuzione.
2. Aggiornare `replicateusers.sh` e `replicatepredefinedroles.sh` con informazioni per gli ambienti di origine e target e il server proxy Internet, se necessario.
3. Eseguire `replicateusers.sh`. È necessario specificare la password predefinita da assegnare agli utenti replicati come parametro della riga comandi, come riportato di seguito.

```
./replicateusers.sh Pwd_for_users
```

Se la password contiene caratteri speciali, utilizzare il carattere di escape appropriato. Fare riferimento alla sezione [Gestione dei caratteri speciali](#).
4. Eseguire lo script `replicatepredefinedroles.sh` per creare assegnazioni dei ruoli identiche a quelle che esistono nell'ambiente di origine.

Replica degli utenti da un dominio di Identity a un altro

Utilizzare gli script in questa sezione per clonare gli utenti di un dominio di Identity in un altro dominio di Identity. L'utente che esegue questi script deve disporre dei ruoli Amministratore dominio di Identity e Amministratore servizi negli ambienti di origine e target.

Windows

Creare `replicateusers.bat` e `replicateusers.ps1` copiando gli script in questa sezione.

1. Creare `replicateusers.ps1` copiando lo script riportato di seguito.

```
# 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. Creare replicateusers.bat copiando lo script riportato di seguito.

```

@ECHO OFF
SET thisdir=%~dp0
SET scriptpath=%thisdir%replicateusers.ps1
SET paramRequiredMessage=Syntax: replicateusers.bat "USER_PASSWORD"

REM USER DEFINED VARIABLES
REM -----
set epmsource="<EPM USER FOR SOURCE ENVIRONMENT>"
set epmpwdsource="<EPM PASSWORD FOR SOURCE ENVIRONMENT>"
set epurlsource="<EPM URL FOR SOURCE ENVIRONMENT>"
set epidentitydomainsource="<EPM IDENTITY DOMAIN FOR SOURCE ENVIRONMENT>"
set epusertarget="<EPM USER FOR TARGET ENVIRONMENT>"
set eppwdtarget="<EPM PASSWORD FOR TARGET ENVIRONMENT>"
set epurltarget="<EPM URL FOR TARGET ENVIRONMENT>"
set epidentitydomaintarget="<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%' -epmsource '%epmsource%' -epmpwdsource
'%epmpwdsource%' -epurlsource '%epurlsource%' -
epidentitydomainsource '%epidentitydomainsource%' -epusertarget
'%epusertarget%' -eppwdtarget '%eppwdtarget%' -epurltarget
'%epurltarget%' -epidentitydomaintarget
'%epidentitydomaintarget%' -proxyserverusername
'%proxyserverusername%' -proxyserverpassword
'%proxyserverpassword%' -proxyserverdomain '%proxyserverdomain%' -
userpassword '%~1' -resetpassword '%resetpassword%' -emailtoaddress
'%emailtoaddress%'"

```

3. Aggiornare `replicateusers.bat`. Fare riferimento alla tabella riportata di seguito per le informazioni sui valori da specificare.

Parametro	Descrizione
<code>epmusersource</code>	Nome utente di un utente con i ruoli Amministratore dominio di Identity e Amministratore servizi nell'ambiente di origine. Esempi: Windows: <code>set epmusersource="jDoe"</code> Linux/UNIX: <code>epmusersource="jDoe"</code>
<code>epmpwdsource</code>	La password dell'utente o il percorso assoluto del file delle password cifrate. Esempi: Windows: <code>set epmpwdsource="Example"</code> Linux/UNIX: <code>epmpwdsource="Example"</code>
<code>epmurlsource</code>	URL dell'ambiente da cui devono essere copiati gli utenti. Esempi: Windows: <code>set epmurlsource="https://example.oraclecloud.com"</code> Linux/UNIX: <code>epmurlsource="https://example.oraclecloud.com"</code>
<code>epmidentitydomainsource</code>	Nome del dominio di Identity utilizzato dall'ambiente di origine. Esempi: Windows: <code>set epmidentitydomainsource="example_source_dom"</code> Linux/UNIX: <code>epmidentitydomainsource="example_source_dom"</code>
<code>epmusertarget</code>	Nome utente di un utente con i ruoli Amministratore dominio di Identity e Amministratore servizi nell'ambiente target. Esempi: Windows: <code>set epmusertarget="John.Doe"</code> Linux/UNIX: <code>set epmusertarget="John.Doe"</code>
<code>epmpwdtarget</code>	La password dell'utente o il percorso assoluto del file delle password cifrate. Esempi: Windows: <code>set epmpwdtarget="Example1"</code> Linux/UNIX: <code>epmpwdtarget="Example1"</code>
<code>epmurltarget</code>	URL dell'ambiente in cui devono essere creati gli utenti. Esempi: Windows: <code>set epmurltarget="https://example.oraclecloud.com"</code> Linux/UNIX: <code>epmurltarget="https://example.oraclecloud.com"</code>
<code>epmidentitydomaintarget</code>	Nome del dominio di Identity utilizzato dall'ambiente target. Esempi: Windows: <code>set epmidentitydomaintarget="example_source_dom"</code> Linux/UNIX: <code>epmidentitydomaintarget="example_target_dom"</code>

Parametro	Descrizione
proxyserverusername	<p>Il nome utente per l'autenticazione di una sessione sicura con il server proxy che controlla l'accesso a Internet. Eliminare tutte le occorrenze di questa proprietà, se non utilizzate.</p> <p>Esempi: Windows: set proxyserverusername="Example" Linux/UNIX: proxyserverusername="Example"</p>
proxyserverpassword	<p>La password per autenticare l'utente con il server proxy. Eliminare tutte le occorrenze di questa proprietà, se non utilizzate.</p> <p>Esempi: Windows: set proxyserverpassword="examplePwd" Linux/UNIX: proxyserverpassword="examplePwd"</p>
proxyserverdomain	<p>Il nome del dominio definito per il server proxy. Eliminare tutte le occorrenze di questa proprietà, se non utilizzate.</p> <p>Esempi: Windows: set proxyserverdomain="exampleDom" Linux/UNIX: proxyserverdomain="exampleDom"</p>
emailtoaddress	<p>(Facoltativo) Indirizzo e-mail a cui deve essere inviato il report assegnazioni ruoli. Il report viene inviato via e-mail solo se viene specificato questo valore.</p> <p>Esempio: emailtoaddress=john.doe@example.com</p>

Linux/UNIX

1. Creare `replicateusers.sh` copiando lo script riportato di seguito.

```
#!/bin/sh

userpassword="$1"

# USER DEFINED VARIABLES
#-----
javahome=""
epmautomatescript=""
epmusersource=""
epmpwdsource=""
epmurlsource=""
epmidentitydomainsource=""
epmusertarget=""
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. Aggiornare `replicateusers.sh`. Fare riferimento alla tabella precedente per le informazioni sui valori da specificare. Inoltre, è necessario specificare i valori per le proprietà riportate di seguito.
 - `javahome`: il percorso assoluto alla directory dove è installato Java.
 - `epmautomatescript`: posizione di `epmautomatescript.sh`; ad esempio, `epmautomatescript="/home/user1/epmautomate/bin/epmautomate.sh"`

Replica delle assegnazioni dei ruoli predefiniti da un ambiente a un altro

Utilizzare gli script descritti in questa sezione per clonare le assegnazioni dei ruoli predefiniti da un ambiente a un altro. L'utente che esegue questi script deve disporre del ruolo Amministratore servizi in entrambi gli ambienti.

Nota:

In caso di utilizzo della versione in formato PDF di questo documento: per evitare le interruzioni di riga e le informazioni dei piè di pagina che renderebbero inutilizzabili questi script, copiare gli script dalla [Versione HTML di questo argomento](#).

Windows

1. Creare `replicatepredefineroles.ps1` copiando lo script riportato di seguito.

```
# 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 ${epmuserstarget} ${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. Creare `replicatepredefineroles.bat` copiando lo script riportato di seguito.

```
@ECHO OFF
SET thisdir=%~dp0
SET scriptpath=%thisdir%replicatepredefinedroles.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 epidentitydomainsource="<EPM IDENTITY DOMAIN FOR SOURCE ENVIRONMENT>"
set epusertarget="<EPM USER FOR TARGET ENVIRONMENT>"
set epmpwdtarget="<EPM PASSWORD FOR TARGET ENVIRONMENT>"
set epurltarget="<EPM URL FOR TARGET ENVIRONMENT>"
set epidentitydomaintarget="<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%' -
epusersource '%epusersource%' -epmpwdsource '%epmpwdsource%' -
epurlsource '%epurlsource%' -epidentitydomainsource
'%epidentitydomainsource%' -epusertarget '%epusertarget%' -
epmpwdtarget '%epmpwdtarget%' -epurltarget '%epurltarget%' -
epidentitydomaintarget '%epidentitydomaintarget%' -proxyserverusername
'%proxyserverusername%' -proxyserverpassword '%proxyserverpassword%' -
proxyserverdomain '%proxyserverdomain%' -emailtoaddress
'%emailtoaddress%'"
```

3. Aggiornare `replicatepredefineroles.bat` in base alle esigenze. Fare riferimento alla tabella che segue per le informazioni sui valori da impostare per le proprietà in questo file.

Aggiornamento di `replicatepredefineroles.bat`

Parametro	Descrizione
<code>epusersource</code>	Nome utente di un utente con i ruoli Amministratore dominio di Identity e Amministratore servizi nell'ambiente di origine. Esempi: Windows: <code>set epusersource="jDoe"</code> Linux/UNIX: <code>epusersource="jDoe"</code>
<code>epmpwdsource</code>	La password dell'utente o il percorso assoluto del file delle password cifrate. Esempi: Windows: <code>set epmpwdsource="Example"</code> Linux/UNIX: <code>epmpwdsource="Example"</code>

Parametro	Descrizione
epmurlsource	<p>URL dell'ambiente da cui devono essere copiati gli utenti.</p> <p>Esempi:</p> <p>Windows: set epmurlsource="https://example.oraclecloud.com"</p> <p>Linux/UNIX: epmurlsource="https://example.oraclecloud.com"</p>
epmidentitydomainsource	<p>Nome del dominio di Identity utilizzato dall'ambiente di origine.</p> <p>Esempi:</p> <p>Windows: set epmidentitydomainsource="example_source_dom"</p> <p>Linux/UNIX: epmidentitydomainsource="example_source_dom"</p>
epmusertarget	<p>Nome utente di un utente con i ruoli Amministratore dominio di Identity e Amministratore servizi nell'ambiente target.</p> <p>Esempi:</p> <p>Windows: set epmusertarget="John.Doe"</p> <p>Linux/UNIX: set epmusertarget="John.Doe"</p>
epmpwdtarget	<p>La password dell'utente o il percorso assoluto del file delle password cifrate.</p> <p>Esempi:</p> <p>Windows: set epmpwdtarget="Example1"</p> <p>Linux/UNIX: epmpwdtarget="Example1"</p>
epmurltarget	<p>URL dell'ambiente in cui devono essere creati gli utenti.</p> <p>Esempi:</p> <p>Windows: set epmurltarget="https://example.oraclecloud.com"</p> <p>Linux/UNIX: epmurltarget="https://example.oraclecloud.com"</p>
epmidentitydomaintarget	<p>Nome del dominio di Identity utilizzato dall'ambiente target.</p> <p>Esempi:</p> <p>Windows: set epmidentitydomaintarget="example_target_dom"</p> <p>Linux/UNIX: epmidentitydomaintarget="example_target_dom"</p>
proxyserverusername	<p>Il nome utente per l'autenticazione di una sessione sicura con il server proxy che controlla l'accesso a Internet. Eliminare tutte le occorrenze di questa proprietà, se non utilizzate.</p> <p>Esempi:</p> <p>Windows: set proxyserverusername="Example"</p> <p>Linux/UNIX: proxyserverusername="Example"</p>
proxyserverpassword	<p>La password per autenticare l'utente con il server proxy. Eliminare tutte le occorrenze di questa proprietà, se non utilizzate.</p> <p>Esempi:</p> <p>Windows: set proxyserverpassword="examplePwd"</p> <p>Linux/UNIX: proxyserverpassword="examplePwd"</p>

Parametro	Descrizione
proxyserverdomain	Il nome del dominio definito per il server proxy. Eliminare tutte le occorrenze di questa proprietà, se non utilizzate. Esempi: Windows: set proxyserverdomain="exampleDom" Linux/UNIX: proxyserverdomain="exampleDom"
emailtoaddress	(Facoltativo) Indirizzo e-mail a cui deve essere inviato il report assegnazioni ruoli. Il report viene inviato via e-mail solo se viene specificato questo valore. Esempio: emailtoaddress=john.doe@example.com

Linux/UNIX

1. Creare `replicatepredefineroles.sh` copiando lo script riportato di seguito.

```
#!/bin/sh

# USER DEFINED VARIABLES
#-----
javahome="<<JAVA HOME>"
epmautomatescript="<<EPM AUTOMATE SCRIPT LOCATION>"
epmusersource="<<EPM USER FOR SOURCE ENVIRONMENT>"
epmpwdsource="<<EPM PASSWORD FOR SOURCE ENVIRONMENT>"
epmurlsource="<<EPM URL FOR SOURCE ENVIRONMENT>"
epmidentitydomainsource="<<EPM IDENTITY DOMAIN FOR SOURCE ENVIRONMENT>"
epmusertarget="<<EPM USER FOR TARGET ENVIRONMENT>"
epmpwdtarget="<<EPM PASSWORD FOR TARGET ENVIRONMENT>"
epmurltarget="<<EPM URL FOR TARGET ENVIRONMENT>"
epmidentitydomaintarget="<<EPM IDENTITY DOMAIN FOR TARGET ENVIRONMENT>"
proxyserverusername="<<PROXY SERVER USER NAME>"
proxyserverpassword="<<PROXY SERVER PASSWORD>"
proxyserverdomain="<<PROXY SERVER DOMAIN>"
emailtoaddress="<<EMAIL TO ADDRESS>"
#-----

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
}
```

```

fi
echo "Downloading role assignment report"
${epmautomatescript} downloadfile ${roleassignmentreport}
${epmautomatescript} deletedefile ${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
done < ${roleassignmentreport}

# write header line
for f in role-*.csv
do
    sed -i '1iUser 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. Aggiornare `replicatepredefineroles.sh`. Fare riferimento alla tabella precedente per le informazioni sui valori da specificare. Inoltre, è necessario specificare i valori per le proprietà riportate di seguito.

- `javahome`: il percorso assoluto alla directory dove è installato Java.
- `epmautomatescript`: posizione di `epmautomatescript.sh`; ad esempio, `epmautomatescript="/home/user1/epmautomate/bin/epmautomate.sh"`

Creazione di una cadenza di aggiornamento trimestrale di EPM Cloud

Utilizzare questi script per creare una soluzione self-service per ignorare gli aggiornamenti in modo che gli ambienti Oracle Enterprise Performance Management Cloud vengano aggiornati con frequenza trimestrale con un ciclo di test quindicinale. In questo caso gli ambienti di produzione vengono aggiornati due settimane dopo gli ambienti di test.

Questo script può essere utilizzato anche per saltare l'aggiornamento a mesi alterni, se necessario. Per impostazione predefinita, EPM Cloud applica un aggiornamento mensile agli

ambienti. Utilizzare il comando [skipUpdate](#) per ignorare l'applicazione degli aggiornamenti mensili a un ambiente o per visualizzare le richieste correnti di differimento degli aggiornamenti. È possibile automatizzare l'esecuzione manuale dei comandi `skipUpdate` utilizzando gli script inclusi in questa sezione. Questi script automatizzano il processo di differimento degli aggiornamenti in modo che gli aggiornamenti vengano applicati ogni trimestre o a mesi alterni.

 **Note:**

1. Non è possibile saltare gli aggiornamenti per più di due mesi consecutivi. Lo script genera ad esempio un errore se si tenta di aggiornare un ambiente EPM Cloud solo a febbraio, giugno e novembre.
2. Tutti gli aggiornamenti che si sono resi disponibili nel periodo intermedio vengono applicati all'ambiente con l'aggiornamento successivo. Si supponga ad esempio di utilizzare questo script per programmare aggiornamenti trimestrali da applicare solo a febbraio, maggio, agosto e novembre. In questo caso, l'aggiornamento di maggio ad esempio applicherà all'ambiente tutti gli aggiornamenti mensili e le relative patch applicabili di EPM Cloud rilasciati dopo l'aggiornamento di febbraio. È possibile che il processo di manutenzione richieda più tempo del solito quando viene applicato l'aggiornamento.
3. Questo script imposta la cadenza di aggiornamento solo per un trimestre. Eseguire questo script ogni mese in modo che la cadenza di aggiornamento venga configurata per l'intero anno.

- [Script per Windows e istruzioni](#)
- [Script per UNIX/Linux e istruzioni](#)
- [Script Groovy](#)

Esecuzione dello script

1. Per eseguire gli script per Windows e Linux/UNIX, procedere come segue.
 - a. Creare il file `input.properties` e aggiornarlo con le informazioni relative all'ambiente in uso. Salvare il file in una directory locale. Il contenuto di questo file varia a seconda del sistema operativo. Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
 - b. Creare lo script `skip_update.ps1` (Windows PowerShell) o lo script Bash `skip_update.sh` (Linux/UNIX) e salvarlo nella directory in cui si trova il file `input.properties`.
 - c. Avviare lo script.
 - Linux/UNIX: eseguire `./skip_update.sh`.
 - Windows PowerShell: eseguire `skip_update.ps1`.
2. Per eseguire lo script Groovy, utilizzare la schermata Groovy in un processo aziendale EPM Cloud o automatizzare l'esecuzione dello script mediante [runBusinessRule](#). Per informazioni sull'esecuzione dello script Groovy mediante

EPM Automate, fare riferimento alla sezione [Running Commands without Installing EPM Automate \(in lingua inglese\)](#).

Script per Windows e istruzioni

Creare `input.properties` e `skip_update.ps1` copiando gli script di questa sezione.

1. Creare il file `input.properties` copiando lo script riportato di seguito.

```
username=exampleAdmin
password=examplePassword.epw
url=exampleURL
updatemonths=02,05,08,11
```

2. Aggiornare il file `input.properties` specificando i valori dei parametri.

Table 3-11 Parametri di `input.properties`

Parametro	Descrizione
username	Nome utente di un amministratore del servizio.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
url	URL dell'ambiente in cui si desidera impostare la cadenza di aggiornamento non mensile.
updatemonths	Applicare un elenco separato da virgole con i mesi in cui viene aggiornato Oracle Enterprise Performance Management Cloud all'ambiente identificato dal parametro <code>url</code> . Ad esempio, <code>updatemonths=02,05,08,11</code> . I mesi devono essere specificati con due cifre: da 01 per gennaio fino a 12 per dicembre. Assicurarsi di anteporre uno zero per i mesi da gennaio a settembre. Lo script tenta di eseguire il comando skipUpdate per i mesi non inclusi nel valore del parametro <code>updatemonths</code> . Se ad esempio si specifica <code>updatemonths=02,05,08,11</code> , lo script tenta di impostare i flag per saltare gli aggiornamenti per gennaio, marzo, aprile, giugno, luglio, settembre, ottobre e dicembre in modo che gli aggiornamenti vengano eseguiti solo a febbraio, maggio, agosto e novembre.

3. Creare `skip_updates.ps1` copiando lo script riportato di seguito.

```
# 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] $global:monthsarrfromcurrent[$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]
    $thisupdatemonthhint=[int]$thisupdatemonth
    $nextupdatemonth=$updatemonthsarr[$nextint]
    $nextupdatemonthhint=[int]$nextupdatemonth

    if (${nextupdatemonth} -eq "") {
      $nextupdatemonth=$updatemonthsarr[0]
      $nextupdatemonthhint=[int]$nextupdatemonth
    }

    compareUpdateMonths $thisupdatemonthhint $nextupdatemonthhint
  }
}

```

```
validateUpdateMonths
populateFromCurrentArrays
processSkipUpdates
```

Script per UNIX/Linux e istruzioni

Creare `input.properties` e `skip_update.sh` copiando gli script di questa sezione.

1. Creare il file `input.properties` copiando lo script riportato di seguito.

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
password=examplePassword.epw
url=exampleURL
updatemonths=02,05,08,11
```

2. Aggiornare il file `input.properties` specificando i valori dei parametri.

Table 3-12 Parametri di `input.properties`

Parametro	Descrizione
javahome	Posizione di <code>JAVA_HOME</code> .
epmautomatescript	Percorso assoluto dell'eseguibile di EPM Automate (<code>epmautomate.sh</code>).
username	Nome utente di un amministratore del servizio.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
url	URL dell'ambiente in cui si desidera impostare la cadenza di aggiornamento non mensile.
updatemonths	Applicare un elenco separato da virgole con i mesi in cui viene aggiornato Oracle Enterprise Performance Management Cloud all'ambiente identificato dal parametro <code>url</code> . Ad esempio, <code>updatemonths=02,05,08,11</code> . I mesi devono essere specificati con due cifre. Anteporre uno zero per i mesi da gennaio a settembre. Lo script tenta di eseguire il comando <code>skipUpdate</code> per i mesi non inclusi nel valore del parametro <code>updatemonths</code> . Se ad esempio si specifica <code>updatemonths=02,05,08,11</code> , lo script tenta di impostare i flag per saltare gli aggiornamenti per gennaio, marzo, aprile, giugno, luglio, settembre, ottobre e dicembre in modo che gli aggiornamenti vengano eseguiti solo a febbraio, maggio, agosto e novembre.

3. Creare `skip_updates.sh` copiando lo script riportato di seguito.

```
#!/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

```

Script Groovy

Se le password contengono caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#). Assicurarsi inoltre di sostituire i valori dei parametri elencati di seguito a seconda degli ambienti.

Table 3-13 Parametri da modificare

Parametro	Descrizione
user	Nome utente di un amministratore del servizio.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
url	URL dell'ambiente in cui si desidera impostare la cadenza di aggiornamento non mensile.

Table 3-13 (Cont.) Parametri da modificare

Parametro	Descrizione
updatemonths	<p>Applicare un elenco separato da virgole con i mesi in cui viene aggiornato Oracle Enterprise Performance Management Cloud all'ambiente identificato dal parametro url. Ad esempio, updatemonths=02,05,08,11.</p> <p>I mesi devono essere specificati con due cifre: da 01 per gennaio fino a 12 per dicembre. Assicurarsi di anteporre uno zero per i mesi da gennaio a settembre. Lo script tenta di eseguire il comando skipUpdate per i mesi non inclusi nel valore del parametro updatemonths. Se ad esempio si specifica updatemonths=02,05,08,11, lo script tenta di impostare i flag per saltare gli aggiornamenti per gennaio, marzo, aprile, giugno, luglio, settembre, ottobre e dicembre in modo che gli aggiornamenti vengano eseguiti solo a febbraio, maggio, agosto e novembre.</p>

```
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]
    }

    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(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')

```

Creazione di una cadenza di aggiornamento trimestrale di EPM Cloud con cicli di test di sei settimane

Utilizzare lo script di questa sezione per creare una soluzione self-service per ignorare gli aggiornamenti in modo che gli ambienti Oracle Enterprise Performance Management Cloud vengano aggiornati con frequenza trimestrale con un ciclo di test di sei settimane. In questo caso gli ambienti di produzione vengono aggiornati sei settimane dopo gli ambienti di test.

Per impostazione predefinita, EPM Cloud applica un aggiornamento mensile agli ambienti. Utilizzare il comando [skipUpdate](#) per ignorare l'applicazione degli aggiornamenti mensili a un ambiente o per visualizzare le richieste correnti di differimento degli aggiornamenti. È possibile automatizzare l'esecuzione manuale dei comandi `skipUpdate` utilizzando gli script inclusi in questa sezione. Questi script rendono automatico il processo di differimento degli aggiornamenti in modo che vengano applicati ogni trimestre con un ciclo di test di sei settimane.

 **Note:**

1. Non è possibile ignorare gli aggiornamenti per più di tre mesi consecutivi. Lo script genera ad esempio un errore se si tenta di aggiornare un ambiente EPM Cloud solo a febbraio e ottobre.
2. Tutti gli aggiornamenti che si sono resi disponibili nel periodo intermedio vengono applicati all'ambiente con l'aggiornamento successivo. Si supponga ad esempio di utilizzare questo script per programmare aggiornamenti trimestrali da applicare solo a febbraio, maggio, agosto e novembre. In questo caso, l'aggiornamento di maggio ad esempio applicherà all'ambiente tutti gli aggiornamenti mensili e le relative patch applicabili di EPM Cloud rilasciati dopo l'aggiornamento di febbraio. È possibile che il processo di manutenzione richieda più tempo del solito quando viene applicato l'aggiornamento.
3. Questo script imposta la cadenza di aggiornamento solo per un trimestre.

Esempio di scenario: il ciclo di aggiornamento dell'ambiente di test viene fissato il primo venerdì di febbraio (aggiornamento 24.02), maggio (aggiornamento 24.05), agosto (aggiornamento 24.08) e novembre (aggiornamento 24.11). L'ambiente di produzione verrà aggiornato il terzo venerdì di marzo (aggiornamento 24.02) con la versione utilizzata per aggiornare l'ambiente il primo venerdì di febbraio (aggiornamento 24.02). Un aggiornamento analogo all'ambiente di produzione verrà eseguito la terza settimana di giugno (aggiornamento 24.05), settembre (aggiornamento 24.08) e dicembre (aggiornamento 24.11). In questo scenario l'ambiente di produzione non viene aggiornato alla versione corrente ma a quella correntemente in uso nell'ambiente di test.

Script di esempio per Windows

Creare `skip_update.ps1` copiando lo script indicato di seguito. Archivarlo in una directory locale. Per informazioni sull'esecuzione dello script indicato di seguito, fare riferimento alla sezione [Esecuzione dello script](#).

```
# 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]
        $thisupdatemonthhint=[int]$thisupdatemonth
        $nextupdatemonth=$updateversionsarr[$nextint]
        $nextupdatemonthhint=[int]$nextupdatemonth

        if (${nextupdatemonth} -eq "") {
            $nextupdatemonth=$updateversionsarr[0]
            $nextupdatemonthhint=[int]$nextupdatemonth
        }

        compareUpdateMonths $thisupdatemonthhint $nextupdatemonthhint
    }
}

validateUpdateVersions
populateFromCurrentArrays
processSkipUpdates

```

Script di esempio di Linux/UNIX

Creare `skip_update.sh` copiando lo script indicato di seguito. Archivarlo in una directory locale. Per informazioni sull'esecuzione dello script indicato di seguito, fare riferimento alla sezione [Esecuzione dello script](#).

```
#!/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
        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
        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}
        fi
    }
  
```

```

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

```

Script Groovy lato server

Creare lo script Groovy `skip_update.groovy` copiando lo script seguente e poi aggiornandolo. Per informazioni sull'esecuzione dello script indicato di seguito, fare riferimento alla sezione [Esecuzione dello script](#) .
 Aggiornare le variabili seguenti nello script Groovy.

- `username` Il nome utente di un Amministratore servizi nell'ambiente in cui impostare la cadenza di aggiornamento non mensile.
- `password` La password dell'Amministratore servizi oppure il nome e la posizione del file delle password criptate.
- `url` L'URL dell'ambiente in cui si desidera impostare la cadenza di aggiornamento non mensile.
- `updateversions` Un elenco degli aggiornamenti di EPM Cloud da applicare all'ambiente identificato dal parametro `url`, Ad esempio, `updateversions=02,05,08,11`.

È necessario usare due cifre per specificare le versioni. Anteporre uno zero per gli aggiornamenti da 01 a 09. Lo script tenta di eseguire il comando `skipUpdate` per gli aggiornamenti non inclusi nel valore del parametro `updateversions`. Se ad esempio si specifica `updateversions=02,05,08,11`, lo script tenta di impostare i flag per ignorare gli aggiornamenti per 01 (gennaio), 03 (marzo), 04 (aprile), 06 (giugno), 07 (luglio), 09 (settembre), 10 (ottobre) e 12 (dicembre). In questo caso, vengono applicati all'ambiente gli aggiornamenti 02 (febbraio), 05 (maggio), 08 (agosto) e 11 (novembre) di EPM Cloud.

- `podtype` Il tipo di ambiente EPM Cloud. I valori validi sono `test` e `prod`.
- `proxyserverusername` Il nome utente per autenticare una sessione protetta con il server proxy che controlla l'accesso a internet.
- `proxyserverpassword` La password per autenticare l'utente con il server proxy.
- `proxyserverdomain` Il nome del dominio definito per il server proxy.



Note:

Se non si utilizza un server proxy, non specificare alcun valore per i parametri `proxyserverusername`, `proxyserverpassword` e `proxyserverdomain`.

```
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,"ProxyServerUserName="
    + 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')

```

Creazione del file `input.properties` per l'esecuzione degli script `skip_update` in Windows e Linux/UNIX

Per eseguire `skip_update.ps1` o `skip_update.sh`, creare il file `input.properties` e aggiornarlo con le informazioni relative all'ambiente. Salvare il file in una directory locale. Il contenuto del file varia a seconda del sistema operativo.

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 Parametri di input.properties

Parametro	Descrizione
javahome	Posizione di JAVA_HOME. Solo per Linux/UNIX.
epmautomatescript	Percorso assoluto dell'eseguibile di EPM Automate (epmautomate.sh). Solo per Linux/UNIX.
username	Nome utente di un amministratore del servizio.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
url	URL dell'ambiente in cui si desidera impostare la cadenza di aggiornamento non mensile.
updateversions	updateversions Un elenco degli aggiornamenti di EPM Cloud da applicare all'ambiente identificato dal parametro url. Ad esempio, updateversions=02,05,08,11. È necessario usare due cifre per specificare le versioni. Anteporre uno zero per gli aggiornamenti da 01 a 09. Lo script tenta di eseguire il comando skipUpdate per gli aggiornamenti non inclusi nel valore del parametro updateversions. Se ad esempio si specifica updateversions=02,05,08,11, lo script tenta di impostare i flag per ignorare gli aggiornamenti per 01 (gennaio), 03 (marzo), 04 (aprile), 06 (giugno), 07 (luglio), 09 (settembre), 10 (ottobre) e 12 (dicembre). In questo caso, vengono applicati all'ambiente gli aggiornamenti 02 (febbraio), 05 (maggio), 08 (agosto) e 11 (novembre) di EPM Cloud.
podtype	Tipo di ambiente EPM Cloud. I valori validi sono test e prod.

Esecuzione dello script

1. Solo per Windows e Linux/UNIX

- Creare skip_update.ps1 o skip_update.sh copiando lo script da una sezione precedente.
- Creare il file input.properties e salvarlo nella directory in cui è memorizzato lo script skip_update. Il contenuto di questo file varia a seconda del sistema operativo. Fare riferimento alla sezione [Creazione del file input.properties per l'esecuzione degli script skip_update in Windows e Linux/UNIX](#). Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
- Avviare lo script.
 - **Windows PowerShell:** eseguire skip_update.ps1.
 - **Linux/UNIX:** eseguire ./skip_update.sh.

2. Groovy lato server:

- Creare lo script Groovy skip_update.groovy e aggiornarlo quando necessario.

- Utilizzare la schermata Groovy in un processo aziendale EPM Cloud oppure rendere automatica l'esecuzione dello script come indicato nella sezione [runBusinessRule](#). Per informazioni sull'esecuzione dello script Groovy mediante EPM Automate, fare riferimento alla sezione [Running Commands without Installing EPM Automate \(in lingua inglese\)](#).

Scenari campione per Planning, Consolidation, Tax Reporting ed Enterprise Profitability and Cost Management

Gli script disponibili in questa sezione consentono di automatizzare i task negli ambienti Planning (incluso Planning Modules), Financial Consolidation and Close, Tax Reporting ed Enterprise Profitability and Cost Management.

Vedere anche:

- [Automazione dell'esportazione di un numero elevato di celle da un cubo di memorizzazione di aggregazione](#)
Utilizzare lo script di PowerShell o Bash descritto in questa sezione per esportare un numero elevato di celle da un cubo ASO (Aggregate Storage, memorizzazione di aggregazione).
- [Importazione di metadati in un'applicazione](#)
Utilizzare questi script per importare manualmente i metadati dell'applicazione da un file.
- [Importazione di dati, esecuzione di uno script di calcolo e copia dei dati da un database di memorizzazione a blocchi in un database di memorizzazione di aggregazione](#)
Utilizzare questi script per importare i dati da un file, aggiornare il cubo, eseguire una regola business per calcolare il cubo, quindi eseguire il push dei dati in un cubo di memorizzazione di aggregazione (ASO).
- [Esportazione e download di metadati e dati](#)
Utilizzare questi script per esportare i metadati e i dati dell'applicazione e poi scaricare i file di esportazione in una directory locale.
- [Esportazione e download dei dati dell'applicazione](#)
Utilizzare questi script per esportare i dati dell'applicazione e scaricarli in una directory locale.
- [Automatizzazione dell'archiviazione dei record di audit dell'applicazione](#)
Utilizzare gli script per Windows e per Linux descritti in questa sezione per automatizzare il processo di esportazione e archiviazione dei dati di audit dell'applicazione in un computer locale.
- [Caricamento di un file di dati in un ambiente ed esecuzione di una regola di caricamento dati](#)
Utilizzare questi script per caricare un file in un ambiente e quindi eseguire una regola dati per importare i dati dal file in un'applicazione.
- [Automazione dell'integrazione dei dati giornalieri](#)
In questo scenario si esamina l'uso di uno script di esempio per automatizzare l'integrazione dei dati su base regolare.

Automazione dell'esportazione di un numero elevato di celle da un cubo di memorizzazione di aggregazione

Utilizzare lo script di PowerShell o Bash descritto in questa sezione per esportare un numero elevato di celle da un cubo ASO (Aggregate Storage, memorizzazione di aggregazione).

A causa dei limiti imposti da `QUERYRESULTLIMIT` di Oracle Essbase, è impossibile esportare un'elevata quantità di dati dall'interfaccia utente. Lo script di PowerShell descritto in questa sezione suddivide l'operazione di esportazione in un numero specificato di job, esegue ogni job, scarica i dati esportati e concatena i file di esportazione in un unico file di esportazione verificando che sia presente solo un'intestazione.



Nota:

Questi script eseguono un job esistente di tipo esportazione dati. Per istruzioni dettagliate sulla creazione di job, fare riferimento alla sezione *Gestione dei job* nella Guida *Amministrazione di Planning*.

Script PowerShell

```
$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 " $stopLevelMemberForExport " in
hierarchy"
$member = findMember $response $stopLevelMemberForExport
if ( $member.name -ne $stopLevelMemberForExport ) {
    Write-Host $stopLevelMemberForExport " not found in hierarchy,
exiting ..."
    return 128
}
Write-Host "Found member " $stopLevelMemberForExport " in hierarchy"

#retrieve level 0 memebers 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 + $memberArray.Length
    }
    for ($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"

```

Script Bash

```

#!/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 <<< "${memberList}"
    rowMembers=$(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"

```

Per esportare un numero elevato di celle da un cubo ASO (Aggregate Storage, memorizzazione di aggregazione), procedere come segue.

1. Copiare lo script PowerShell o Bash e salvarlo nel file system, ad esempio come ASOCellExport.ps1 o ASOCellExport.sh.
2. Modificare il file di script e impostare i valori dei parametri. Per informazioni dettagliate, fare riferimento alla tabella riportata di seguito.

Tabella 3-15 Valori di variabili da includere negli script PowerShell e Bash

Variabile	Descrizione
user	<p>Dominio e nome utente di un Amministratore servizi nel formato <i>DOMAIN.USER</i>.</p> <p>Esempi:</p> <p>Windows: \$user = 'exampleDomain.jDoe'</p> <p>Linux/UNIX: user = 'exampleDomain.jDoe'</p>

Tabella 3-15 (Cont.) Valori di variabili da includere negli script PowerShell e Bash

Variabile	Descrizione
pass	<p>Password dell'utente con ruolo Amministratore servizi o posizione in cui è disponibile il password file cifrato. Fare riferimento al comando encrypt per informazioni sulla creazione di un password file cifrato.</p> <p>Esempi:</p> <p>Windows: \$pass = 'Example'</p> <p>Linux/UNIX: pass = 'Example'</p>
serverURL	<p>URL dell'ambiente Oracle Enterprise Performance Management Cloud.</p> <p>Esempi:</p> <p>Windows: \$serverURL = 'https://example.oraclecloud.com'</p> <p>Linux/UNIX: serverURL = 'https://example.oraclecloud.com'</p>
applicationName	<p>Nome di un'applicazione Planning, Financial Consolidation and Close, Tax Reporting o Enterprise Profitability and Cost Management.</p> <p>Esempi:</p> <p>Windows: \$applicationName = 'Vision'</p> <p>Linux/UNIX: applicationName = 'Vision'</p>
cubeName	<p>Nome di un cubo nell'applicazione.</p> <p>Esempi:</p> <p>Windows: \$cubeName = 'VisASO'</p> <p>Linux/UNIX: cubeName = 'VisASO'</p>
splitDimension	<p>Nome di una dimensione i cui membri vengono utilizzati per suddividere l'esportazione in gruppi.</p> <p>Esempi:</p> <p>Windows: \$splitDimension = 'Account'</p> <p>Linux/UNIX: splitDimension = 'Account'</p>
topLevelMemberForExport	<p>Nome di un membro della gerarchia secondaria della dimensione in cui viene creato un elenco di membri di livello 0.</p> <p>Esempi:</p> <p>Windows: \$topLevelMemberForExport = 'Total Cash Flow'</p> <p>Linux/UNIX: topLevelMemberForExport = 'Total Cash Flow'</p>
exportJobName	<p>Nome di un job esistente di tipo esportazione dati. Le impostazioni specificate in questo job verranno sovrascritte dai parametri impostati nello script.</p> <p>Esempi:</p> <p>Windows: \$exportJobName = 'ASO Cell Export'</p> <p>Linux/UNIX: exportJobName = 'ASO Cell Export'</p>

Tabella 3-15 (Cont.) Valori di variabili da includere negli script PowerShell e Bash

Variabile	Descrizione
exportFilePrefix	<p>Prefisso di nome file per identificare in modo univoco i file generati dal job di esportazione.</p> <p>Esempi:</p> <p>Windows: \$exportFilePrefix = 'cashflow'</p> <p>Linux/UNIX: exportFilePrefix = 'cashflow'</p>
columnMembers	<p>Colonne di membro da includere nell'esportazione.</p> <p>Esempi:</p> <p>Windows: \$columnMembers = 'Period'</p> <p>Linux/UNIX: columnMembers = 'Period'</p>
povMembers	<p>Punti di vista da includere nell'esportazione. I membri POV devono includere tutte le altre dimensioni e possono includere funzioni come mostrato di seguito.</p> <p>ILvl0Descendants (YearTotal), ILvl0Descendants (Year), ILvl0Descendants (Scenario), ILvl0Descendants (Version), ILvl0Descendants (P_TP), ILvl0Descendants (AltYear)</p> <p>Esempi:</p> <p>Windows: \$povMembers = 'YTD'</p> <p>Linux/UNIX: povMembers = 'YTD'</p>
numberOfExportFiles	<p>Numero di job da eseguire per l'operazione di esportazione. Se l'esportazione continua ad avere esito negativo a causa dei limiti di query, aumentare questo numero.</p> <p>Esempi:</p> <p>Windows: \$numberOfExportFiles = 3</p> <p>Linux/UNIX: numberOfExportFiles = 3</p>

- Utilizzando la utility di pianificazione di Windows o un job cron, programmare l'esecuzione dello script per un'ora appropriata. Vedere [Automazione dell'esecuzione degli script](#) per informazioni sulla procedura dettagliata.

Importazione di metadati in un'applicazione

Utilizzare questi script per importare manualmente i metadati dell'applicazione da un file.

Questi script consentono di eseguire le attività indicate di seguito.

- Collegamento a un ambiente.
- Caricamento di un file di metadati.
- Importazione di metadati dal file caricato nell'applicazione mediante un job.
- Aggiornamento del cubo.
- Disconnessione.

Script di esempio per Windows

Creare `importMetadata.ps1` copiando lo script seguente. Archivarlo in una directory locale.

```
$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
```

Script di esempio di Linux/UNIX

Creare `importMetadata.sh` copiando lo script seguente. Archivarlo in una directory locale.

```
#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "${serviceURL}"
${epmautomatescript} uploadfile "${file1}"
${epmautomatescript} importmetadata "${jobName}" "${file1}"
${epmautomatescript} refreshcube
${epmautomatescript} logout
```

Creazione del file `input.properties`

Creare il file `input.properties` copiando una delle opzioni indicate di seguito e aggiornandolo con le informazioni relative all'ambiente in uso. Salvare il file nella directory in cui è archiviato `importMetadata.ps1` o `importMetadata.sh`.

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
```

Tabella 3-16 Parametri di input.properties

Parametro	Descrizione
javahome	Posizione di JAVA_HOME. Solo per Linux/UNIX.
epmautomatescript	Percorso assoluto dell'eseguibile di EPM Automate (epmautomate.sh). Solo per Linux/UNIX.
username	Nome utente di un Amministratore servizi a cui è assegnato anche il ruolo Amministratore del dominio di Identity.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
serviceURL	URL dell'ambiente dal quale generare lo snapshot.
File1	Nome del file ZIP contenente i metadati da importare.
JobName	Job da usare per l'importazione dei metadati.

Esecuzione degli script

1. Creare `importMetadata.ps1` o `importMetadata.sh` copiando lo script da una sezione precedente.
2. Creare il file `input.properties` e salvarlo nella directory in cui è memorizzato lo script `importMetadata`. Il contenuto di questo file varia a seconda del sistema operativo. Fare riferimento alla sezione [Creazione del file input.properties](#). Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
3. Avviare lo script.
 - **Windows PowerShell:** eseguire `importMetadata.ps1`.
 - **Linux/UNIX:** eseguire `./importMetadata.sh`.

Importazione di dati, esecuzione di uno script di calcolo e copia dei dati da un database di memorizzazione a blocchi in un database di memorizzazione di aggregazione

Utilizzare questi script per importare i dati da un file, aggiornare il cubo, eseguire una regola business per calcolare il cubo, quindi eseguire il push dei dati in un cubo di memorizzazione di aggregazione (ASO).

Questi script consentono di eseguire le azioni indicate di seguito.

- Collegamento a un ambiente.
- Caricamento di un file `data.csv`.
- Importazione di dati da `data.csv` nell'applicazione mediante il job `loadingqldata`.
- Aggiornamento del cubo.

- Esecuzione di regole business per la trasformazione dei dati.
- Push dei dati in un database di memorizzazione di aggregazione mediante un job.
- Disconnessione.

Script di esempio per Windows

Creare `importDataPlus.ps1` copiando lo script riportato di seguito. Salvarlo in una directory locale.

```
$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
```

Script di esempio di Linux/UNIX

Creare `importDataPlus.ps1` copiando lo script riportato di seguito. Salvarlo in una directory locale.

```
#!/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
```

Creazione del file input.properties

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
```

Tabella 3-17 Parametri di input.properties

Parametro	Descrizione
javahome	Posizione di JAVA_HOME. Solo per Linux/UNIX.
epmautomatescript	Percorso assoluto dell'eseguibile di EPM Automate (epmautomate.sh). Solo per Linux/UNIX.
username	Nome utente di un Amministratore servizi a cui è assegnato anche il ruolo Amministratore del dominio di Identity.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
serviceURL	URL dell'ambiente dal quale generare lo snapshot.
File1	Il file di importazione dal quale caricare i dati nell'applicazione.
iimportDataJobName	Il nome del job da usare per importare i dati.
businessRuleName	La regola business da eseguire sui dati importati.

Tabella 3-17 (Cont.) Parametri di input.properties

Parametro	Descrizione
planTypeMapName	La mappa del tipo di piano da usare per copiare dati da un database BSO in un database ASO oppure in un altro database BSO.
param1Key	Prompt runtime 1 per l'esecuzione della regola business.
param1Value	Valore del prompt runtime 1.
param2Key	Prompt runtime 2 per l'esecuzione della regola business.
param2Value	Valore del prompt runtime 2.
clearData	Indica se è necessario eliminare i dati nel database di ricezione. Specificare <code>false</code> per mantenere i dati.

Esecuzione degli script

1. Creare `importDataPlus.ps1` o `importDataPlus.sh` copiando lo script da una sezione precedente.
2. Creare il file `input.properties` e salvarlo nella directory in cui è memorizzato lo script `importDataPlus`. Il contenuto di questo file varia a seconda del sistema operativo. Fare riferimento alla sezione [Creazione del file input.properties](#). Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
3. Avviare lo script.
 - **Windows PowerShell:** eseguire `importDataPlus.ps1`.
 - **Linux/UNIX:** eseguire `./importDataPlus.sh`.

Esportazione e download di metadati e dati

Utilizzare questi script per esportare i metadati e i dati dell'applicazione e poi scaricare i file di esportazione in una directory locale.

Questi script consentono di completare le attività indicate di seguito.

- Collegamento a un ambiente.
- Esportazione di metadati in un file ZIP mediante un job specificato.
- Esportazione dei dati dell'applicazione in un file ZIP mediante un job specificato.
- Elenco dei contenuti della carella inbox/outbox.
- Download dei file di dati esportati nel computer locale.
- Scollegarsi.

Script di esempio per Windows

Creare `exportDownloadMetadataAndData.ps1` copiando lo script seguente. Archivarlo in una directory locale.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./
input.properties -raw)
$username="$($inputproperties.username) "
$passwordfile="$($inputproperties.passwordfile) "
$serviceURL="$($inputproperties.serviceURL) "
$exportFile1="$($inputproperties.exportFile1) "
$exportFile2="$($inputproperties.exportFile2) "
$exportMetaJobName="$($inputproperties.exportMetaJobName) "
$exportDataJobName="$($inputproperties.exportDataJobName) "

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate exportmetadata ${exportMetaJobName} ${exportFile1}
epmautomate exportdata ${exportDataJobName} ${exportFile2}
epmautomate listfiles
epmautomate downloadfile ${exportFile1}
epmautomate downloadfile f${exportFile2}
epmautomate logout
```

Script di esempio di Linux/UNIX

Creare `exportDownloadMetadataAndData.sh` copiando lo script seguente. Archivarlo in una directory locale.

```
#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "$
{serviceURL}"
${epmautomatescript} exportmetadata "${exportMetaJobName}" "$
{exportFile1}"
${epmautomatescript} exportdata "${exportDataJobName}" "${exportFile2}"
${epmautomatescript} listfiles
${epmautomatescript} downloadfile "${exportFile1}"
${epmautomatescript} downloadfile "${exportFile2}"
${epmautomatescript} logout
```

Creazione del file delle proprietà

Creare il file `input.properties` copiando una delle opzioni indicate di seguito e aggiornandolo con le informazioni relative all'ambiente in uso. Salvare il file nella directory in cui è archiviato `exportDownloadMetadataAndData.ps1` o `exportDownloadMetadataAndData.sh`.

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
```

Tabella 3-18 Parametri di input.properties

Parametro	Descrizione
javahome	Posizione di <code>JAVA_HOME</code> . Solo per Linux/UNIX.
epmautomatescript	Percorso assoluto dell'eseguibile di EPM Automate (<code>epmautomate.sh</code>). Solo per Linux/UNIX.
username	Nome utente di un Amministratore servizi a cui è assegnato anche il ruolo Amministratore del dominio di Identity.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
serviceURL	URL dell'ambiente dal quale generare lo snapshot.
exportFile1	Il nome del file nel quale esportare i metadati.
exportFile2	Consente di specificare il nome del file nel quale esportare i dati.
exportDataJobName1	Job da usare per l'esportazione dei metadati.
exportDataJobName2	Job da usare per l'esportazione dei dati.

Esecuzione degli script

1. Creare `exportDownloadMetadataAndData.ps1` o `exportDownloadMetadataAndData.sh` copiando lo script da una sezione precedente.
2. Creare il file `input.properties` e salvarlo nella directory in cui è memorizzato lo script `exportDownloadMetadataAndData`. Il contenuto di questo file varia a seconda del sistema operativo. Fare riferimento alla sezione [Creazione del file delle proprietà](#). Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
3. Avviare lo script.
 - **Windows PowerShell:** eseguire `exportDownloadMetadataAndData.ps1`.
 - **Linux/UNIX:** eseguire `./exportDownloadMetadataAndData.sh`.

Esportazione e download dei dati dell'applicazione

Utilizzare questi script per esportare i dati dell'applicazione e scaricarli in una directory locale.

Questi script consentono di eseguire le operazioni indicate di seguito.

- Collegamento a un ambiente.
- Backup di due set di dati utilizzando i job specificati dall'utente.
- Download dei file di dati esportati.
- Disconnessione.

Script di esempio per Windows

Creare `exportDownloadData.ps1` copiando lo script riportato di seguito. Salvarlo in una directory locale.

```
$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
```

Script di esempio di Linux/UNIX

Creare `exportDownloadData.sh` copiando lo script riportato di seguito. Salvarlo in una directory locale.

```
#!/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
    
```

Creazione del file input.properties

Creare il file `input.properties` copiando una delle opzioni indicate di seguito e aggiornandolo con le informazioni relative all'ambiente in uso. Salvare il file nella directory in cui è archiviato `exportDownloadData.ps1` o `exportDownloadData.sh`.

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
    
```

Tabella 3-19 Parametri di input.properties

Parametro	Descrizione
javahome	Posizione di JAVA_HOME. Solo per Linux/UNIX.
epmautomatescript	Percorso assoluto dell'eseguibile di EPM Automate (epmautomate.sh). Solo per Linux/UNIX.
username	Nome utente di un Amministratore servizi a cui è assegnato anche il ruolo Amministratore del dominio di Identity.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
serviceURL	URL dell'ambiente dal quale generare lo snapshot.
exportFile1 e exportFile2	Il nome del file in cui devono essere esportati i dati.
exportDataJobName1 e exportDataJobName2	Job da usare per l'esportazione dei dati.

Esecuzione degli script

1. Creare `exportDownloadData.ps1` o `exportDownloadData.sh` copiando lo script da una sezione precedente.

2. Creare il file `input.properties` e salvarlo nella directory in cui è memorizzato lo script `exportDownloadData`. Il contenuto di questo file varia a seconda del sistema operativo. Fare riferimento alla sezione [Tabella 1](#). Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
3. Avviare lo script.
 - **Windows PowerShell:** eseguire `exportDownloadData.ps1`.
 - **Linux/UNIX:** eseguire `./exportDownloadData.sh`.

Automatizzazione dell'archiviazione dei record di audit dell'applicazione

Utilizzare gli script per Windows e per Linux descritti in questa sezione per automatizzare il processo di esportazione e archiviazione dei dati di audit dell'applicazione in un computer locale.

I dati di audit dell'applicazione vengono mantenuti solo per 365 giorni. Customizzare questi script ed eseguirli una volta ogni 180 giorni oppure nei tempi previsti dai criteri di conservazione dati per evitare di perdere dati di audit cronologici dopo 365 giorni.



Note:

Questi script sono sviluppati per archiviare i dati in aree di memorizzazione locali. È possibile modificarli per archiviare i file di dati di audit esportati in aree di memorizzazione di rete o nel cloud di memorizzazione, ad esempio Oracle Object Storage.

Table 3-20 Parametri e relativi valori

Parametro	Valore
<code>url</code>	URL dell'ambiente. Esempi: <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>
<code>user</code>	Nome utente di un amministratore servizi per l'accesso all'ambiente per lo scaricamento dei dati di audit. Esempi: <ul style="list-style-type: none"> • Windows: <code>set user=ExampleAdmin</code> • Linux: <code>user=ExampleAdmin</code>

Table 3-20 (Cont.) Parametri e relativi valori

Parametro	Valore
password	Password dell'amministratore servizi (scelta non consigliata) o nome e posizione del password file cifrato. Vedere il comando encrypt per informazioni sulla creazione di un password file cifrato. Esempi: <ul style="list-style-type: none"> Windows: set password="C:\mySecuredir\password.epw" Linux: password="/home/user1/mySecuredir/password.epw"
AuditFileName	Nome del file di dati di audit. Per renderlo univoco, lo script aggiunge l'indicatore orario dell'esportazione dei dati di audit al nome del file. Esempio <ul style="list-style-type: none"> Windows: set AuditFileName=AuditData Linux: AuditFileName=AuditData
NumberOfBackups	Numero di file di backup da mantenere nell'area di memorizzazione. Il valore predefinito è 10. Superato questo numero, il backup meno recente viene sostituito. Esempio <ul style="list-style-type: none"> Windows: set NumberOfBackups=20 Linux: NumberOfBackups=20
Solo per script per Linux	
epmautomatescript	Posizione in cui è installato EPM Automate. Esempio: /home/user1/epmautomate/bin/epmautomate.sh
javahome	Posizione di <code>JAVA_HOME</code> . Esempio: /home/user1/jdk1.8.0_191/

Script per Windows

Creare un file batch, ad esempio `AuditExport.bat`, contenente uno script simile a quello riportato di seguito per automatizzare l'esportazione e lo scaricamento dei dati di audit in un computer locale.

```
@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%

```

Script per Linux

Creare uno script shell, ad esempio `AuditExport.sh`, contenente uno script simile a quello riportato di seguito per automatizzare l'esportazione e lo scaricamento dei dati di audit in un computer locale.

```

#!/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 --

```

Caricamento di un file di dati in un ambiente ed esecuzione di una regola di caricamento dati

Utilizzare questi script per caricare un file in un ambiente e quindi eseguire una regola dati per importare i dati dal file in un'applicazione.

Prerequisiti

- In Gestione dati sono disponibili le seguenti definizioni:
 - Una definizione di regola di caricamento dati denominata `VisionActual`. Si presume che la regola dei dati non specifichi un percorso file per il file di input.
 - Definizioni di periodo da `Mar-15` a `Jun-15`
- Un file di dati con formato corretto (`GLActual.dat`) che contenga dati.

Per importare i dati ed eseguire una regola di caricamento dati è necessario eseguire i comandi per completare la seguente procedura.

- Collegamento all'ambiente.
- Caricare un file `GLActual.dat` che contiene dati per i periodi da `Mar-15` a `Jun-15` nella cartella `inbox/Vision` di Gestione dati.
- Importare i dati da `GLActual.dat` in Gestione dati utilizzando la regola di caricamento dati `VisionActual` con periodo di inizio `Mar-15`, periodo di fine `Jun-15` e modalità di importazione `REPLACE`.
- Esportare i dati con l'opzione `STORE_DATA` per unire i dati della tabella intermedia di Data Management con i dati dell'applicazione esistenti.

- Scollegarsi.

Script di esempio per Windows

Creare `runDataLoadRule.ps1` copiando lo script seguente. Archivarlo in una directory locale.

```
$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
```

Script di esempio di Linux/UNIX

Creare `runDataLoadRule.sh` copiando lo script seguente. Archivarlo in una directory locale.

```
#!/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
```

Creazione del file `input.properties`

Creare il file `input.properties` copiando una delle opzioni indicate di seguito e aggiornandolo con le informazioni relative all'ambiente in uso. Salvare il file nella directory in cui è archiviato `runDataLoadRule.ps1` o `importMetadata.sh`.

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

```

Tabella 3-21 Parametri di input.properties

Parametro	Descrizione
javahome	Posizione di JAVA_HOME. Solo per Linux/UNIX.
epmautomatescript	Percorso assoluto dell'eseguibile di EPM Automate (epmautomate.sh). Solo per Linux/UNIX.
username	Nome utente di un Amministratore servizi a cui è assegnato anche il ruolo Amministratore del dominio di Identity.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
serviceURL	URL dell'ambiente dal quale generare lo snapshot.
dataFile	Il file che contiene i dati da importare usando la regola dati.
dataFileUploadLocation	Posizione nella quale deve essere caricato il file di dati.
dataRuleName	Il nome della regola di caricamento dati definito in Integrazione dati.
startPeriod	Il primo periodo per il quale devono essere caricati i dati. È necessario definire questo nome di periodo nel mapping dei periodi di Integrazione dati.
endPeriod	In caso di più periodi di caricamento dati, l'ultimo periodo per il quale caricare i dati. Se il caricamento è per un periodo singolo, usare lo stesso periodo come periodo di inizio. È necessario definire questo nome di periodo nel mapping dei periodi di Integrazione dati.
importMode	Modalità per l'importazione di dati in Integrazione dati. Usare APPEND, REPLACE o RECALCULATE. Usare NONE per ignorare l'importazione di dati nelle tabelle intermedie.

Tabella 3-21 (Cont.) Parametri di input.properties

Parametro	Descrizione
exportMode	<p>Modalità per l'esportazione di dati nell'applicazione. Utilizzare Integrazione dati. Utilizzare <code>STORE_DATA</code>, <code>ADD_DATA</code>, <code>SUBTRACT_DATA</code> o <code>REPLACE_DATA</code>. Utilizzare <code>NONE</code> per ignorare l'esportazione di dati da Integrazione dati nell'applicazione.</p> <div style="border: 1px solid #0070c0; padding: 10px; margin-top: 10px;"> <p> Nota:</p> <p>In Financial Consolidation and Close sono supportate solo le modalità <code>MERGE</code> e <code>NONE</code>.</p> </div>

Esecuzione dello script

1. Creare `runDataLoadRule.ps1` o `runDataLoadRule.sh` copiando lo script da una sezione precedente.
2. Creare il file `input.properties` e salvarlo nella directory in cui è memorizzato lo script `runDataLoadRule`. Il contenuto di questo file varia a seconda del sistema operativo. Fare riferimento alla sezione [Creazione del file input.properties](#). Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
3. Avviare lo script.
 - **Windows PowerShell:** eseguire `runDataLoadRule.ps1`.
 - **Linux/UNIX:** eseguire `./runDataLoadRule.sh`.

Automazione dell'integrazione dei dati giornalieri

In questo scenario si esamina l'uso di uno script di esempio per automatizzare l'integrazione dei dati su base regolare.

Creare un file batch (`.bat`) o shell (`.sh`) contenente uno script simile a quello riportato di seguito per rendere automatiche le attività relative all'integrazione dei dati. Lo script di esempio di Windows riportato di seguito automatizza l'integrazione dei dati dell'applicazione mediante il completamento delle attività indicate.

- Collegarsi a un ambiente.
- Eliminare `DailyPlanData` se presente.
- Caricare `DailyPlanData` nel servizio.
- Eseguire la regola business `Clear Plan Targets` per il tipo di piano `Plan1`.
- Importare i dati utilizzando il nome job `LoadDailyPlan`.
- Eseguire la regola business `Balance Sheet - Plan`.
- Eseguire la regola business `Allocate Plan Targets`.

- Eliminare `DailyTarget.zip` se presente.
- Esportare i dati in `DailyTarget.zip` utilizzando il nome di job `ExportDailyTarget`.
- Eseguire il download di `DailyTarget.zip` sul server e aggiungere l'indicatore orario.
- Scollegarsi dall'ambiente.

 **Nota:**

Se si assegna un nuovo scopo a questo script per un uso personale, assicurarsi di modificare i valori dei parametri `SET url` e `SET user`. Inoltre, è possibile modificare i valori dei parametri `dataimportfilename`, `dataexportfilename`, `importdatajobname`, `exportdatajobname`, `br_clear`, `br_calculatebalancesheet` e `br_allocatetarget` in base alle proprie esigenze.

Vedere [Automazione dell'esecuzione degli script](#) per informazioni su come pianificare lo script mediante Utilità di pianificazione di Windows.

```
@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%

```

Scenari di esempio per Account Reconciliation

Vedere anche:

- [Caricamento di saldi preformattati in un periodo](#)
 Utilizzare questi script per importare i dati mappati da un file caricato in un ambiente Account Reconciliation.
- [Caricamento e importazione di uno snapshot di backup](#)
 Utilizzare questi script per caricare e importare uno snapshot di backup in un ambiente Account Reconciliation.

- [Archiviazione di transazioni con corrispondenza meno recenti e rimozione di transazioni archiviate](#)
 Utilizzare gli script in questa sezione per archiviare le transazioni con corrispondenza risalenti a un numero di giorni uguale o superiore a un valore specificato, inclusi i relativi dettagli di supporto e adeguamento e poi rimuovere le transazioni archiviate da Account Reconciliation. Le transazioni con corrispondenza archiviate vengono registrate in un file ZIP.

Caricamento di saldi preformattati in un periodo

Utilizzare questi script per importare i dati mappati da un file caricato in un ambiente Account Reconciliation.

Script di esempio per Windows

Creare un file denominato `runPreformattedBalances.ps1` copiando lo script seguente. Archivarlo in una directory locale.

```
$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
```

Script di esempio di Linux/UNIX

Creare un file denominato `runPreformattedBalances.sh` copiando lo script seguente. Archivarlo in una directory locale.

```
#!/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
```

Esempio di file input.properties

Per eseguire gli script `runPreformattedBalances`, creare il file `input.properties` e aggiornarlo con le informazioni relative all'ambiente. Salvare il file nella directory in cui è archiviato `runPreformattedBalances.sh` o `runPreformattedBalances.ps1`.

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
```

Tabella 3-22 Parametri di input.properties

Parametro	Descrizione
javahome	Posizione di <code>JAVA_HOME</code> . Solo per Linux/UNIX.
epmautomatescript	Percorso assoluto dell'eseguibile di EPM Automate (<code>epmautomate.sh</code>). Solo per Linux/UNIX.
username	Nome utente di un amministratore del servizio.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
serviceURL	URL dell'ambiente host dell'applicazione in cui caricare i saldi preformattati.
dataFile	Un file CSV contenente i saldi preformattati, in genere creati da un General Ledger, da caricare nell'applicazione. Questo file deve essere stato caricato in precedenza nell'ambiente utilizzando il comando uploadFile .
period	Il periodo di riconciliazione nel quale devono essere caricati i saldi preformattati.
balanceType	Il tipo di saldi preformattati contenuti in <code>dataFile</code> .
currencyBucket	Il gruppo di valute per i saldi preformattati.

Esecuzione dello script

1. Creare `runPreformattedBalances.ps1` o `runPreformattedBalances.sh` copiando lo script da una sezione precedente.
2. **Solo per Windows e Linux/UNIX**
 - Creare il file `input.properties` e salvarlo nella directory in cui è memorizzato lo script `runPreformattedBalances`. Il contenuto di questo file varia a seconda del sistema operativo. Fare riferimento alla sezione [Tabella 1](#). Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
 - Avviare lo script.
 - **Windows PowerShell:** eseguire `runPreformattedBalances.ps1`.
 - **Linux/UNIX:** eseguire `./runPreformattedBalances.sh`.

Caricamento e importazione di uno snapshot di backup

Utilizzare questi script per caricare e importare uno snapshot di backup in un ambiente Account Reconciliation.

Script di esempio per Windows

Creare un file denominato `importBackupSnapshot.ps1` copiando lo script seguente. Archivarlo in una directory locale.

```
$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
```

Script di esempio di Linux/UNIX

Creare un file denominato `importBackupSnapshot.sh` copiando lo script seguente. Archivarlo in una directory locale.

```
#!/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
```

Esempio di file input.properties

Per eseguire gli script `importBackupSnapshot`, creare il file `input.properties` e aggiornarlo con le informazioni relative all'ambiente. Salvare il file nella directory in cui è archiviato `importBackupSnapshot.sh` o `importBackupSnapshot.ps1`.

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
```

Tabella 3-23 Parametri di input.properties

Parametro	Descrizione
javahome	Posizione di <code>JAVA_HOME</code> . Solo per Linux/UNIX.
epmautomatescript	Percorso assoluto dell'eseguibile di EPM Automate (<code>epmautomate.sh</code>). Solo per Linux/UNIX.
username	Nome utente di un amministratore del servizio.
password	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
serviceURL	URL dell'ambiente in cui importare lo snapshot.
snapshotName	Il nome dello snapshot dal quale importare artifact e dati. Lo snapshot deve essere stato caricato in precedenza nell'ambiente mediante il comando uploadFile .
userPassword	La password predefinita da assegnare a tutti i nuovi utenti creati nel dominio di Identity in conseguenza dell'importazione dello snapshot.

Esecuzione dello script

1. Creare `importBackupSnapshot.ps1` o `importBackupSnapshot.sh` copiando lo script da una sezione precedente.

2. Creare il file `input.properties` e salvarlo nella directory in cui è memorizzato lo script `runPreformattedBalances`. Il contenuto di questo file varia a seconda del sistema operativo. Fare riferimento alla sezione [Esempio di file `input.properties`](#). Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
3. Avviare lo script.
 - **Windows PowerShell:** eseguire `importBackupSnapshot.ps1`.
 - **Linux/UNIX:** eseguire `./importBackupSnapshot.sh`.

Archiviazione di transazioni con corrispondenza meno recenti e rimozione di transazioni archiviate

Utilizzare gli script in questa sezione per archiviare le transazioni con corrispondenza risalenti a un numero di giorni uguale o superiore a un valore specificato, inclusi i relativi dettagli di supporto e adeguamento e poi rimuovere le transazioni archiviate da Account Reconciliation. Le transazioni con corrispondenza archiviate vengono registrate in un file ZIP.

Funzionamento dello script

1. Accede all'ambiente utilizzando le informazioni contenute nel file `input.properties`.
2. Esegue il comando `archiveTmTransactions` per creare un archivio. Per i file ZIP e di log risultanti vengono utilizzati i nomi predefiniti
`Archive_Transactions_INTERCO_JOB_ID.zip`
`Archive_Transactions_INTERCO_JOB_ID.log`
`epmautomate archiveTmTransactions INTERCO 365 filterOperator=contains filterValue=14001`
 È possibile modificare i parametri del comando modificando il file `input.properties`.
3. Scarica nel computer locale il file di log e il file ZIP che contiene le transazioni archiviate. Lo script visualizza un messaggio di errore se non vengono rilevate transazioni corrispondenti.
4. Copie del file .ZIP che contiene le transazioni archiviate nell'area di memorizzazione oggetti Oracle.
5. Esegue il comando `purgeArchivedTmTransactions` (con l'ID del job `archiveTmTransactions`) per eliminare le transazioni con corrispondenza archiviate dall'applicazione.

Esecuzione dello script

1. Creare il file `input.properties` e aggiornarlo con le informazioni relative all'ambiente in uso. Salvare il file in una directory locale. Nell'ambito di questa descrizione, la directory viene indicata come `parentsnapshotdirectory`. Il contenuto di questo file varia a seconda del sistema operativo. Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire gli script.
2. Creare lo script `transaction_match.ps1` (Windows PowerShell) o `transaction_match.sh` (Linux/UNIX) e salvarlo nella directory `parentsnapshotdirectory` dove si trova il file `input.properties`.

3. Avviare lo script.

- Linux/UNIX: eseguire `./transaction_match.sh`.
- Windows PowerShell: eseguire `transaction_match.ps1`.

Creazione dello script `input.properties`

Creare `input.properties` copiando e aggiornando lo script seguente.

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
epusername=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
```



Nota:

Solo Windows: rimuovere le proprietà indicate di seguito dal file `input.properties`.

- `javahome=JAVA_HOME`
- `epmautomatescript=EPM_AUTOMATE_LOCATION`

Se per l'ambiente di rete Windows non è abilitato `authentication at proxy server`, rimuovere dal file `input.properties` le proprietà indicate di seguito.

- `proxyserverusername`
- `proxyserverpassword`
- `proxyserverdomain`

Tabella 3-24 Parametri di `input.properties`

Parametro	Descrizione
<code>javahome</code>	La directory in cui è installato il JDK usato da EPM Automate. Eliminare questa voce dalla versione Windows di <code>input.properties</code> . Esempio: <code>javahome=./home/JDK/bin</code>

Tabella 3-24 (Cont.) Parametri di input.properties

Parametro	Descrizione
epmautomatescript	La directory in cui è installato EPM Automate. Eliminare questa voce dalla versione Windows di input.properties. Esempio: epmautomatescript=./home/utils/EPMAutomate/bin
epmusername	Il nome utente di un amministratore servizi, utente avanzato, utente o visualizzatore autorizzato ad archiviare transazioni con corrispondenza. Esempio: epmusername=ServiceAdmin
epmuserpassword	Il file delle password cifrate per l'utente identificato come epmusername. Esempio: epmpassword=myPwd.epw
epmurl	L'URL dell'ambiente in cui archiviare le transazioni con corrispondenza. Esempio: epmurl=https://test-cloudpln.pbcs.us1.oraclecloud.com
objectstorageusername	L'ID di un utente che dispone dei diritti di accesso in scrittura richiesti per lo strumento di memorizzazione oggetti cloud di Oracle. Per gli utenti creati in un provider di identità federata, specificare il nome completamente qualificato dell'utente, ad esempio exampleIdP/jdoe o exampleIdP/john.doe@example.com, dove exampleIdP indica il nome del provider di identità federata. Per altri utenti, specificare l'ID utente. Esempio: epmusername=myIdP/jdoe
objectstoragepassword	La password Swift o il token di autenticazione associato all'utente identificato in objectstorageusername. Questa password non corrisponde a quella utilizzata dall'utente per accedere alla console di memorizzazione degli oggetti. Il token di autenticazione è un token generato da Oracle utilizzato per l'autenticazione con API di terze parti, ad esempio per l'autenticazione con un client Swift. Per istruzioni per la creazione di questo token, fare riferimento alla sezione Come creare un token di autenticazione nella <i>Documentazione dell'infrastruttura Oracle Cloud</i> . Esempio: objectstoragepassword=jDoe_PWD
objectstorageurl	L'URL del gruppo di memorizzazione oggetti cloud di Oracle con allegato un nome oggetto facoltativo. Esempio: objectstorageurl=https://swiftobjectstorage.region_identifier.oraclecloud.com/v1/namespace/MT_Archives/2023_archives
matchtype	L'identificativo (TextID) del tipo di corrispondenza dal quale archiviare le transazioni con corrispondenza. Esempio: matchtype=cashrecon

Tabella 3-24 (Cont.) Parametri di input.properties

Parametro	Descrizione
age	Il numero di giorni trascorsi dalla creazione della corrispondenza della transazione. Una transazione con corrispondenza che risale a un numero di giorni superiore o uguale a questo valore verrà archiviata. Esempio: age=180
filteroperator	Le condizioni di filtro per identificare i conti contenenti transazioni con corrispondenza per l'archiviazione. Deve trattarsi di equals, not_equals, starts_with, ends_with, contains o not_contains. Questo valore viene utilizzato insieme al valore filterValue per identificare i conti a partire dai quali devono essere archiviate le transazioni con corrispondenza. Esempio: filteroperator=not_equals
filtervalues	Uno o più valori di filtro per identificare le transazioni da archiviare. Se per filterOperator è specificato equals o not_equals, è possibile utilizzare un elenco separato da spazi per specificare più valori. Se vengono specificati più valori, vengono selezionate per l'archiviazione le transazioni contenute in conti che soddisfano la combinazione operatore e valore di filtro. Esempio: filterValue=101-120 filterValue=140-202
proxyserverusername	Il nome utente per l'autenticazione di una sessione sicura con il server proxy che controlla l'accesso a Internet. Esempio: proxyserverusername=myProxyserver
proxyserverpassword	La password per autenticare l'utente con il server proxy. Esempio: proxyserverpassword=myProxyserver_pwd
proxyserverdomain	Il nome del dominio definito per il server proxy. Esempio: proxyserverdomain=myProxyDomain

Script per Windows

Creare transaction_match.ps1 copiando lo script seguente. Memorizzarlo nella cartella in cui è inserito input.properties.

```
# 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."

```

Script Linux/UNIX

Creare `transaction_match.sh` copiando lo script seguente. Memorizzarlo nella cartella in cui è inserito `input.properties`.

```

#!/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."

```

Scenari di esempio per Profitability and Cost Management

In questi scenari si esamina l'utilizzo dei comandi EPM Automate per eseguire alcuni task comuni di Profitability and Cost Management.

Vedere anche:

- [Importazione di metadati in un'applicazione](#)
Utilizzare questi script per caricare un file di metadati e importare i metadati della dimensione dal file in un'applicazione Profitability and Cost Management.
- [Importazione dei dati ed esecuzione delle regole del programma](#)
Utilizzare questi script per caricare file di dati e importare i dati dai file caricati in un processo aziendale Profitability and Cost Management.

Importazione di metadati in un'applicazione

Utilizzare questi script per caricare un file di metadati e importare i metadati della dimensione dal file in un'applicazione Profitability and Cost Management.

Questi script consentono di eseguire le operazioni indicate di seguito.

- Collegamento all'ambiente.
- Caricamento di un file di metadati.
- Importazione di metadati dal file caricato in un'applicazione.
- Abilitazione dell'applicazione.
- Disconnessione.

Script per Windows

Creare `importMetadata.ps1` copiando lo script riportato di seguito.

```
$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 loadimdata ${applicationName} dataFileName=${dataFileName}
epmautomate enableapp ${applicationName}
epmautomate logout
```

Script Linux/UNIX

Creare `importMetadata.sh` copiando lo script riportato di seguito.

```
#!/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

```

Creazione del file input.properties

Per eseguire gli script `importMetadata`, creare il file `input.properties` e aggiornarlo con le informazioni relative all'ambiente. Salvare il file nella directory in cui è archiviato

`importMetadata.psl` o `importMetadata.sh`.

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

```

Tabella 3-25 Parametri di input.properties

Parametro	Descrizione
<code>javahome</code>	Posizione di <code>JAVA_HOME</code> . Solo per Linux/UNIX.
<code>epmautomatescript</code>	Percorso assoluto dell'eseguibile di EPM Automate (<code>epmautomate.sh</code>). Solo per Linux/UNIX.
<code>username</code>	Nome utente di un Amministratore servizi a cui è assegnato anche il ruolo Amministratore del dominio di Identity.
<code>password</code>	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
<code>serviceURL</code>	URL dell'ambiente dal quale generare lo snapshot.
<code>applicationName</code>	Nome di Profitability and Cost Management in cui caricare i dati.
<code>dataFileName</code>	Nome del file contenente i metadati da importare.
<code>dataFileNameDestination</code>	Posizione di caricamento del file di metadati.

Esecuzione degli script

1. Creare `importMetadata.ps1` o `importMetadata.sh` copiando lo script da una sezione precedente.
2. Creare il file `input.properties` e salvarlo nella directory in cui è memorizzato lo script `importMetadata`. Il contenuto di questo file varia a seconda del sistema operativo. Fare riferimento alla sezione [Creazione del file `input.properties`](#). Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
3. Avviare lo script.
 - **Windows PowerShell:** eseguire `importMetadata.ps1`.
 - **Linux/UNIX:** eseguire `./importMetadata.sh`.

Importazione dei dati ed esecuzione delle regole del programma

Utilizzare questi script per caricare file di dati e importare i dati dai file caricati in un processo aziendale Profitability and Cost Management.

Questi script completano i passi indicati di seguito.

- Collegamento all'ambiente.
- Caricamento di un file di dati contenente i dati da caricare.
- Caricamento del file di regole contenente le regole dati.
- Caricamento di dati dal file dati nell'applicazione per sovrascrivere i valori esistenti.
- Esecuzione di tutte le regole nel file delle regole.
- Disconnessione.

Script per Windows

Creare un file denominato `importData.ps1` copiando lo script seguente. Archivarlo in una directory locale.

```
$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
```

Script Linux/UNIX

Creare un file denominato `importData.sh` copiando lo script seguente. Archivarlo in una directory locale.

```
#!/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
```

Creazione del file `input.properties`

Per eseguire gli script `importData`, creare il file `input.properties` e aggiornarlo con le informazioni relative all'ambiente. Salvare il file nella directory in cui è archiviato `importData.ps1` o `importData.sh`.

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
```

Tabella 3-26 Parametri di `input.properties`

Parametro	Descrizione
<code>javahome</code>	Posizione di <code>JAVA_HOME</code> . Solo per Linux/UNIX.
<code>epmautomatescript</code>	Percorso assoluto dell'eseguibile di EPM Automate (<code>epmautomate.sh</code>). Solo per Linux/UNIX.
<code>username</code>	Nome utente di un Amministratore servizi a cui è assegnato anche il ruolo Amministratore del dominio di Identity.
<code>password</code>	Password dell'Amministratore servizi o nome e posizione del password file cifrato.
<code>serviceURL</code>	URL dell'ambiente dal quale generare lo snapshot.
<code>applicationName</code>	Nome di Profitability and Cost Management in cui caricare i dati.
<code>dataFileName</code>	Nome del file contenente i dati da importare.
<code>rulesFileName</code>	Nome del file contenente le regole da eseguire nei dati importati.
<code>fileDestination</code>	Posizione nella quale devono essere caricati il file di dati e delle regole.
<code>clearAllDataFlag</code>	Specificare se cancellare i dati esistenti nel cubo dell'applicazione. Impostare <code>false</code> se non si desidera cancellare i dati esistenti.
<code>dataLoadValue</code>	Specifica come gestire i dati esistenti. Specificare <code>ADD_TO_EXISTING</code> per mantenere i dati esistenti nel cubo.

Esecuzione degli script

1. Creare `importData.ps1` o `importData.sh` copiando lo script dalla sezione precedente.
2. Creare il file `input.properties` e salvarlo nella directory in cui è memorizzato lo script `importData`. Il contenuto di questo file varia a seconda del sistema operativo. Fare riferimento alla sezione [Creazione del file `input.properties`](#). Assicurarsi di avere i privilegi di scrittura in questa directory. Per Windows, potrebbe essere necessario avviare PowerShell utilizzando l'opzione **Esegui come amministratore** per poter essere in grado di eseguire lo script.
3. Avviare lo script.
 - **Windows PowerShell:** eseguire `importData.ps1`.
 - **Linux/UNIX:** eseguire `./importData.sh`.

Scenari di esempio per Oracle Enterprise Data Management Cloud

In questi scenari di esempio viene descritto l'uso dei comandi di EPM Automate per sincronizzare le dimensioni delle applicazioni tra Oracle Enterprise Data Management Cloud e Oracle Enterprise Performance Management Cloud.

Vedere anche:

- [Sincronizzazione di dimensioni e mapping di Oracle Enterprise Data Management Cloud mediante applicazioni EPM Cloud](#)
In questo scenario di esempio viene analizzata la sincronizzazione di una dimensione tra un'applicazione Oracle Enterprise Data Management Cloud e un'applicazione Oracle Enterprise Performance Management Cloud.
- [Sincronizzazione delle dimensioni di EPM Cloud mediante applicazioni Oracle Enterprise Data Management Cloud](#)
In questo scenario di esempio viene analizzata la sincronizzazione di una dimensione tra un'applicazione Oracle Enterprise Data Management Cloud e un'applicazione Oracle Enterprise Performance Management Cloud.

Sincronizzazione di dimensioni e mapping di Oracle Enterprise Data Management Cloud mediante applicazioni EPM Cloud

In questo scenario di esempio viene analizzata la sincronizzazione di una dimensione tra un'applicazione Oracle Enterprise Data Management Cloud e un'applicazione Oracle Enterprise Performance Management Cloud.

Utilizzare gli script riportati in questa sezione per completare i task elencati di seguito:

- Esportazione di una dimensione da un'applicazione Oracle Enterprise Data Management Cloud
- Esportazione di mapping da un'applicazione Oracle Enterprise Data Management Cloud
- Copia di file di esportazione in un ambiente EPM Cloud
- Importazione di metadati e mapping di dimensioni nell'applicazione EPM Cloud

Per sincronizzare una dimensione e i mapping tra un'applicazione Oracle Enterprise Data Management Cloud e un'applicazione EPM Cloud, procedere come segue.

1. Creare un file di script. A tale scopo, copiare il seguente script:

```
rem Integration example to sync application dimensions between EDM and
rem EPM Cloud
rem Windows script for demonstration purposes only; do not use in
rem 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 exportdimension %EDMAPP% %EDMDIM% %DIMFILE%
call epmautomate exportdimensionmapping %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. Modificare il file di script e impostare i valori dei parametri richiesti. Per una spiegazione e un esempio di parametri, fare riferimento alla sezione [Parametri per l'esecuzione degli script](#).
3. Eseguire lo script manualmente oppure programmarne l'esecuzione, a seconda dei casi. Fare riferimento alla sezione [Automazione dell'esecuzione degli script](#).

Parametri per l'esecuzione degli script

I file di script descritti in questa sezione richiedono di specificare alcuni dei valori di parametro elencati nella seguente tabella. Non tutti questi parametri vengono utilizzati in tutti gli script.

Tabella 3-27 Valori di parametro per i file di script

Parametro	Descrizione
EDMUSER	ID di accesso dell'utente di un amministratore dei servizi di Oracle Enterprise Data Management Cloud. Esempio: EDMUSER=jdoe@example.com
EDMSVR	URL dell'ambiente Oracle Enterprise Data Management Cloud. Esempio: EDMSVR=https:// example.oraclecloud.com
EDMPWDFILE	Nome e posizione del password file (EPW) cifrato dell'amministratore dei servizi di Oracle Enterprise Data Management Cloud. Esempio: EDMPWDFILE=edm_jdoe.epw
EDMAPP	Nome di una dimensione dell'applicazione Oracle Enterprise Data Management Cloud. Esempio: EDMAPP=USOperations
EDMDIM	Nome della dimensione da esportare o importare. Esempio: EDMDIM=entity

Tabella 3-27 (Cont.) Valori di parametro per i file di script

Parametro	Descrizione
EDMLOC	Nome della posizione da esportare. Esempio: EDMLOC=Loc1
EPMUSER	Nome di accesso di un amministratore dei servizi di EPM Cloud. Esempio: EPMUSER=john.doe@example.com
EPMSVR	URL dell'ambiente EPM Cloud. Esempio: EPMSVR=https://example.oraclecloud.com
EPMIMPJOB	Nome di un job di importazione esistente di tipo importazione di metadati nell'ambiente EPM Cloud. Esempio: EPMIMPJOB=imp_DIMMetadata
EPMEXPJOB	Nome di un job esistente di tipo esportazione di metadati nell'ambiente EPM Cloud. Esempio: EPMEXPJOB=Exp_DIMMetadata
PWDFILE	Nome e posizione del password file (EPW) cifrato dell'amministratore dei servizi di EPM Cloud. Fare riferimento al comando encrypt . Esempio: PWDFILE=pwd_jdoe.epw
DIMFILE	Nome del file che conterrà i dati delle dimensioni esportate. Esempio: DIMFILE=entity_file.CSV
MAPFILE	Nome del file che conterrà i dati dei mapping esportati. Esempio: MAPFILE=map_file.CSV

Sincronizzazione delle dimensioni di EPM Cloud mediante applicazioni Oracle Enterprise Data Management Cloud

In questo scenario di esempio viene analizzata la sincronizzazione di una dimensione tra un'applicazione Oracle Enterprise Data Management Cloud e un'applicazione Oracle Enterprise Performance Management Cloud.

Utilizzare gli script riportati in questa sezione per completare i task elencati di seguito:

- Esportazione di metadati (dimensioni) da un'applicazione EPM Cloud
- Copia di file di esportazione contenenti i dati delle dimensioni in un ambiente Oracle Enterprise Data Management Cloud
- Importazione dei metadati delle dimensioni nell'applicazione Oracle Enterprise Data Management Cloud

Per sincronizzare una dimensione tra un'applicazione EPM Cloud e un'applicazione Oracle Enterprise Data Management Cloud, procedere come segue.

1. Creare un file di script Windows. A tale scopo, copiare il seguente script:

```
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=apppname
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

```

Modificare il file di script e impostare i valori dei parametri richiesti. Per una spiegazione e un esempio di parametri, fare riferimento alla sezione [Parametri per l'esecuzione degli script](#).

2. Eseguire lo script manualmente oppure programmarne l'esecuzione, a seconda dei casi. Fare riferimento alla sezione [Automazione dell'esecuzione degli script](#).

Automazione dell'esecuzione degli script

Un amministratore di servizi programma gli script in Utilità di pianificazione di Windows o utilizza un job cron per automatizzare le attività mediante EPM Automate.

Per programmare l'esecuzione degli script di EPM Automate mediante Utilità di pianificazione di Windows, procedere come segue.

1. Fare clic su **Start, Pannello di controllo** e quindi **Strumenti di amministrazione**.
2. Aprire **Utilità di pianificazione**.
3. Selezionare **Azione**, quindi **Crea attività di base**.
4. Immettere un nome di task e una descrizione facoltativa, quindi fare clic su **Avanti**.
5. In **Attivazione attività** selezionare una programmazione per l'esecuzione dello script, quindi fare clic su **Avanti**.
6. Nella schermata successiva specificare gli altri parametri della programmazione, quindi fare clic su **Avanti**.
7. In Azione assicurarsi che sia selezionato **Avvio programma**.

8. In **Avvio programma** procedere come segue.
 - a. In **Programma o script** sfogliare per selezionare lo script che si desidera programmare.
 - b. In **Aggiungi argomenti (facoltativo)** immettere la password dell'amministratore del servizio identificato dal parametro dello script `SET user,`
 - c. In **Inizio (facoltativo)** immettere la posizione in cui è installato lo strumento EPM Automate. In genere si trova in `C:/Oracle/EPMAutomate/bin.`
 - d. Fare clic su **Avanti**.
9. In **Riepilogo** selezionare **Apri la finestra di dialogo Proprietà quando viene scelto Fine**, quindi fare clic su **Fine**.
10. In **Generale** selezionare le opzioni di sicurezza, quindi fare clic su **OK**.
 - **Esegui indipendentemente dalla connessione dell'utente**
 - **Esegui con i privilegi più elevati**

Monitoraggio delle attività di EPM Automate

Per aiutare a identificare lo stato dell'operazione inizializzata, EPM Automate visualizza i codici di stato nella console di esecuzione.

Fare riferimento alla sezione [Codici di uscita](#).

Utilizzare la console job per controllare i job eseguiti utilizzando EPM Automate. Fare riferimento alla sezione *Gestione dei job in Amministrazione di Planning* per i dettagli.

4

Esecuzione di comandi senza installare EPM Automate

Utilizzando Groovy, è possibile eseguire comandi di selezione direttamente in Oracle Enterprise Performance Management Cloud. Non è necessaria un'installazione di EPM Automate per eseguire comandi sul lato server.



Note:

In questo scenario, lo script Groovy viene scritto per essere eseguito direttamente in EPM Cloud e non in un computer client.

Contenuto del capitolo

- [Ambienti che supportano l'esecuzione di comandi sul lato server](#)
- [Fonti di informazioni](#)
- [Comandi supportati](#)
- [Metodi da utilizzare per eseguire EPM Automate utilizzando Groovy su lato server](#)
- [Clonazione di un ambiente tramite uno script Groovy su lato server](#)
- [Invio del report attività via e-mail tramite uno script Groovy su lato server](#)

Ambienti che supportano l'esecuzione di comandi sul lato server

Il supporto degli script Groovy per l'esecuzione di comandi EPM Automate sul lato server è disponibile solo negli ambienti elencati di seguito.

- Processi aziendali Planning e Planning Modules distribuiti in ambienti EPM Enterprise Cloud Service.
- Enterprise Planning and Budgeting Cloud.
- Planning and Budgeting Cloud con l'opzione Plus One
- FreeForm
- Enterprise Profitability and Cost Management
- Financial Consolidation and Close in ambienti EPM Enterprise Cloud Service.
- Tax Reporting in ambienti EPM Enterprise Cloud Service.
- Strategic Workforce Planning
- Sales Planning

Gli script Groovy che incorporano comandi EPM Automate possono essere scritti ed eseguiti solo negli ambienti Oracle Enterprise Performance Management Cloud precedenti. Gli script scritti in questi ambienti possono tuttavia eseguire comandi EPM Automate in qualsiasi

ambiente EPM Cloud. È ad esempio possibile creare lo script in un ambiente EPM Enterprise Cloud Service di Planning e fare in modo che esegua comandi in un ambiente Narrative Reporting che non supporta gli script Groovy.

Fonti di informazioni

Per informazioni dettagliate, fare riferimento alle fonti in *Progettazione con Calculation Manager per Oracle Enterprise Performance Management Cloud* elencate di seguito.

- Informazioni sulle regole business Groovy
- Creazione di una regola business di Groovy

Comandi supportati

Ad eccezione dei comandi elencati di seguito, tutti i comandi di EPM Automate possono essere eseguiti tramite Groovy.

- [downloadFile](#)
- [upgrade](#)
- [uploadFile](#)

I comandi elencati di seguito non possono essere eseguiti nell'ambiente che esegue lo script Groovy.

- [recreate](#)
- [replay](#)
- [resetService](#)
- [restructureCube](#)

Note:

- Con l'esecuzione del comando [encrypt](#), viene creato sul server il password file cifrato, che verrà rimosso dopo sette giorni se inutilizzato.
- Per il corretto funzionamento del comando [feedback](#), tutti i file e gli screenshot utilizzati come allegato devono essere disponibili nella posizione di caricamento predefinita. Fare riferimento alla sezione [Posizione di caricamento predefinita](#). Questa è la posizione in cui vengono memorizzati i file se non si specifica una posizione nel comando [uploadFile](#).

Metodi da utilizzare per eseguire EPM Automate utilizzando Groovy su lato server

- `getEPMAutomate ()` Questo metodo statico fornisce un'istanza della classe `EpmAutomate`, che può quindi essere utilizzata per richiamare i comandi di EPM Automate.

- `execute ()` Questo metodo della classe `EpmAutomate` viene utilizzato per eseguire un comando di EPM Automate. Passare il nome del comando di EPM Automate come primo parametro e le opzioni del comando come parametri successivi. Questo metodo restituisce un'istanza della classe `EpmAutomateStatus`.
- `getStatus ()` Questo metodo della classe `EPMAutomateStatus` restituisce lo stato dell'esecuzione restituito dal comando. Un valore restituito pari a 0 indica che il comando è riuscito, mentre un valore diverso da zero indica che il comando non è riuscito.
- `getOutput ()` Questo metodo della classe `EPMAutomateStatus` restituisce l'output stringa del comando. È ad esempio possibile utilizzare questo metodo per restituire l'output dei comandi `getApplicationAdminMode` e `getDailyMaintenanceStartTime`. Se lo stato restituzione del comando è diverso da zero, questo metodo restituisce il messaggio di errore.
- `getItemsList ()` Questo metodo della classe `EPMAutomateStatus` restituisce l'output elenco del comando. È ad esempio possibile utilizzare questo metodo per restituire l'output dei comandi `getSubstVar`, `listBackups` e `listFiles`.

Clonazione di un ambiente tramite uno script Groovy su lato server

È possibile includere comandi EPM Automate in script Groovy su lato server per clonare ambienti per finalità di disaster recovery. In questo modo è possibile impostare le azioni di disaster recovery senza footprint in locale.

Se le password contengono caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#). Assicurarsi inoltre di sostituire i valori dei parametri elencati di seguito a seconda degli ambienti.

Table 4-1 Parametri da modificare

Parametro	Descrizione
<code>password</code>	Password dell'amministratore dei servizi che esegue l'operazione di clonazione nell'ambiente di origine.
<code>targetpassword</code>	Password dell'amministratore di servizi che esegue l'operazione di clonazione nell'ambiente target.
<code>username</code>	ID utente dell'amministratore di servizi nell'ambiente di origine.
<code>targetusername</code>	ID utente dell'amministratore di servizi nell'ambiente target. Questo utente deve essere assegnato anche al ruolo Amministratore del dominio di Identity nell'ambiente target.
<code>email_id</code>	Indirizzo e-mail a cui si intende inviare informazioni sul processo di clonazione.

Script per la cifratura della password

```
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())

```

Script per la clonazione dell'ambiente

Questo script utilizza i password file cifrati creati utilizzando lo script precedente.

```

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())
```

Invio del report attività via e-mail tramite uno script Groovy su lato server

Questo script può essere utilizzato per inviare tramite e-mail il report attività a un elenco di destinatari. È quindi possibile programmare l'esecuzione giornaliera dello script in modo da ottenere il report attività giornaliero. Lo script esegue le funzioni elencate di seguito.

- Cifra la password dell'amministratore servizi che esegue lo script Groovy.
- Accede all'ambiente Oracle Enterprise Performance Management Cloud utilizzando la password cifrata.
- Invia tramite e-mail il report attività disponibile nell'ambiente a un elenco di destinatari, in genere amministratori servizi
- Eseguce la disconnessione dall'ambiente.

Se le password contengono caratteri speciali, fare riferimento alla sezione [Gestione dei caratteri speciali](#). Assicurarsi inoltre di sostituire i valori dei parametri elencati di seguito a seconda degli ambienti.

Table 4-2 Parametri da modificare

Parametro	Descrizione
user	ID utente di un amministratore servizi per l'accesso all'ambiente.
password	Password dell'amministratore servizi.
url	URL dell'ambiente EPM Cloud da cui deve essere inviato tramite e-mail il report attività.
emailaddresses	Elenco separato da punti e virgole contenente indirizzi e-mail a cui deve essere inviato il report attività.

Per informazioni dettagliate sull'utilizzo delle regole Groovy, fare riferimento alla sezione *Utilizzo delle regole Groovy nel manuale Amministrazione di Planning*.

```
/*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

Replica di un ambiente di EPM Cloud

Questi passi sono necessari per configurare un ambiente secondario di Oracle Enterprise Performance Management Cloud al fine di garantire la disponibilità del servizio se il centro dati Oracle principale risulta non disponibile a causa di circostanze impreviste.



Nota:

Le procedure presentate in questa appendice non sono valide per Narrative Reporting.

- [Configurazione della replica giornaliera dell'artifact](#)
- [Configurazione di repliche su richiesta](#)
- [Configurazione dell'ambiente secondario](#)

Configurazione della replica giornaliera

Per replicare un ambiente, utilizzare EPM Automate per copiare dall'ambiente principale a quello secondario lo snapshot dell'artifact creato durante la manutenzione giornaliera.

Oracle esegue ogni giorno un task di manutenzione di routine su ogni ambiente. Durante la manutenzione del servizio, Oracle crea uno snapshot di manutenzione mediante la creazione di un backup dei contenuti dell'ambiente (dati e artifact esistenti, comprese le assegnazioni di utenti e ruoli dal dominio di Identity).

Per configurare la replica giornaliera del servizio, procedere nel seguente modo.

1. Creare un file di script contenente i seguenti comandi di EPM Automate. Questo script replica lo snapshot dell'applicazione dall'ambiente principale all'ambiente secondario.



Nota:

Assicurarsi di modificare il nome utente, il password file, i nomi del dominio di identità e gli URL del servizio. Per informazioni sulla creazione di un password file cifrato, fare riferimento al comando [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. Utilizzando uno scheduler, ad esempio Utilità di pianificazione di Windows, programmare l'esecuzione del file di script in modo che venga eseguito due ore dopo l'inizio della finestra di manutenzione.
3. Impostare la stessa ora di inizio della finestra di manutenzione per l'ambiente principale e gli ambienti secondari. Per ulteriori informazioni, fare riferimento alla sezione Impostazione dell'orario di manutenzione del servizio in *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori*.

Configurazione di repliche su richiesta

Per ridurre il valore RPO, è possibile creare snapshot su richiesta dell'ambiente principale, quindi copiarli nell'ambiente secondario.

Ad esempio, è possibile creare e programmare l'esecuzione di uno script di EPM Automate ogni sei ore tra le singole repliche giornaliere per ridurre il valore RPO da 24 a 6 ore.



Nota:

Durante la creazione dello snapshot su richiesta, l'ambiente principale viene impostato sulla modalità di sola lettura per alcuni minuti.

Per configurare la replica su richiesta, procedere nel seguente modo.

1. Creare un file di script contenente i seguenti comandi di EPM Automate. Questo script replica lo snapshot dell'applicazione dall'ambiente principale all'ambiente secondario.



Nota:

Assicurarsi di modificare il nome utente, il password file, i nomi del dominio di identità e gli URL del servizio. Per informazioni sulla creazione di un password file cifrato, fare riferimento al comando [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. Utilizzando uno scheduler, ad esempio Utilità di pianificazione di Windows, programmare l'esecuzione del file di script in modo che venga eseguito quando necessario in modo conforme al valore RPO desiderato.

Configurazione dell'ambiente secondario

Configurare l'ambiente secondario per attivarlo.

Completare questa procedura solo se è necessario attivare l'ambiente secondario quando l'ambiente principale non è disponibile per un periodo prolungato di tempo. Prima di configurare l'ambiente secondario, fare riferimento ai seguenti argomenti in *Guida introduttiva a Oracle Enterprise Performance Management Cloud per gli amministratori*:

- Percorsi di migrazione per snapshot di EPM Cloud precedenti
- Percorsi di migrazione per snapshot di EPM Standard Cloud Service ed EPM Enterprise Cloud Service

Per configurare l'ambiente secondario, procedere nel seguente modo.

1. Avviare una sessione di EPM Automate e completare le seguenti attività.

- Eseguire l'accesso all'ambiente secondario utilizzando un account che disponga di entrambi i ruoli Amministratore servizi e Amministratore del dominio di Identity. Assicurarsi di specificare il nome utente, la password, il nome di dominio e a l'URL del servizio.
- Ricreare l'ambiente secondario.
 - Se l'ambiente principale è un ambiente Planning, Tax Reporting, Financial Consolidation and Close o Enterprise Profitability and Cost Management, utilizzare:

```
epmautomate recreate -f
```
 - Se l'ambiente principale non è un ambiente Planning, Tax Reporting, Financial Consolidation and Close o Enterprise Profitability and Cost Management, utilizzare:

```
epmautomate recreate -f TempServiceType=PRIMARY_APPLICATION_TYPE dove  
PRIMARY_APPLICATION_TYPE is ARCS, EDMCS, PCMCS oppure EPRCS.
```
- Importare gli artifact dell'applicazione e del dominio di Identity dallo snapshot.
- Scollegarsi.

Per completare le attività precedenti, eseguire i comandi seguenti. Fare riferimento agli argomenti sui seguenti comandi:

- [login](#)
- [recreate](#)
- [importSnapshot](#)

```
epmautomate login serviceAdmin secondaryPassword.epw secondary_URL  
epmautomate recreate -f
```

```
epmautomate importsnapshot "Artifact Snapshot" importUsers=true  
epmautomate logout
```

2. Eseguire l'accesso all'ambiente secondario del servizio e verificare che tutti i dati siano disponibili.
3. Inviare l'URL dell'ambiente secondario a tutti gli utenti.

A

Preparazione all'esecuzione del comando `simulateConcurrentUsage`

Per simulare il caricamento nell'ambiente, il comando [simulateConcurrentUsage](#) supporta le operazioni seguenti:

- Apertura di form
- Salvataggio di form
- Esecuzione di regole business
- Esecuzione di regole dati
- Apertura di griglie ad hoc
- Esecuzione dei registri dei report di gestione
- Esecuzione dei report di gestione

Procedura

1. Creare il file `requirement.csv`. Fare riferimento alla sezione [Creazione del file requirement.csv](#).
2. Creare i file di input specificando i dettagli delle operazioni incluse in `requirement.csv`. Fare riferimento alle sezioni:
 - [File di input per l'apertura dei form](#)
 - [File di input per il salvataggio dei form](#)
 - [Esecuzione del file di input di una regola business](#)
 - [Esecuzione del file di input della regola dati](#)
 - [File di input della griglia ad hoc](#)
 - [Esecuzione del file di input del registro](#)
 - [Esecuzione del file di input del report](#)
3. Creare il file `UserVarMemberMapping.csv` contenente i dettagli delle variabili utente. Fare riferimento alla sezione [Creazione del file UserVarMemberMapping.csv](#).
4. Creare un file ZIP contenente i file creati nei passi precedenti e caricarlo nell'ambiente. Fare riferimento alla sezione [Creazione e caricamento del file ZIP di input nell'ambiente](#).
5. Eseguire il comando [simulateConcurrentUsage](#) utilizzando il file ZIP caricato.

Creazione del file `requirement.csv`

Iniziare creando il file `requirement.csv` che elenca i dettagli dei casi d'uso per cui eseguire il test. Ogni riga del file CSV identifica il tipo di operazione da eseguire, il nome dell'artifact, il numero di utenti concorrenti, il file di input che specifica i dettagli dell'operazione ed eventuali informazioni aggiuntive correlate all'operazione. Ad esempio, per aprire due form, salvarli ed

eseguire due regole business, specificare 6 righe nel file CSV di input. La prima riga di requirement.csv deve contenere le informazioni seguenti:

```
#Type of Operation,Artifact Name,Number of Users,Input File,Additional Info
```

Ogni riga successiva del file contiene una singola operazione con i relativi parametri. Alcune operazioni potrebbero non richiedere tutti i valori di parametro indicati. La tabella seguente fornisce spiegazioni per le voci previste del file.



Note:

Tutti i valori sono obbligatori salvo quando diversamente indicato nella tabella.

Table A-1 Formato di requirement.csv

Campo	Descrizione
Type of operation	Deve essere uno dei seguenti: <ul style="list-style-type: none"> • Open Form • Save Form • Run Business Rule • Run Data Rule • Ad Hoc Grid • Execute Report • Execute Book
Artifact Name	Questo valore dipende dal tipo di operazione. <ul style="list-style-type: none"> • Open Form: il nome e la posizione del form da aprire. • Save Form: il nome e la posizione del form da salvare. • Run Business Rule: il nome della regola business. • Data Rule: il nome della regola dati. • Ad Hoc Grid: non applicabile (lasciare vuoto). • Execute Report: il nome e la posizione del report. • Execute Book: il nome e la posizione del registro.
Number of Users	Il numero di utenti necessari a simulare l'utilizzo concorrente.
Input File	Il nome del file CSV contenente i valori del POV, i prompt runtime o altri valori specifici del caso da usare.
Additional Info	Parametri aggiuntivi richiesti per l'operazione. Applicabile solo alle griglie ad hoc. Lasciare vuoto per altri casi d'uso.

Note: i nomi degli artifact devono corrispondere a quelli nell'applicazione e devono essere nello stesso caso.

Esempio di file requirement.csv:

```
# 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,
```

Creazione dei file di input

A ogni caso d'uso identificato in `requirement.csv` deve corrispondere un file di input che fornisce tutti i parametri richiesti per l'esecuzione.

È necessario che il file di input contenga una voce per il numero di utenti specificato per il caso d'uso in `requirement.csv`.

Se il numero di voci nel file di input è minore del numero di utenti concorrenti per tale caso d'uso in `requirement.csv`, EPM Automate ripete alcune voci del file di input per eseguire i casi d'uso fino a quando l'operazione non viene eseguita per il numero di utenti identificati in `requirement.csv`.

Ad esempio, se la voce del caso d'uso in `requirement.csv` per l'operazione Esegui regola business è la seguente:

```
Run Business Rule, Copy Budget,10,br_input_file.csv,
```

anche `br_input_file.csv` dovrebbe contenere dieci voci. Se `br_input_file.csv` ne contiene solo sei, EPM Automate le usa per i primi sei utenti. Per i quattro utenti successivi, EPM Automate riutilizza le prime quattro voci in `br_input_file.csv`.

Se il numero di voci nel file di input è maggiore del numero di utenti specificato per il caso d'uso, EPM Automate ignora le ultime voci extra del file di input.

- [File di input per l'apertura dei form](#)
- [File di input per il salvataggio dei form](#)
- [Esecuzione del file di input di una regola business](#)
- [Esecuzione del file di input della regola dati](#)
- [File di input della griglia ad hoc](#)
- [Esecuzione del file di input del report](#)
- [Esecuzione del file di input del registro](#)

File di input per l'apertura dei form

In questo file, utilizzato come riferimento in `requirement.csv` per supportare l'apertura dei form, sono incluse le voci POV nel formato: `pov=[DIM 1:MEMBER 1],[DIM 2:MEMBER 2],[DIM 3:MEMBER 3]`, e così via.

In questa discussione, DIM 1, DIM 2 e così via sono i nomi delle dimensioni mentre MEMBER 1, MEMBER 2 e così via sono i relativi valori dei membri della dimensione per il POV.

Un esempio di file di input:

```
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:

Se per i form specificati in `requirement.csv` è richiesta l'impostazione di variabili utente, è necessario creare anche un file `UserVarMemberMapping.csv`. Fare riferimento alla sezione [Creazione del file UserVarMemberMapping.csv](#).

File di input per il salvataggio dei form

In questo file, utilizzato come riferimento in `requirement.csv` per supportare il salvataggio dei form, sono inclusi il POV e i valori di input delle celle nel formato seguente:

Un esempio di file di input:

```
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 questo esempio:

- DIM indica il nome di una dimensione e MEMBER il valore del membro della dimensione mentre
- CELL COLUMN HEADER identifica il nome dell'intestazione di colonna e CELL ROW HEADER il nome dell'intestazione di riga.

Ad esempio:

```
pov=[Version View:Working],[Sales Entity:International Sales];
cells=[FY16->x-----x->Pct|P293:Maintenance->4120:Support|1]
```



Note:

Se per i form specificati in `requirement.csv` è richiesta l'impostazione di variabili utente, è necessario creare anche un file `UserVarMemberMapping.csv`. Fare riferimento alla sezione [Creazione del file UserVarMemberMapping.csv](#).

Esecuzione del file di input di una regola business

In questo file, utilizzato come riferimento in `requirement.csv` per supportare l'esecuzione delle regole business, sono inclusi i valori dei parametri runtime nel formato:

```
rtp=[RTP1:Value1],[RTP2:Value2] e così via.
```

Se non è richiesto alcun valore dei parametri runtime per la regola business, includere `rtp=[]`. In questo esempio RTP1, RTP2 e così via identificano i nomi dei prompt runtime mentre VALUE1, VALUE2 ne identificano i valori. Accertarsi di aggiungere il numero di prompt runtime richiesto.

Un esempio di file di input:

```
rtp=[Period:Q1],[Entity:USA]
rtp=[Period:Q2],[Entity:USA]
rtp=[Period:Q3],[Entity:USA]
rtp=[Period:Q4],[Entity:USA]
```

Esecuzione del file di input della regola dati

In questo file, utilizzato come riferimento in `requirement.csv` per supportare l'esecuzione delle regole dati, è necessario specificare il periodo di inizio e quello di fine, la modalità di importazione e di esportazione e un nome facoltativo del file di importazione disponibile nell'ambiente. Se non si specifica un nome di file, verrà utilizzato il nome specificato nella regola dati. Formato delle singole righe:

```
startperiod=START PERIOD;endperiod=END PERIOD;importmode=IMPORT_MODE;exportmode=EXPORT_MODE;filename=FILE NAME.
```

Un esempio di file di input:

```
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
```

File di input della griglia ad hoc

In questo file, utilizzato come riferimento in `requirement.csv` per supportare l'apertura delle griglie ad hoc, deve essere specificata la griglia da aprire. Ogni riga del file deve essere nel formato indicato di seguito.

```
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,..]
```

Un esempio di file di input:

```
fileName=dropdown.xlsx#sheet4;pov=[HSP_View:BaseData],  
[Scenario:Forecast],[Product:No Product],[Entity:Sales Mid-  
Atlantic];rows = [ Account ]; cols= [Year, Period, Version]
```

 **Note:**

Il file identificato con `fileName` nel file CSV di input deve contenere la definizione della griglia ad hoc nel foglio specificato. Ad esempio, nel file di input precedente, come origine della definizione della griglia ad hoc viene specificato il valore `sheet4` di `dropdown.xlsx`. È necessario che questo file Excel sia disponibile, insieme con `requirement.csv` e il file CSV di input, nel file `INPUT_FILE.zip` utilizzato per eseguire il comando `simulateConcurrentUsage`.

Esecuzione del file di input del report

In questo file, utilizzato come riferimento in `requirement.csv` per supportare l'apertura dei report di gestione, deve essere specificato il report da aprire. Ogni riga del file deve essere nel formato indicato di seguito.

```
format=[REPORT_FORMAT];globalPov=[POV];prompts=[PROMPT_1],[PROMPT_2]
```

`globalPov` e `prompts` sono facoltativi.

 **Note:**

- I formati di report supportati sono `pdf` e `embedded`.
- Se il nome della dimensione `globalPov` o i suoi membri contengono il carattere due punti (`:`) o punto e virgola (`;`), farlo precedere dal carattere di escape `\`. Ad esempio il nome di dimensione `Version:View` deve essere specificato come `Version\\:View`.

Esempio di file di input per generare un pdf da un report con il POV globale `[Version View:Working],[Sales Entity:International Sales]` e i prompt `[Actual;Budget],[Year:2018]`:

```
format=pdf;globalPov=[Version View:Working],[Sales  
Entity:International Sales];prompts=[Actual;Budget],[Year:2018]
```

Esecuzione del file di input del registro

In questo file, utilizzato come riferimento in `requirement.csv` per supportare l'apertura dei registri in Report, deve essere specificato il registro da aprire. Ogni riga del file deve essere nel formato indicato di seguito.

```
format=BOOK_FORMAT
```

oppure

```
format=BOOK_FORMAT;globalPov=POV
```

Esempio di file di input per generare un pdf da un registro con il POV globale *[Version View:Working],[Sales Entity:International Sales]*:

```
format=pdf;globalPov=[Version View:Working],[Sales Entity:International Sales]
```

Note:

- I formati di registro supportati sono PDF e XLSX.
- Se il nome della dimensione `globalPov` o i suoi membri contengono il carattere due punti (:) o punto e virgola (;), farlo precedere dal carattere di escape `\`. Ad esempio il nome di dimensione `Version:View` deve essere specificato come `Version\\:View`.

Creazione del file UserVarMemberMapping.csv

Questo file è obbligatorio se per impostare i form specificati nel file di input dei casi d'uso di Form aperto o Salva form è richiesta l'impostazione di variabili utente. Questo file non è necessario per altri casi d'uso.

La prima riga di questo file è l'intestazione `#Dimension,User Variable,Member`

Le voci successive contengono il mapping di dimensione, variabile utente e membro dimensione.

Esempio di file `UserVarMemberMapping.csv`:

```
#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
```

Creazione e caricamento del file ZIP di input nell'ambiente

Utilizzare uno strumento come 7 Zip e creare un file ZIP contenente `requirement.csv`, i file di input dei casi d'uso corrispondenti e `UserVarMemberMapping.csv`, se necessario.

Utilizzare il comando `uploadFile` per caricare il file ZIP ottenuto nella cartella inbox dell'ambiente (esempio di sintassi del comando `epmautomate uploadFile "C:/uploads/INPUT_FILE.zip" inbox`) in cui eseguire la simulazione.

Esempio di report di simulazione dell'utilizzo concorrente

La simulazione dell'utilizzo concorrente viene inviata per impostazione predefinita all'utente che esegue il comando `simulateConcurrentUsage`. Se si specificano destinatari e-mail, il report viene inviato via e-mail solo a essi.

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

In questo report vengono identificate le informazioni indicate di seguito.

Colonna	Descrizione
Operation #	Il numero di sequenza del caso d'uso in <code>requirement.csv</code> .
Operation	Il tipo di operazione specificato in <code>requirement.csv</code> .
Artifact Name	Il nome dell'artifact specificato in <code>requirement.csv</code> .
Users	Il numero di utenti specificato in <code>requirement.csv</code> .
iterations	Il numero di esecuzioni del caso d'uso secondo quanto specificato nel parametro <code>iterations</code> .
Min. Duration	Il tempo minimo richiesto per l'esecuzione del caso d'uso da parte di un utente.
Max. Duration	Il tempo massimo richiesto per l'esecuzione del caso d'uso da parte di un utente.
Avg. Duration	Il tempo medio richiesto per l'esecuzione del caso d'uso da parte di un utente.
Return Status	Lo stato del caso d'uso. Se l'esecuzione del caso d'uso non riesce, viene visualizzato <code>Failed</code> .

B

Preparazione per l'esecuzione del comando di ripetizione

Il comando di ripetizione viene utilizzato per eseguire un test delle performance di un ambiente in caso di carico al fine di verificare che l'esperienza utente sia accettabile quando il servizio deve gestire un carico specifico. Prima di poter caricare gli ambienti di test, è necessario eseguire alcune procedure.

In questa appendice viene descritta la procedura che gli amministratori dei servizi devono completare prima di eseguire il comando di ripetizione di EPM Automate.

- [Informazioni sul comando di ripetizione](#)
- [Prerequisiti](#)
- [Creazione di file HAR](#)
- [Creazione di file di ripetizione](#)
- [Generazione di file di tracce](#)
- [Sessione di ripetizione di esempio](#)

Informazioni sul comando di ripetizione

Il comando di ripetizione consente di ripetere il caricamento di Oracle Smart View for Office, dell'API REST Oracle Enterprise Performance Management Cloud o di EPM Automate in un ambiente per consentire il test delle performance in caso di carichi di particolare entità al fine di verificare che l'esperienza utente sia accettabile quando l'ambiente è sottoposto a un carico specifico.

È ad esempio possibile eseguire il test dell'esperienza utente in un ambiente di test con un carico di particolare entità per assicurarsi che il servizio abbia performance ottimali in seguito alla migrazione dell'applicazione dall'ambiente di test all'ambiente di produzione.

Prerequisiti

Quando si esegue il comando utilizzando un file di ripetizione, EPM Automate esegue ciascuna riga del file di ripetizione in parallelo per esercitare un carico sul servizio in modo tale da poter eseguire test per verificare che l'esperienza utente sia accettabile quando il servizio deve gestire un carico rilevante.

- Identificare i form che richiedono maggiore impegno di elaborazione nell'ambiente. I candidati più idonei sono i form che devono gestire grandi quantità di dati oppure form che includono calcoli complessi, ad esempio, i form utilizzati per sottoporre previsioni, processi che coinvolgono la creazione di report ad hoc e report statistici possono generare carichi rilevanti sul servizio. Analogamente, attività quali l'esecuzione di regole business, di report, di API REST che richiedono molte risorse e di comandi EPM Automate (ad esempio runBusinessRule, runDataRule, exportData, exportMetadata, restructureCube) possono sottoporre l'ambiente a un carico ingente ed essere candidate per il test di carico.

- Installare Fiddler, se necessario. EPM Automate richiede un file in formato HAR (HTTP Archive) 1.1 contenente record dell'interazione di Oracle Smart View for Office, dell'API REST Oracle Enterprise Performance Management Cloud o di EPM Automate con l'ambiente EPM Cloud. In genere si utilizza Fiddler per generare il file HAR che acquisisce il log dell'interazione con EPM Cloud.
- Eseguire le principali attività identificate in precedenza. Utilizzare Smart View per eseguire attività quali l'apertura e il salvataggio di form, l'esecuzione di regole business e la creazione di report e utilizzare Fiddler per acquisire i dettagli relativi alle attività ed esportarli in file HAR. Analogamente, eseguire le API REST e i comandi EPM Automate e fare in modo che Fiddler acquisisca i dettagli. Per dettagli, fare riferimento alla sezione [Creazione di file HAR](#).
- Creare un file di ripetizione in formato CSV contenente l'elenco delle credenziali (nomi utente e password) e il nome dei file HAR da eseguire. Ogni riga del file può contenere il nome utente e la password di un utente univoco per simulare più sessioni utente simultanee. Per dettagli, fare riferimento alla sezione [Creazione di file di ripetizione](#).
L'utente le cui credenziali sono specificate in una riga per l'esecuzione di un file HAR non deve necessariamente corrispondere all'utente che ha eseguito la sessione utilizzata per creare il file HAR. Tale utente non deve tuttavia disporre dei diritti per eseguire queste attività nell'ambiente.

Per una procedura dettagliata relativa all'esecuzione del comando di ripetizione, fare riferimento alla sezione [Sessione di ripetizione di esempio](#).

Creazione di file HAR

Il file HAR acquisisce tracce dell'interazione di Oracle Smart View for Office, dell'API REST o di EPM Automate con Oracle Enterprise Performance Management Cloud.

Dal momento che Fiddler acquisisce le informazioni su tutto il traffico HTTP(S), durante la creazione dei file HAR evitare di eseguire attività che potrebbero aggiungere operazioni di traccia in Fiddler.

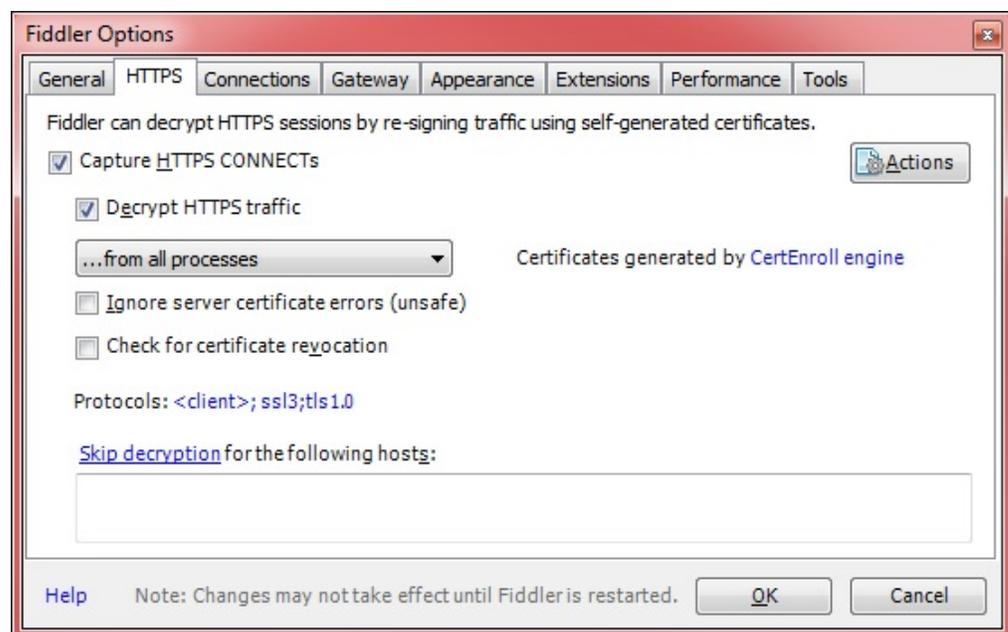
Per creare un file HAR, procedere nel seguente modo.

1. Avviare Fiddler.
2. Assicurarsi che Fiddler sia configurato per decifrare il traffico HTTPS da tutti i processi.
 - a. Selezionare **Strumenti**, quindi **Opzioni** e infine **HTTPS**.
 - b. Selezionare l'opzione **Decrypt HTTPS traffic**, se non è già selezionata.

In Fiddler vengono visualizzate le informazioni sul certificato principale utilizzato per intercettare il traffico HTTPS. In genere, questo certificato può essere considerato attendibile.

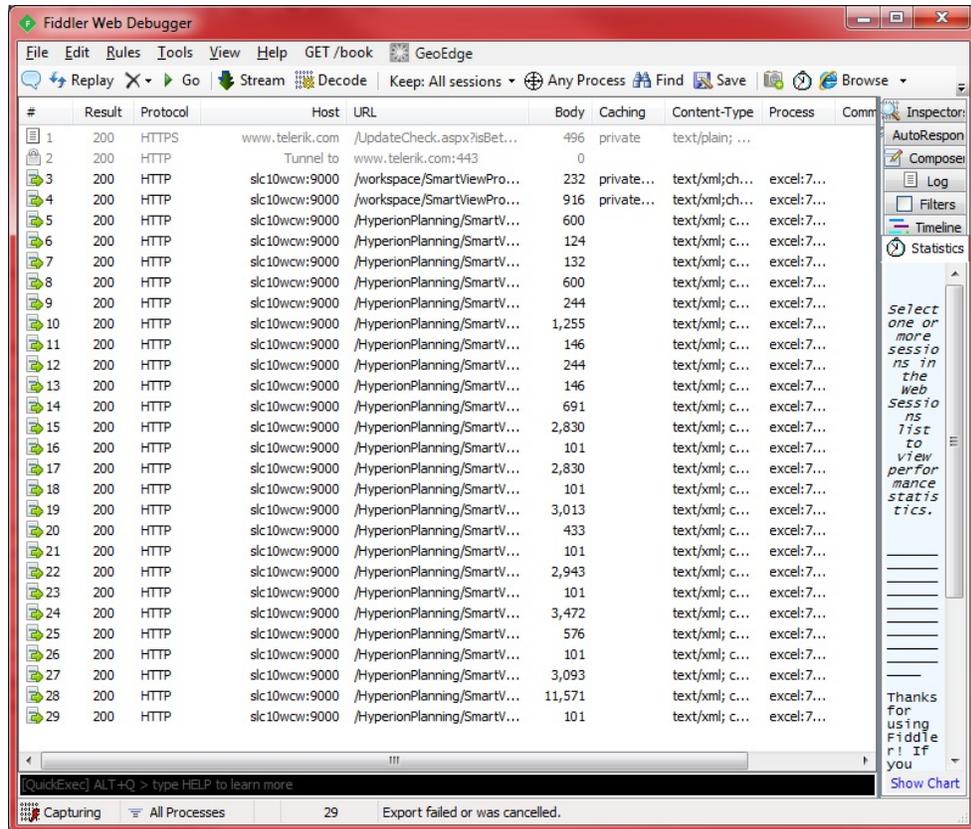


- c. Fare clic su **Yes** se si desidera aggiungere il certificato principale all'elenco di autorità di certificazione attendibili. In caso contrario, scegliere **No**.
- d. **Facoltativo:** se al passo precedente si seleziona **No**, è possibile selezionare **Ignore server certificate errors** per disattivare gli avvisi di sicurezza di Fiddler relativi alla decifrazione del traffico HTTPS.
- e. Fare clic su **OK**.

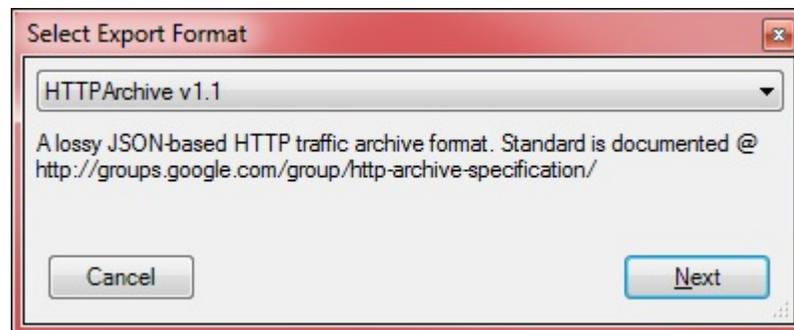


- 3. Avviare Smart View e accedere all'ambiente per il quale si desidera acquisire la traccia.
- 4. Utilizzando Smart View, l'API REST o EPM Automate, eseguire le attività che generano un carico di elaborazione rilevante sull'ambiente. Aprire ad esempio form in Smart View in modo che Fiddler possa registrare l'attività.

Fiddler registra i processi avviati.



5. Nello strumento Fiddler completare i passi indicati di seguito.
 - a. Selezionare File, Export Sessions, quindi All Sessions o Selected Sessions. In caso di connessione ad altri siti Web durante l'esecuzione di Fiddler, selezionare Selected Sessions per scegliere le sessioni rilevanti per l'ambiente.
 - b. In Select Export Format selezionare HTTPArchive v1.1 come formato di esportazione.
 - c. Fare clic su Next.



- d. In Export As HTTPArchive v1.1 selezionare la directory in cui si desidera memorizzare il file, quindi specificare un nome file.
 - e. Fare clic su Save.

Creazione di file di ripetizione

Un file di ripetizione è un file CSV contenente le credenziali (nome utente e password) e il nome dei file HAR da eseguire per caricare il sistema utilizzando il comando di ripetizione di EPM Automate.

Assicurarsi che il nome utente e la password specificati dispongano dei diritti di esecuzione delle attività incluse nel file HAR.

Durante l'esecuzione del comando di ripetizione, EPM Automate esegue ciascuna riga del file di ripetizione in parallelo per esercitare il carico sul servizio. Ad esempio, se il file di ripetizione contiene 10 righe, EPM Automate ripete 10 sessioni in modo da consentire l'esecuzione di test per verificare che l'esperienza utente sia accettabile quando il servizio deve gestire un carico specificato. Ogni attività inclusa nel file HAR viene eseguita in serie.

Per informazioni sull'esecuzione del comando di ripetizione, fare riferimento alla sezione [replay](#).

Per creare un file di ripetizione, procedere nel seguente modo.

1. Aprire Microsoft Office Excel e creare un nuovo foglio di lavoro.
2. Immettere un nome utente, una password e la posizione di un file HAR rispettivamente nelle colonne A, B e C della riga 1.

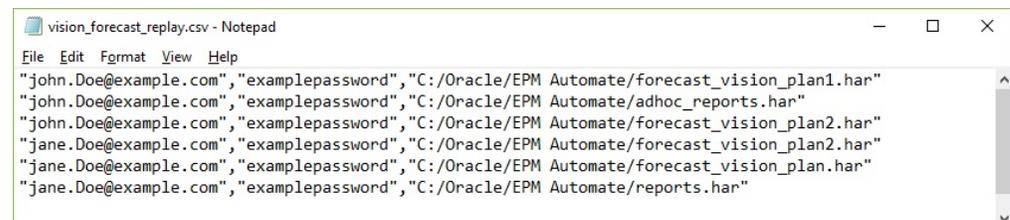
Ripetere questo passo per creare altre righe.

Nota:

È necessario specificare il percorso assoluto della posizione del file HAR. Utilizzare la barra (/) come separatore di directory nei percorsi di file; non utilizzare le barre rovesciate (\).

3. Salvare il file
4. In **Salva con nome** completare i passi indicati di seguito.
 - a. Selezionare la directory in cui si desidera memorizzare il file di ripetizione.
 - b. In **Nome file** specificare un nome e in **Salva come** selezionare CSV (delimitato dal separatore di elenco) (*.csv).
 - c. Fare clic su **Save**.

Un file di ripetizione di esempio può essere simile a quello riportato di seguito:



```

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"
  
```

Generazione di file di trace

Durante l'esecuzione del comando di ripetizione è possibile generare file di traccia da condividere con il Supporto Oracle per risolvere eventuali problemi riscontrati. Il Supporto Oracle utilizza i file di traccia per analizzare come l'ambiente ha gestito un'attività di Oracle Smart View for Office.

Utilizzare il parametro facoltativo `trace=true` con il comando di ripetizione per generare file di traccia in formato XML. Se si utilizza questo parametro, per ogni attività nel file HAR EPM Automate crea un file di traccia contenente la risposta di Smart View all'attività.

I file di traccia vengono denominati `trace-N.xml`; ad esempio `trace-1.xml`, dove `N` rappresenta un contatore che inizia da 1. Se nel file di ripetizione sono specificati più file HAR con lo stesso nome, EPM Automate consolida i file di traccia in una cartella.

I file di traccia correlati a un file HAR vengono memorizzati in una cartella all'interno della directory da cui si esegue EPM Automate. EPM Automate crea una cartella per ogni file HAR incluso nel file di ripetizione. Per denominare le cartelle, EPM Automate utilizza una combinazione di ora di sistema del server corrente e nome file HAR nel formato `YYYY_MM_DD_HH_MM_SS_HAR_FILE_NAME`. Ad esempio, se il nome del file HAR è `forecast1.har`, il nome della cartella può essere `2016_06_08_10_21_42_forecast1`.

Sessione di ripetizione di esempio

Descrive come eseguire il comando di ripetizione utilizzando più file HAR.

In questa sezione si presuppone che:

- Siano stati creati i seguenti file HAR. Ogni file HAR può contenere lo stesso set di attività. Per informazioni dettagliate, fare riferimento alla sezione [Creazione di file HAR](#).
 - `C:\Oracle\EPM Automate\forecast_vision_plan1.har`
 - `C:\Oracle\EPM Automate\forecast_vision_plan2.har`
 - `C:\Oracle\EPM Automate\forecast_plan2.har`
- Sia stato creato un file di ripetizione `C:/Oracle/EPM Automate/vision_forecast_replay.csv` con il seguente contenuto (per dettagli, fare riferimento alla sezione [Creazione di file di ripetizione](#)):

Nota:

Nel file di ripetizione, utilizzare la barra (/) come separatore di directory nei percorsi di file; non utilizzare le barre rovesciate (\).

```
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
```

Per eseguire il comando di ripetizione, procedere nel seguente modo.

1. In una finestra del prompt dei comandi passare alla directory di installazione di EPM Automate, ad esempio `C:\Oracle\EPM Automate\bin`.

2. Eseguire l'accesso a un ambiente come Amministratore servizi, quindi eseguire il comando di ripetizione:

```
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 visualizza le informazioni di ripetizione nella console e termina l'elaborazione alla fine della durata specificata (12 minuti nell'esempio precedente). Crea inoltre le cartelle e i file di traccia perché il comando precedente include il parametro `trace=true`.

Poiché il comando è stato eseguito da `C:\Oracle\EPM Automate\bin`, EPM Automate ha memorizzato i file di traccia nelle seguenti cartelle. Si noti che tali cartelle sono state denominate in base ai nomi dei file HAR.

- `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. Scollegarsi dall'ambiente:

```
epmautomate logout
```

C

Gestione dei caratteri speciali

È possibile che le password, le password proxy e i valori dei parametri dei comandi di Oracle Enterprise Performance Management Cloud contengano caratteri speciali. Per consentire a EPM Automate di gestire questi caratteri sono necessari interventi specifici.

Negli esempi contenuti in questa sezione viene utilizzata una password campione per illustrare l'utilizzo dei caratteri speciali.

Oracle consiglia di racchiudere le coppie parametro-valore tra virgolette doppie.

Windows

Per questi caratteri speciali è necessario utilizzare caratteri di escape racchiudendo tra virgolette doppie (") il carattere speciale o il valore di parametro contenente il carattere speciale.



Nota:

Non è possibile eseguire EPM Automate da una cartella contenente & nel relativo nome, ad esempio C:\Oracle\A&B.

Tabella C-1 Gestione dei caratteri speciali: Windows

Carattere	Descrizione	Esempio con carattere di escape
)	Parentesi chiusa	<ul style="list-style-type: none">Example") "pwd1 oppure"Example)pwd1 "
<	Minore di	<ul style="list-style-type: none">Example"<"pwd1 oppure"Example<pwd1 "
>	Maggiore di	<ul style="list-style-type: none">Example">"pwd1 oppure"Example>pwd1 "
&	E commerciale	<ul style="list-style-type: none">Example"&"pwd1 oppure"Example&pwd1 "
	Barra verticale	<ul style="list-style-type: none">Example" "pwd1 oppure"Example pwd1 "
"	Virgolette	<ul style="list-style-type: none">Example""pwd1 oppure"Example"pwd1 "

Utilizzo del punto esclamativo nelle password in testo normale nei file batch di Windows

È consigliabile gestire l'utilizzo del punto esclamativo (!) nelle password in testo normale nei file batch di Windows utilizzati con EPM Automate nel modo descritto di seguito.

1. Anteporre due simboli di accento circonflesso (^) al punto esclamativo come carattere di escape. Ad esempio, se la password è Welc0me!, codificarla nel seguente modo: Welc0me^^!
2. Aggiornare il file `bash` in modo da impostare `DisableDelayedExpansion` all'inizio del file includendo la seguente dichiarazione:

```
setlocal DisableDelayedExpansion
```
3. Rimuovere la dichiarazione `setlocal EnableExtensions EnableDelayedExpansion`, se presente nello script.

UNIX/Linux

Nei sistemi operativi UNIX e Linux, il carattere di escape da utilizzare per questi caratteri speciali è la barra rovesciata (\).

 **Nota:**

- Per il carattere di escape ! (punto esclamativo), utilizzare un apice intorno alla password oppure la barra rovesciata (\) come carattere di escape.
- Per il carattere di escape \, \$, ' e ", utilizzare la virgoletta doppia intorno alla password oppure la barra rovesciata (\) come carattere di escape.

Tabella C-2 Gestione dei caratteri speciali: UNIX/Linux

Carattere	Descrizione	Esempio con carattere di escape
(Parentesi aperta	Example \(pwd1
)	Parentesi chiusa	Example\)pwd1
<	Minore di	Example\<pwd1
>	Maggiore di	Example\>pwd1
`	Apostrofo	Example\'pwd1
!	Punto esclamativo	<ul style="list-style-type: none"> • 'Example!pwd1' oppure • Example\!pwd1
#	Cancelletto	Example\#pwd1
&	E commerciale	Example\&pwd1
	Barra verticale	Example\ pwd1
;	Punto e virgola	Example\;pwd1
.	Punto	Example\.pwd1
"	Virgolette	<ul style="list-style-type: none"> • Example\"pwd1 oppure • "Example\"pwd1"
'	Virgoletta singola	<ul style="list-style-type: none"> • Example\'pwd1 oppure • "Example\'pwd1"
\$	Segno di dollaro	<ul style="list-style-type: none"> • Example\ \$pwd1 oppure • "Example\ \$pwd1"

Tabella C-2 (Cont.) Gestione dei caratteri speciali: UNIX/Linux

Carattere	Descrizione	Esempio con carattere di escape
\	Barra rovesciata	<ul style="list-style-type: none"> • Example\\pwd1 oppure • "Example\\pwd1"

Utilizzo del punto esclamativo nelle password in testo normale negli script di UNIX o Linux

Negli script di UNIX/Linux, se una password di EPM Automate memorizzata in una variabile shell contiene caratteri speciali, utilizzare tre barre rovesciate come sequenza di escape e racchiudere la stringa tra virgolette doppie. Ad esempio, la password `lzi[ACO(e*7Qd)jE` inclusa nella variabile shell `password` deve essere resa nello script come segue:

```
password="lzi[ACO\\(e*7Qd\\)jE"
```

D

Comandi specifici di ciascun servizio EPM Cloud

- [Comandi di Account Reconciliation](#)
- [Comandi di Financial Consolidation and Close](#)
- [Comandi di Narrative Reporting](#)
- [Comandi di Oracle Enterprise Data Management Cloud](#)
- [Comandi di Planning, Planning Modules, FreeForm, Strategic Workforce Planning e Sales Planning](#)
- [Comandi di Profitability and Cost Management](#)
- [Comandi di Enterprise Profitability and Cost Management](#)
- [Comandi di Tax Reporting](#)

Comandi di Account Reconciliation

Comandi di EPM Automate per 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

Comandi di Financial Consolidation and Close

Comandi di EPM Automate per 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		

Comandi di Narrative Reporting

Comandi di EPM Automate per 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

Comandi di Oracle Enterprise Data Management Cloud

Comandi di EPM Automate per 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

Comandi di Planning, Planning Modules, FreeForm, Strategic Workforce Planning, e Sales Planning

Comandi EPM Automate per Planning, Planning Modules, FreeForm, Strategic Workforce Planning e Sales Planning

addUsers	getApplicationAdminMode	replay
addUsersToGroup	getDailyMaintenanceStartTime	resetService
addUserToGroups	getEssbaseQryGovExecTime	restoreBackup
applicationAdminMode	getIdleSessionTimeout	restructureCube
assignRole	getIPAllowlist	roleAssignmentAuditReport
autoPredict * Fare riferimento alla nota a piè di pagina	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

* Questo comando non è supportato per FreeForm, Strategic Workforce Planning e Sales Planning.

** Questo comando non è supportato per FreeForm.

Comandi di Profitability and Cost Management

Comandi di EPM Automate per 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

Comandi di Enterprise Profitability and Cost Management

Comandi di EPM Automate per 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		

Comandi di Tax Reporting

Comandi di EPM Automate per 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
