

Oracle® Fusion Cloud EPM

EPM Automate 작업



F28904-27



Oracle Fusion Cloud EPM EPM Automate 작업NOT_SUPPORTED

F28904-27

Copyright © 2016, 2025, Oracle and/or its affiliates.

주요 작성자: 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.

목차

설명서 접근성

설명서 피드백

1 EPM Automate 정보

EPM Automate 설치	1
용량 및 포트 요구사항	2
지원되는 플랫폼	2
Java Runtime Environment 및 EPM Automate	4
OpenJDK 사용	4
Windows 지침	4
Linux/UNIX/macOS X 지침	5
서버측 EPM Automate 명령 실행	5
EPM Automate 업데이트	6
EPM Automate 설치 해제	6
EPM Automate 암호화 레벨 이해	6
OAuth 2.0 인증 프로토콜 사용	6

2 명령 참조

EPM Automate 명령 실행 정보	1
필수 조건	1
기본 파일 위치	2
Transport Layer Security Protocol 사용	4
EPM Automate 명령 사용	4
매개변수 파일을 사용하여 명령 실행을 위한 값 전달	4
일부 Enterprise Profitability and Cost Management 명령에서 대체 변수 사용	7
하나의 매개변수에 여러 값 지정	8
일별 유지관리 중 동작	8
EPM Automate 실행	8
Windows	9
Linux	10

EPM Automate의 여러 인스턴스 실행	10
명령 개요	13
EPM Automate 명령	18
addUsers	18
addUsersToGroup	19
addUsersToTeam	20
addUserToGroups	21
applicationAdminMode	22
applyDataGrants	23
archiveTmTransactions	23
assignRole	25
autoPredict	26
calculateModel	27
clearCube	29
clearDataByPointOfView	29
clearDataByProfile	30
clearPOV	31
cloneEnvironment	32
compactCube	36
copyDataByPointOfView	37
copyDataByProfile	38
copyFileFromInstance	38
copyFromObjectStorage	39
copyFromSFTP	40
copyOwnershipDataToNextYear	41
copyPOV	42
copySnapshotFromInstance	43
copyToObjectStorage	44
copyToSFTP	45
createGroups	46
createNRSnapshot	47
createReconciliations	47
deleteFile	48
deleteGroups	49
deletePointOfView	50
deletePOV	50
deployCube	51
deployEJTemplates	52
deployFormTemplates	53
deployTaskManagerTemplate	53
dismissIPMInsights	54
downloadFile	55

enableApp	55
enableQueryTracking	56
encrypt	56
essbaseBlockAnalysisReport	57
executeAggregationProcess	58
executeBurstDefinition	59
executeReportBurstingDefinition	60
exportAccessControl	60
exportAppAudit	61
exportAppSecurity	62
exportARApplicationProperties	62
exportBackgroundImage	63
exportCellLevelSecurity	64
exportConsolidationJournals	64
exportData	65
exportDataManagement	65
exportDimension	66
exportDimensionMapping	67
exportEJournals	67
exportEssbaseData	69
exportJobConsole	69
exportLibraryArtifact	72
exportLibraryDocument	73
exportLogoImage	74
exportMapping	75
exportMetadata	75
exportOwnershipData	76
exportQueryResults	76
exportSnapshot	78
exportTemplate	79
exportTaskManagerAccessControl	80
exportValidIntersections	80
extractDimension	81
extractPackage	82
feedback	83
getApplicationAdminMode	84
getDailyMaintenanceStartTime	85
getEssbaseQryGovExecTime	85
getIdleSessionTimeout	86
getIPAllowlist(사용되지 않음)	86
getRestrictedDataAccess	87
getSubstVar	88

getVirusScanOnFileUploads	88
groupAssignmentAuditReport	89
help	90
importAppAudit	90
importAppSecurity	91
importARApplicationProperties	92
importBackgroundImage	92
importBalances	93
importCellLevelSecurity	93
importConsolidationJournals	94
importData	95
importDataManagement	96
importDimension	96
importEJDimensionMembers	97
importJobConsole	98
importLibraryArtifact	99
importLibraryDocument	100
importLogoImage	101
importMapping	101
importMetadata	102
importOwnershipData	103
importPreMappedBalances	104
importPreMappedTransactions	105
importProfiles	105
importRates	106
importRCAAttributeValues	106
importReconciliationAttributes	107
importSDMDimensionMembers	108
importSnapshot	109
importSupplementalCollectionData	111
importSupplementalData	113
importTemplate	113
importTMAttributeValues	114
importTmPremappedTransactions	115
importValidIntersections	116
invalidLoginReport	117
listBackups	118
listFiles	119
loadData	120
loadDimensionViewpoint	121
loadDimData	122
loadViewpoint	123

login	123
logout	127
maskData	127
mergeDataSlices	128
mergeSlices	128
optimizeASOCube	129
programDocumentationReport	130
provisionReport	131
purgeArchivedTmTransactions	132
purgeTmTransactions	133
recomputeOwnershipData	134
recreate	135
refreshCube	137
refreshReportPackageDatasource	138
removeUserFromGroups	138
removeUsers	139
removeUsersFromGroup	140
removeUsersFromTeam	141
renameSnapshot	142
replay	143
resetService	144
restoreBackup	145
restructureCube	146
roleAssignmentAuditReport	146
roleAssignmentReport	148
runAutoAlert	149
runAutomatch	150
runBatch	150
runBusinessRule	151
runCalc	152
runComplianceReport	153
runDailyMaintenance	154
runDataRule	155
runDMReport	157
runIntegration	158
runIntercompanyMatchingReport	162
runMatchingReport	163
runPipeline	164
runPlanTypeMap	165
runRuleSet	166
runSupplementalDataReport	166
runTaskManagerReport	167

sendMail	168
setApplicationAdminMode	169
setDailyMaintenanceStartTime	170
setDemoDates	171
setEJJournalStatus	172
setEncryptionKey	173
setEssbaseQryGovExecTime	173
setIdleSessionTimeout	174
setIPAllowlist(사용되지 않음)	174
setManualDataAccess	176
setPeriodStatus	177
setRestrictedDataAccess	177
setSubstVars	178
setVirusScanOnFileUploads	179
simulateConcurrentUsage	179
skipUpdate	183
snapshotCompareReport	185
sortMember	186
unassignRole	187
updateGuidedLearningSettings	188
updateUsers	189
upgrade	190
uploadFile	191
userAuditReport(사용되지 않음)	192
userGroupReport	193
userLoginReport	194
validateConsolidationMetadata	195
validateModel	195
종료 코드	196

3 명령 실행 샘플 시나리오

샘플 스크립트 복사 정보	1
모든 서비스의 샘플 시나리오	1
애플리케이션 스냅샷을 컴퓨터에 백업	2
사용자에게 일별 유지관리 완료에 대해 알리기	5
Oracle Object Storage로 또는 Oracle Object Storage에서 스냅샷 복사	12
사용자 생성 및 사전 정의된 역할에 사용자 지정	14
라이센스를 가진 사용자(역할에 지정된 사용자) 수 계산	17
역할에 지정된 사용자 감사 보고서 생성	19
역할 지정 및 취소 감사 보고서 생성	23
개인 정보 보호 법률을 준수하도록 액세스 로그 및 활동 보고서 마스크	26

로컬 컴퓨터로 활동 보고서 다운로드 자동화	31
환경에서 액세스 로그 다운로드	35
환경 복제 자동화	38
기본 환경에서 일별 유지관리가 완료된 후 매일 기본 환경에서 대기 환경으로 복제	42
환경에서 불필요한 파일 제거	50
환경에서 파일 찾기 및 다운로드	52
감사를 위해 이전 Cloud EPM 환경 재생성	53
데이터베이스 액세스 감사 및 준수 자동화	64
사용자 및 사전 정의된 역할 지정 복제	74
다른 ID 도메인에 ID 도메인의 사용자 복제	75
환경 간에 사전 정의된 역할 지정 복제	81
분기별 Cloud EPM 업그레이드 주기 생성	88
Windows 스크립트 및 지침	89
UNIX/Linux 스크립트 및 지침	92
Groovy 스크립트	96
6주 테스트 주기로 분기별 Cloud EPM 업그레이드 주기 생성	99
Planning, Consolidation, Tax Reporting, Enterprise Profitability and Cost Management 샘플 시나리오	113
집계 저장영역 큐브에서 다수의 셀 익스포트 자동화	114
애플리케이션으로 메타데이터 импорт	123
데이터 импорт, 계산 스크립트 실행 및 블록 저장영역 데이터베이스에서 집계 저장영역 데이터베이스로 데이터 복사	125
메타데이터와 데이터를 익스포트 및 다운로드	128
애플리케이션 데이터 익스포트 및 다운로드	130
애플리케이션 감사 레코드 아카이브 자동화	132
Windows 스크립트	133
Linux 스크립트	134
데이터 파일을 환경으로 업로드 및 데이터 로드 규칙 실행	135
일일 데이터 통합 자동화	138
Account Reconciliation의 샘플 시나리오	140
기간에 미리 형식 지정된 잔액 로드	140
백업 스냅샷 업로드 및 импорт	142
오래된 일치 트랜잭션 아카이브 및 아카이브된 트랜잭션 비우기	144
Profitability and Cost Management의 샘플 시나리오	150
애플리케이션으로 메타데이터 импорт	150
데이터 импорт 및 프로그램 규칙 실행	152
Cloud EDM의 샘플 시나리오	155
Cloud EDM 차원 및 매핑을 Cloud EPM 애플리케이션과 동기화	155
Cloud EPM 차원을 Cloud EDM 애플리케이션과 동기화	157
스크립트 실행 자동화	158
EPM Automate 활동 모니터	159

4 EPM Automate를 설치하지 않고 명령 실행

서버측 명령 실행을 지원하는 환경	1
정보 소스	2
지원되는 명령	2
서버측 Groovy를 사용하여 EPM Automate를 실행하는 데 사용할 방법	2
서버측 Groovy 스크립트를 사용하여 환경 복제	3
서버측 Groovy 스크립트를 사용하여 활동 보고서를 전자메일로 보내기	4

5 Cloud EPM 환경 복제

일별 복제 설정	1
주문형 복제 설정	2
보조 환경 구성	2

A simulateConcurrentUsage 명령 실행 준비

requirement.csv 파일 생성	A-2
입력 파일 생성	A-3
양식 입력 파일 열기	A-3
양식 입력 파일 저장	A-4
비즈니스 규칙 입력 파일 실행	A-6
비즈니스 규칙 세트 입력 파일 실행	A-6
데이터 규칙 입력 파일 실행	A-6
임시 그리드 입력 파일	A-7
보고서 입력 파일 실행	A-7
장부 입력 파일 실행	A-8
UserVarMemberMapping.csv 파일 생성	A-8
options.xml 파일 생성	A-9
users.csv 파일 생성	A-9
입력 ZIP 파일 생성 및 환경에 업로드	A-9
샘플 동시 사용 시뮬레이션 보고서	A-9

B Replay 명령 실행 준비

Replay 명령 정보	B-1
필수 조건	B-1
HAR 파일 생성	B-2
재생 파일 생성	B-5
추적 파일 생성	B-5
샘플 재생 세션	B-6

C 특수 문자 처리

D 개별 Cloud EPM 서비스 관련 명령

Account Reconciliation 명령	D-2
Financial Consolidation and Close 명령	D-3
Narrative Reporting 명령	D-4
Cloud EDM 명령	D-5
Planning, Planning 모듈, 무형식, Strategic Workforce Planning, Sales Planning 명령	D-6
Profitability and Cost Management 명령	D-7
Enterprise Profitability and Cost Management 명령	D-8
Tax Reporting 명령	D-9

설명서 접근성

오라클의 접근성 개선 노력에 대한 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>에서 Oracle Accessibility Program 웹 사이트를 방문하십시오.

오라클 고객지원센터 액세스

지원 서비스를 구매한 오라클 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>를 참조하거나, 청각 장애가 있는 경우 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>를 방문하십시오.

설명서 피드백

이 문서에 대한 피드백을 제공하려면 모든 Oracle 도움말 센터 항목의 페이지 맨아래에 있는 [피드백] 버튼을 누릅니다. epmdoc_ww@oracle.com으로 전자메일을 보낼 수도 있습니다.

1

EPM Automate 정보

EPM Automate를 사용하면 사용자가 Oracle Fusion Cloud Enterprise Performance Management 환경 내에서 원격으로 태스크를 수행할 수 있습니다.

Cloud EPM 서비스 관리자는 다음을 비롯한 많은 반복가능 태스크를 자동화할 수 있습니다.

- 메타데이터, 데이터, 아티팩트 및 애플리케이션 스냅샷, 템플릿 및 데이터 관리 매핑 임포트 및 익스포트
- 환경으로 파일 업로드, 파일 나열 및 서비스에서 파일 삭제
- 서비스에서 스냅샷, 보고서, 메타데이터 및 데이터 파일 다운로드
- 데이터에 대해 비즈니스 규칙 실행 및 애플리케이션 새로고침
- 한 데이터베이스에서 다른 데이터베이스로 데이터 복사. 일반적으로, 블록 저장영역 데이터베이스에서 집계 저장영역 데이터베이스로 복사 또는 블록 저장영역 데이터베이스에서 다른 블록 저장영역 데이터베이스로 복사
- Data Management बै치 규칙 실행
- Data Management 보고서, 프로비저닝 보고서 및 사용자 감사 보고서 생성
- 사전 매핑된 잔액 데이터, 환율, 사전 매핑된 트랜잭션, 잔액 데이터 및 프로파일을 Account Reconciliation으로 임포트
- 조정 프로세스를 시작할 기간에 프로파일 복사
- Profitability and Cost Management 애플리케이션의 계산 큐브 배포
- Enterprise Profitability and Cost Management 및 Profitability and Cost Management 애플리케이션에서 POV 지우기, 복사, 삭제
- 환경에서 Oracle Smart View for Office 또는 REST API 로드를 재생하여 과도한 로드 상태에서 성능 테스트 사용
- 파일의 보충 데이터를 Financial Consolidation and Close로 임포트

다양한 태스크를 완료할 수 있는 스크립트를 생성한 다음 스케줄러를 사용하여 실행을 자동화할 수 있습니다. 예를 들어 환경에서 일별 유지관리 백업을 다운로드하는 스크립트를 생성하여 아티팩트 및 데이터의 로컬 백업을 생성할 수 있습니다.



[자습서: EPM Automate를 사용하여 Planning 명령을 실행하는 방법](#)

EPM Automate 설치

EPM Automate를 설치하여 명령을 실행합니다. 일부 명령은 EPM Automate를 설치하지 않고도 Oracle Fusion Cloud Enterprise Performance Management에서 Groovy 스크립트를 사용하여 직접 실행할 수도 있습니다.

Cloud EPM 환경에서는 Windows, Linux/UNIX, macOS X용 EPM Automate 설치 프로그램을 제공합니다.

Windows 버전 10 이상에서는 Windows 관리자만 EPM Automate 설치가 허용되므로 Windows 관리자만 설치 및 업그레이드할 수 있습니다. EPM Automate를 설치한 사용자 또는 다른 Windows 관리자가 업그레이드를 할 수 있습니다.

이 섹션의 내용:

- [용량 및 포트 요구사항](#)
- [지원되는 플랫폼](#)
- [Java Runtime Environment 및 EPM Automate](#)
- [OpenJDK 사용](#)
- [Windows 지침](#)
- [Linux/UNIX/macOS X 지침](#)
- [서버측 EPM Automate 명령 실행](#)

용량 및 포트 요구사항

EPM Automate는 경량 클라이언트이므로 대형 클라이언트 풋프린트가 필요하지 않습니다. 모든 처리는 Oracle Fusion Cloud Enterprise Performance Management에서 발생합니다.

보안 HTTP 연결을 통해 외부 호스트에 액세스할 수 있는 표준 클라이언트 머신, 가상 머신, OIC(Oracle Integration Cloud) 머신에 EPM Automate를 설치할 수 있습니다.

EPM Automate는 표준 TLS 포트(포트 443)를 사용하여 Cloud EPM에 연결합니다. EPM Automate에 대한 추가 송신 포트를 열 필요가 없습니다.

현재 EPM Automate는 상호 TLS(mTLS) 인증을 지원하지 않습니다.

지원되는 플랫폼

EPM Automate는 보안 HTTP 연결을 통해 외부 호스트에 액세스할 수 있는 가상 머신과 OIC(Oracle Integration Cloud) 머신에 설치할 수 있습니다.

Note

- EPM Automate는 현재 운영 체제 공급업체에서 지원하는 64비트 운영 체제에서만 사용할 수 있습니다.
- EPM Automate는 SOCKS 프록시에서는 작동하지 않습니다. HTTP/HTTPS 프록시에서만 작동합니다.
- EPM Automate는 프록시 서버 연결에 기본, 다이제스트, Kerberos, Negotiate, NTLM 인증 메커니즘을 지원합니다.
- EPM Automate는 Google APIGEE, IBM Data Power, 기타 역방향 프록시 서버 등 API 게이트웨이를 통해 Oracle Fusion Cloud Enterprise Performance Management에 연결할 수 있습니다.

이렇게 하려면 /epmcloud 같은 컨텍스트 없이 Cloud EPM 환경의 URL로 타겟을 설정하여 게이트웨이 또는 역방향 프록시를 구성하십시오. 예를 들어 `https://epm-idDomain.epm.dataCenterRegion.oraclecloud.com`이 있습니다. 그런 다음 [login](#) 명령에서 Cloud EPM URL 대신 역방향 프록시 URL을 사용하십시오. 구성 정보는 게이트웨이 또는 프록시 서버 설명서를 참조하십시오.

프록시 설정을 구성할 때 응답 코드를 수정하지 않고 Cloud EPM에서 EPM Automate로 전달하여 EPM Automate에서 200, 206, 400, 404, 500, 501 등의 응답 코드를 올바르게 처리할 수 있도록 해야 합니다. 예를 들어 IBM Datapower의 경우 proxy HTTP Response를 ON으로 설정합니다. 또한 API 게이트웨이는 HTTP 메소드 (GET, POST, PUT, PATCH, DELETE)를 허용해야 합니다.

Linux 및 UNIX 컴퓨터에서는 EPM Automate가 다음 환경 변수를 찾아 HTTP 또는 HTTPS 프록시 설정을 확인합니다.

- proxyHost
- proxyPort

http 프록시 설정 예:

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

```
export proxyPort=8000
```

https 프록시 설정 예:

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

```
export proxyPort=8080
```

Note

EPM Automate는 특히 명령 실행 자동화의 경우 OAuth 2.0 인증 프로토콜을 통해 Cloud EPM 환경(OAuth에 대해 구성된 경우)에 액세스하여 명령을 실행할 수 있습니다.

기본 인증을 사용하는 환경에서는 EPM Automate가 회사 SSO(ID 제공자) 인증서로 작동하지 않습니다. 사용자가 회사 인증서를 사용하여 로그인할 수 없으므로 EPM Automate에 액세스하는 데 필요한 사용자 계정을 ID 도메인에서 유지관리해야 합니다. SSO에 대해 구독을 구성한 경우 EPM Automate 사용자도 해당 ID 도메인 인증서로 로그인할 수 있도록 설정해야 합니다. *Oracle Cloud Identity Management 관리*에서 ID 도메인 인증서로 사인인 사용을 참조하십시오.

다운로드 지침: *관리자용 시작 가이드*의 클라이언트 다운로드 및 설치

Java Runtime Environment 및 EPM Automate

Windows에 EPM Automate를 설치하면 필요한 JRE(Java Runtime Environment)가 설치됩니다. 그러나 Linux, Unix, macOS X 설치 프로그램에는 JRE가 포함되어 있지 않습니다. EPM Automate를 사용하려면 JRE 설치(버전 1.7)에 대한 액세스 권한이 있어야 합니다.

Java SE 구독을 별도로 구매하지 않아도 EPM Automate와 함께 Oracle Java Standard Edition(SE)을 사용할 수 있습니다. EPM Automate와 함께 제공되는 Oracle JDK 라이선스에 대한 자세한 내용은 [오라클 고객지원센터 문서 1557737.1 "다른 Oracle 제품의 일부로 사용되는 경우 Java SE 자격 지원"](#)을 참조하십시오.

OpenJDK 사용

Linux, Unix, macOS X 플랫폼에서는 JRE 대신 OpenJDK 버전 17을 사용할 수 있습니다.

Oracle의 프로덕션용 GPL 라이선스 무료 JDK인 OpenJDK는 <https://openjdk.java.net>에서 다운로드할 수 있습니다. 이 웹 사이트에서 OpenJDK 설치 지침도 확인할 수 있습니다.

EPM Automate 세션을 시작하기 전에 JAVA_HOME 환경 변수가 OpenJDK 설치를 가리키도록 설정합니다.

macOS X 예(Bash 셸 가정) - 홈 디렉토리에 설치된 OpenJDK 버전 17를 사용합니다.

```
cd ~/
export JAVA_HOME=$(/usr/jdk-17.jdk/Contents/Home)
```

Linux 예(Bash 셸 가정) - 홈 디렉토리에 설치된 OpenJDK 버전 17를 사용합니다.

```
cd ~/
export JAVA_HOME=/openjdk/jdk-17.0.12
```

Windows 지침

기본적으로 EPM Automate는 C:\Program Files\Oracle\EPM Automate 폴더에 설치됩니다. 관리자 Windows 그룹에 속한 사용자만 전체 권한을 가지는 보호된 폴더입니다. EPM Automate를 설치하려면 관리자 Windows 그룹의 멤버여야 합니다.

EPM Automate를 설치하려면 다음을 수행합니다.

1. 브라우저를 사용하여 EPM Automate를 설치할 Windows 컴퓨터의 환경에 액세스합니다.
2. 홈 페이지에서 사용자 이름을 눌러 **설정 및 작업**에 액세스합니다.
3. **다운로드**를 누릅니다.
4. 다운로드 페이지의 EPM Automate 섹션에서 **Windows용 다운로드**를 누릅니다.
5. 설치 프로그램을 컴퓨터에 저장합니다.
6. 설치 프로그램(EPM Automate.exe)을 마우스 오른쪽 버튼으로 누르고 **관리자로 실행**을 선택합니다.
7. **사용자 계정 제어**에서 **예**를 누릅니다.
8. 화면에 표시되는 프롬프트에 따라 설치를 완료합니다.

Linux/UNIX/macOS X 지침

EPM Automate를 사용하려면 Java 17 배포에 대한 액세스 권한이 필요합니다. 환경 변수 `JAVA_HOME`은 해당 JRE 설치를 가리키도록 설정되어야 합니다.

EPM Automate를 설치하려면 다음을 수행합니다.

1. 환경에 액세스합니다.
2. 홈 페이지에서 사용자 이름을 눌러 **설정 및 작업**에 액세스합니다.
3. 다운로드를 누릅니다.
4. 다운로드 페이지의 EPM Automate 섹션에서 **Linux/macOS X용 다운로드**를 누릅니다.
5. 읽기/쓰기/실행 권한이 있는 디렉토리에 설치 프로그램(`EPMAutomate.tar`)을 저장합니다.
6. 다음 단계를 완료합니다.
 - a. 설치 프로그램의 콘텐츠를 추출합니다.
 - b. 필요한 환경 변수를 설정합니다.
 - c. 디렉토리를 설치 프로그램의 콘텐츠를 추출한 디렉토리 내 `epmautomate/bin`으로 변경합니다.
 - d. `epmautomate.sh`를 실행합니다.

다음 스크립트에서는 필요한 경우 실제 디렉토리 경로를 사용해야 합니다. 예를 들어 `EPMAutomate.tar`이 `HOME/oracle/epmautomate/bin`으로 추출된 경우 `EPMAutomate_extract_directory`의 값으로 사용합니다.

macOS X 예(Bash 셸 가정) - 홈 디렉토리에서 설치하고 실행합니다.

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

Linux 예(Bash 셸 가정) - 홈 디렉토리에서 설치하고 실행합니다. Java 버전 17.0.12로 가정합니다.

```
cd ~/
tar xf directory_containing_downloaded_installer/EPMAutomate.tar
export JAVA_HOME=/opt/jdk_17.0.12
export PATH ~/Downloads/epmautomate/bin:$PATH
cd EPMAutomate_extract_directory/epmautomate/bin
./epmautomate.sh
```

서버측 EPM Automate 명령 실행

일부 EPM Automate 명령은 Oracle Fusion Cloud Enterprise Performance Management에서 Groovy를 사용하여 직접 실행할 수 있습니다. Groovy 스크립트를 사용하여 명령을 실행하기 위해 EPM Automate를 설치할 필요는 없습니다.

서버측 명령 실행은 EPM Automate 명령을 실행하기 위해 클라이언트 컴퓨터에서 Groovy 스크립트를 실행하는 것과 다릅니다.

자세한 내용은 [EPM Automate를 설치하지 않고 명령 실행](#)을 참조하십시오.

EPM Automate 업데이트

EPM Automate를 사용하면 최신 버전으로 자동 업그레이드할 수 있습니다.

로그인하여 세션을 시작한 후 EPM Automate가 현재 설치된 버전을 확인합니다. 설치된 버전이 최신 버전이 아닌 경우 새 버전을 사용할 수 있다는 메시지가 표시됩니다.

EPM Automate를 자동으로 업그레이드하려면 [upgrade](#) 명령을 실행합니다.

EPM Automate 설치 해제

Note

클라이언트 컴퓨터에서 EPM Automate를 제거하려는 경우에만 다음 단계를 사용하십시오. 현재 EPM Automate 버전을 업데이트하려면 [EPM Automate 업데이트](#)의 지침을 사용하십시오.

Windows

EPM Automate를 설치 해제하려면 다음을 수행합니다.

1. 설정에서 **앱, 설치된 앱** 순으로 엽니다.
2. 설치된 애플리케이션 목록에서 EPM Automate를 찾습니다.
3. 오버플로우 메뉴에서 **설치 해제**를 선택합니다.
4. **설치 해제**를 누릅니다.
5. 설치 디렉토리(일반적으로 C:\Program Files\Oracle\EPM Automate)에서 모든 남아 있는 파일을 삭제합니다.

Linux/UNIX/macOS X

EPM Automate 디렉토리 및 `EPMAutomate.tar` 파일을 삭제합니다.

EPM Automate 암호화 레벨 이해

Oracle Fusion Cloud Enterprise Performance Management에서는 강력한 암호화와 함께 TLS(Transport Layer Security) 1.2 또는 1.3을 사용하여 EPM Automate를 사용하는 통신을 보호합니다.

OAuth 2.0 인증 프로토콜 사용

EPM Automate는 특히 명령 실행 자동화의 경우 OAuth 2.0 인증 프로토콜을 통해 Oracle Fusion Cloud Enterprise Performance Management 환경에 액세스하여 명령을 실행할 수 있습니다.

OAuth 2.0 액세스를 사용으로 설정하려면 ID 도메인 관리자가 Oracle Cloud ID 서비스에서 애플리케이션을 공용 클라이언트로 등록해야 합니다. OAuth는 구독이 아닌 애플리케이션에 적용됩니다.

OAuth 2.0 설정에 대한 자세한 지침은 *Oracle Fusion Cloud EPM용 REST API 가이드*의 OAuth 2를 사용하여 인증을 참조하십시오.

Note

환경에 OAuth가 사용으로 설정된 경우에도 기본 인증이 작동합니다. 나중에 사용하려면 기존의 암호화된 비밀번호 파일을 덮어쓰지 않아야 합니다.

새로고침 토큰 및 클라이언트 ID가 포함된 암호화된 비밀번호 파일 생성

EPM Automate에서 환경에 액세스하는 데 OAuth 2.0을 사용하려는 서비스 관리자는 암호화된 비밀번호 파일을 생성하는 데 다음 세부정보가 있어야 합니다. 파일이 생성되면 환경에 로그인하는 데 사용됩니다.

- 새로고침 토큰
새로고침 토큰을 가져오는 방법에 대한 자세한 지침은 *Oracle Fusion Cloud EPM용 REST API 가이드의 OAuth 2로 인증 - OCI(Gen 2) 환경만 해당*을 참조하십시오.
- 클라이언트 ID
ID 도메인 관리자가 OAuth에 대해 애플리케이션을 구성하면 클라이언트 ID가 생성됩니다. 애플리케이션 구성 탭의 **일반 정보**에 표시되어 있습니다.

OAuth 인증에 사용되는 암호화된 비밀번호 파일을 생성하려면 다음을 수행합니다.

1. EPM Automate 세션을 시작합니다.
2. 다음과 유사한 명령을 실행합니다.

```
epmautomate encrypt REFRESH_TOKEN ENCRYPTION_KEY PASSWORD_FILE
```

ClientID=CLIENT_ID. 여기서 REFRESH_TOKEN은 보안 저장소의 암호 해독된 새로고침 토큰이고, ENCRYPTION_KEY는 비밀번호를 암호화하기 위한 개인 키이고, PASSWORD_FILE은 암호화된 새로고침 토큰을 저장하는 파일의 이름 및 위치입니다. 비밀번호 파일은 .epw 확장자를 사용해야 합니다.

자세한 지침은 [encrypt](#)를 참조하십시오.
3. 새로 생성된 비밀번호 파일을 사용하여 OAuth로 로그인합니다. 자동화된 스크립트 실행의 경우 새로 생성된 비밀번호 파일을 가리키도록 스크립트를 업데이트해야 합니다.

2

명령 참조

- [EPM Automate 명령 실행 정보](#)
- [EPM Automate 실행](#)
- [명령 개요](#)
- [EPM Automate 명령](#)
- [종료 코드](#)

일부 EPM Automate 명령은 모든 비즈니스 프로세스에 적용되고 일부는 비즈니스 프로세스 그룹에 적용됩니다. 달리 지정되지 않은 경우 특정 비즈니스 프로세스(예: Planning)에 적용할 수 있는 명령은 다른 비즈니스 프로세스(예: Financial Consolidation and Close)에서 작동하지 않습니다. 지원하지 않는 비즈니스 프로세스에 대해 명령을 실행하려고 하면 오류가 발생합니다. 각 비즈니스 프로세스에 적용 가능한 명령 목록을 보려면 [개별 Cloud EPM 서비스 관련 명령](#)을 참조하십시오.

EPM Automate 명령 실행 정보

모든 Oracle Fusion Cloud Enterprise Performance Management 서비스에서는 원격 환경 관리에 EPM Automate 명령을 사용합니다.

- [필수 조건](#)
- [기본 파일 위치](#)
- [Transport Layer Security Protocol 사용](#)
- [EPM Automate 명령 사용](#)
- [하나의 매개변수에 여러 값 지정](#)
- [일별 유지관리 중 동작](#)

필수 조건

이 섹션에는 환경의 기본 파일 위치 및 Oracle Fusion Cloud Enterprise Performance Management 인증서 사용 등 EPM Automate를 사용하기 위한 필수 조건이 나와 있습니다.

일반

모든 Cloud EPM 사용자는 해당 ID 도메인 인증서를 사용하여 EPM Automate를 통해 환경에 연결할 수 있습니다. 사용자에게 지정된 사전 정의된 역할 및 애플리케이션 역할에 따라 사용자가 실행할 수 있는 명령이 결정됩니다.

- ID 도메인의 사용자를 추가하거나 삭제하는 명령을 실행하려면 ID 도메인 관리자 역할이 필요합니다.
- 명령을 실행하는 데 필요한 파일은 모두 환경 내에 있어야 합니다. [uploadFile](#) 명령을 사용하여 파일을 업로드합니다.
각 서비스에서 사용되는 기본 파일 위치에 대한 자세한 내용은 [기본 파일 위치](#)를 참조하십시오.
- 명령에서 파일 확장자 사용법:

- 파일 작업을 수행하는 명령을 실행하려면 파일 확장자를 포함한 전체 파일 이름(예: data.csv)을 지정합니다. 파일 작업 명령의 예로 deletefile, listfiles, uploadfile 등이 있습니다.
- 마이그레이션 작업을 수행하는 명령을 실행하려면 파일 확장자를 사용하지 마십시오. 마이그레이션 작업을 수행하려면 스냅샷의 이름을 지정해야 합니다.
- 공백이 포함된 매개변수 값(예: 설명, 위치 이름 및 폴더 경로)은 따옴표로 묶어야 합니다.

Planning

- 작업
다음 섹션에서 설명하는 대부분의 명령에는 작업이 필요합니다. 작업은 데이터 임포트 또는 익스포트와 같은 조치로, 즉시 시작할 수 있거나 나중에 시작되도록 예약할 수 있습니다. 예를 들어 데이터 임포트 또는 익스포트와 데이터베이스 새로고침입니다.
작업(Job) 콘솔을 사용하는 경우 다음 작업(Operation)을 수행하는 데 적합한 작업(Job)을 생성해야 합니다. Planning에서 작업을 생성하는 방법에 대한 자세한 지침은 *Planning 관리*의 작업 관리를 참조하십시오.
 - 애플리케이션으로 데이터 임포트
 - 애플리케이션에서 데이터 익스포트
 - 애플리케이션으로 메타데이터 임포트
 - 애플리케이션에서 메타데이터 익스포트
 - 한 블록 저장영역 데이터베이스에서 집계 저장영역 데이터베이스로 데이터를 복사하거나, 블록 저장영역 데이터베이스에서 다른 블록 저장영역 데이터베이스로 데이터를 복사합니다.
- 비즈니스 규칙
제외할 비즈니스 규칙이 애플리케이션에 있어야 합니다.
Calculation Manager를 사용하여 애플리케이션에 배포되는 비즈니스 규칙을 생성합니다. *Calculation Manager를 사용하여 디자인*을 참조하십시오.

Data Management

- 데이터 규칙
데이터 로드 규칙은 Data Management에서 파일의 데이터를 로드하는 방법을 정의합니다. EPM Automate를 사용하여 데이터를 로드하려면 사전 정의된 데이터 로드 규칙이 있어야 합니다.
- बै치
Data Management에 정의된 बै치를 사용하여 데이터를 로드할 수 있습니다. 사용자는 बै치를 사용하여 여러 로드 규칙을 बै치로 결합하고 직렬 또는 병렬 모드로 실행할 수 있습니다.

기본 파일 위치

기본 업로드 위치

기본적으로 Oracle Fusion Cloud Enterprise Performance Management에 업로드된 모든 파일은 마이그레이션에서 액세스할 수 있는 기본 위치에 저장됩니다.

Migration에서 처리할 파일(예: 서비스로 임포트할 스냅샷)을 기본 위치로 업로드해야 합니다.

인박스 및 아웃박스

인박스 및 아웃박스 위치는 Cloud EPM 비즈니스 프로세스 간에 서로 다를 수 있습니다. 인박스를 사용하여 Profitability and Cost Management 이외의 비즈니스 프로세스를 통해 임포트하거나

다르게 처리할 파일을 업로드합니다. Data Management는 인박스 또는 인박스 내 디렉토리의 파일을 처리할 수 있습니다.

Typically, Cloud EPM은 비즈니스 프로세스를 통해 생성한 파일(예: 데이터 또는 메타데이터 익스포트 파일)을 아웃박스에 저장합니다.

- 이러한 애플리케이션은 EPM Automate에서 파일을 업로드하는 인박스와 다운로드할 파일을 저장하는 아웃박스에 액세스할 수 있습니다. 이러한 애플리케이션에 고유한 프로세스를 사용하여 처리하려는 경우 이 위치에 파일을 업로드해야 합니다. 파일을 아웃박스로 업로드할 수도 있습니다.
 - Planning
 - Planning 모듈
 - Account Reconciliation
 - 재무 통합 및 마감
 - Tax Reporting
 - Narrative Reporting
 - Enterprise Profitability and Cost Management

인박스/아웃박스 탐색기를 사용하여 기본 위치에 저장된 파일을 찾을 수 있습니다. EPM Automate를 사용하여 생성하는 애플리케이션 스냅샷은 인박스/아웃박스 탐색기에 나열되지 않습니다. 마이그레이션 스냅샷 탭에서 볼 수 있습니다.

- Profitability and Cost Management 프로세스를 사용하여 처리할 파일은 profitinbox에 업로드해야 합니다. 파일을 profitoutbox에 업로드할 수도 있습니다. Profitability and Cost Management 프로세스에서 익스포트한 파일은 profitinbox에 저장됩니다. 파일 탐색기를 사용하여 이러한 파일을 찾습니다.
- Data Management를 사용하여 처리할 파일은 인박스 또는 해당 하위 폴더에 있어야 합니다. 기본적으로 데이터 관리를 사용하여 익스포트한 파일은 아웃박스에 저장되고, 데이터 관리 보고서 출력은 데이터 관리 outbox/report 폴더에 저장됩니다. Data Management 파일 브라우저를 사용하여 이러한 파일을 찾습니다.
- Oracle Fusion Cloud Enterprise Data Management는 업로드, 복사 또는 다운로드된 임포트 및 익스포트 파일에 기본 위치를 사용합니다. 기본 위치의 파일은 ListFiles 명령을 사용하여 볼 수 있습니다.

로그 파일

각 EPM Automate 명령 실행은 디버그 파일을 생성하며, 명령이 성공하면 파일이 자동으로 삭제됩니다. 명령 실행 중에 오류가 발생하면 실패한 명령의 디버그 파일이 EPM Automate를 실행하는 디렉토리에서 유지관리됩니다. 기본적으로 Oracle/epm automate/bin 디렉토리 (Windows) 또는 home/user/epmautomate/bin(Linux/UNIX)입니다.

EPM Automate 디버그 파일은 다음과 같은 이름 지정 규칙을 사용합니다.

commandname_date_timestamp.log. 예를 들어 2020년 11월 23일 09:28:02에 실패한 listfiles 명령을 실행할 경우 디버그 파일 이름은 listfiles_23_11_2020_09_28_02.log입니다.

실패한 명령의 디버그 파일 생성은 숨길 수 없습니다. 그러나 다음 Windows 예제와 같이 -d, 디버그 파일 이름, 오류 및 출력 스트림(-d >> c:\logs\LOG_FILE_NAME.log 2>&1)을 명령 끝에 추가하면 디버그 정보 및 명령 출력을 다른 디렉토리에 있는 파일에 쓸 수 있습니다.

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

Transport Layer Security Protocol 사용

EPM Automate는 TLS(Transport Layer Security) 프로토콜 1.2 이상을 지원하는 운영 체제에 설치되어야 합니다.

인증 및 데이터 암호화에 대한 최상위 레벨 보안을 보장하기 위해 EPM Automate는 TLS 1.2 및 1.3만 지원합니다. EPM Automate가 실행된 컴퓨터에서 TLS 1.2 또는 1.3이 사용으로 설정되지 않은 경우 EPMAT-7: Unable to connect. Unsupported Protocol: HTTPS 오류가 표시됩니다. 이 오류를 해결하려면 IT 관리자와 협력하여 TLS 1.2 또는 1.3을 사용으로 설정합니다.

TLS 1.2 및 1.3을 사용으로 설정하는 절차는 운영 체제에 따라 다릅니다. 다음 정보 소스를 사용하십시오. 다른 지원되는 운영 체제의 경우 유사한 웹 리소스를 사용할 수 있습니다.

- Windows 컴퓨터에서 TLS 1.3를 사용하는 방법에 대한 자세한 내용은 [Windows 10에서 TLS 1.3을 사용으로 설정하는 방법](#) 정보를 확인하십시오.
- Red Hat Enterprise Linux에 대해 TLS 1.3을 사용으로 설정하는 방법에 대한 내용은 [Red Hat Enterprise Linux 8의 Transport Layer Security 버전 1.3](#) 정보를 확인하십시오.

EPM Automate 명령 사용

명령 매개변수 순서

명령의 모든 필수 매개변수는 명령 사용에서 확인되는 순서로 전달되어야 합니다. 필수 매개변수 및 해당 값은 임의 순서로 전달할 수 있는 선택적 매개변수 앞에 옵니다. 선택적 매개변수는 위치와 관련이 없습니다.

예를 들어 다음과 같은 login 명령 사용을 고려해 보십시오.

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

이 명령에는 필수 매개변수 3개(USERNAME, PASSWORD, CLOUD_EPM_BASE_URL)가 있으며 사용법에 확인된 순서대로 표시되어야 합니다. 이 순서가 유지되지 않으면 명령은 오류를 반환합니다. 선택적 매개변수인 ProxyServerUserName, ProxyServerPassword, ProxyServerDomain과 해당 값은 임의 순서로 지정할 수 있습니다.

EPM Automate 명령은 대소문자를 구분합니까?

EPM Automate 명령은 대소문자를 구분하지 않습니다. 명령 이름 입력 방법은 명령 실행에 영향을 주지 않습니다. 예를 들어 addUsers 명령을 addusers, ADDUSERS 또는 AdDuSeRs로 입력할 수 있습니다.

EPM Automate 명령 매개변수에서 대소문자 구분 여부

EPM Automate 명령 매개변수는 대소문자를 구분하지 않습니다. 명령 매개변수 이름이 입력된 방식은 명령 실행에 영향을 주지 않습니다. 예를 들어 명령 실행에 영향을 주지 않고 FileName 매개변수를 filename, fileName 또는 fIlEnAmE으로 입력할 수 있습니다.

매개변수 파일을 사용하여 명령 실행을 위한 값 전달

대부분의 EPM Automate 명령에는 런타임 매개변수가 필요합니다. 명령 프롬프트에서 직접 이러한 값을 제공하거나 명령이 사용할 매개변수 파일에 저장할 수 있습니다.

명령을 실행하는 데 필요한 인증서(Oracle Fusion Cloud Enterprise Performance Management 사용자 이름과 암호화된 또는 일반 텍스트 비밀번호)를 비롯한 모든 매개변수 값은 명령 프롬프트에 입력하는 대신 매개변수 파일에 저장할 수 있습니다. 매개변수 파일을 사용하면 명령을 실행하는 동안 재사용성 및 보안이 향상됩니다.

매개변수 파일 정보

매개변수는 명령에 따라 다르기 때문에 각 명령에는 고유한 매개변수 파일이 필요합니다. 예를 들어, [copyFromObjectStorage](#) 명령과 [copyFromSFTP](#) 명령은 서로 다른 매개변수를 사용합니다.

명령 매개변수는 위치와 관련됩니다. 즉, 특정 순서로 제공해야 합니다. 각 명령의 매개변수 파일은 해당 명령의 필수 매개변수에 대한 사용 패턴 및 시퀀스를 따라야 합니다. 선택적 매개변수는 필수 매개변수 뒤에 지정되며 임의 순서로 나열될 수 있습니다.

예를 들어, 암호화된 비밀번호를 사용하여 [login](#) 명령을 실행하기 위한 필수 매개변수의 사용 패턴은 다음과 같습니다.

```
epmautomate login USERNAME PASSWORD_FILE URL
```

login 명령을 실행하는 데 사용하는 매개변수 파일(예: *login_encrypted.txt*)은 이전 사용 패턴에 지정된 정확한 순서로 매개변수 값을 나열해야 합니다.

```
serviceAdmin
C:\mySecuredir\password.epw
https://test-cloud-pln.pbc.us1.oraclecloud.com
```

매개변수 파일이 모든 필수 매개변수를 포함할 필요는 없습니다. 예를 들어, *new_login_encrypted.txt* 파일은 username 매개변수의 값을 포함하지 않습니다.

```
C:\mySecuredir\password.epw
https://test-cloud-pln.pbc.us1.oraclecloud.com
```

username(serviceAdmin) 값은 매개변수 파일에서 생략되므로 명령 프롬프트에 제공해야 합니다.

이러한 파일을 사용하여 login 명령을 실행하는 예를 보려면 [명령에서 매개변수 파일 사용](#)을 참조하십시오.

매개변수 파일 생성

매개변수 파일의 콘텐츠는 파일이 사용되는 명령의 사용 패턴과 일치해야 합니다. 필수 매개변수는 위치와 관련되므로 사용 패턴에 표시된 정확한 순서로 나타나야 합니다. 매개변수 파일을 생성하려면:

1. 텍스트 편집기를 사용하여 새 파일을 생성하고 폴더에 저장합니다.
2. 매개변수 파일을 생성할 명령의 사용 패턴을 복사하여 이전 단계에서 생성한 텍스트 파일에 붙여넣습니다. 예를 들어, [cloneEnvironment](#) 명령에 대한 매개변수 파일을 생성하는 경우 사용 패턴은 다음과 같습니다.

```
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]
[ApplicationProperties=true|false]
```

3. 파일을 편집합니다.

- a. 첫번째 필수 매개변수 앞에 있는 텍스트를 삭제합니다. 이 예에서는 `epmAutomate cloneEnvironment`를 삭제합니다.
- b. 각 필수 매개변수를 사용할 값으로 바꿉니다.
- c. 각 선택적 매개변수를 사용할 값으로 바꿉니다.
- d. 사용하지 않을 선택적 매개변수를 삭제합니다. EPM Automate는 해당 기본값을 사용합니다.
- e. 각 필수 및 선택적 매개변수 값 뒤에 줄 바꿈을 삽입합니다. 예를 들어, 파일의 콘텐츠는 다음과 유사할 수 있습니다.

```
serviceAdmin
Password.epw
https://epm-test-ociarcs.epm.us.region.ocs.oc-test.com
UsersAndPreDefinedRoles=true
ApplicationProperties=false
storedSnapshotsAndFiles=true
DailyMaintenanceStartTime=false
```

4. 매개변수 파일을 저장합니다.

명령에서 매개변수 파일 사용

`-p FILE_NAME` 옵션을 사용하여 매개변수 파일을 명령에 전달합니다.

명령에서 `-p FILE_NAME` 옵션의 위치에 따라 파일 콘텐츠가 사용되는 방식이 결정됩니다. EPM Automate는 `-p FILE_NAME` 옵션이 표시되는 위치에 매개변수 파일의 콘텐츠를 삽입합니다.

모든 필수 값이 포함된 `login_encrypted.txt` 매개변수 파일을 사용하면(이전 섹션 참조) 다음 명령을 실행하여 세션을 시작합니다.

```
epmautomate login -p login_encrypted.txt
```

이전 섹션에서 참조된 `username` 필수 값을 생략하는 `new_login_encrypted.txt` 매개변수 파일을 사용하면 다음 명령을 실행하여 세션을 시작합니다.

```
epmautomate login serviceAdmin -p new_login_encrypted.txt
```

여러 라인의 매개변수 입력

`-p` 옵션을 사용하여 여러 라인 입력을 사용으로 설정하면 명령 매개변수를 여러 라인으로 전달할 수 있습니다. 모든 명령 매개변수를 지정한 후 새 라인에 `+`(더하기 문자)를 입력하여 명령 실행을 시작합니다. 예는 다음과 같습니다.

```
c:\Oracle\EPM Automate\bin>epmautomate login -p -
Enter multiline input (end with +):
serviceAdmin
C:\mySecuredir\password.epw
https://test-cloud-pln.pbcs.us1.oraclecloud.com
+
Login successful

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

① Note

- 여러 라인 입력 모드에서는 매개변수 파일을 사용할 수 없습니다. 콘솔에서는 매개변수를 수동으로 입력해야 합니다.
- 각 매개변수는 별도의 라인에 입력해야 합니다. 여러 라인 입력 모드를 호출한 후 한 라인에 여러 매개변수를 입력하면 EPMAT-7:Invalid or missing parameter 오류가 표시됩니다.

일부 Enterprise Profitability and Cost Management 명령에서 대체 변수 사용

Enterprise Profitability and Cost Management 작업은 작업 매개변수로 대체 변수를 지원합니다. EPM Automate에서 명령 실행의 일부로 이러한 작업을 실행할 때 해결되는 대체 변수를 사용할 수 있습니다.

이러한 명령을 실행하는 데 사용되는 작업은 다음 대체 변수를 지원합니다.

- [calculateModel](#)
- [clearPOV](#)
- [copyPOV](#)
- [deletePOV](#)
- [validateModel](#)

대체 변수는 이러한 명령에서 사용되는 모든 매개변수(필수 및 선택적)에 대해 지원됩니다.

명령으로 대체 변수 지정

명령에서 대체 변수를 사용하려면 &(앰퍼샌드) 문자로 매개변수 값을 시작합니다. 예를 들어, 다음 calculateModel 샘플 명령에서 실제 값 FY22 및 :ALL_RULES 대신 대체 변수 &currYear 및 &execType을 사용할 수 있습니다.

- **Sample command *미사용* substitution variables:**epmautomate calculateModel FY22::Jan::Actual::Working "10 Actuals Allocation Process" ALL_RULES clearCalculatedData=true executeCalculations=true optimizeForReporting=true comment="Running all rules to calculate a POV"
- **대체 변수 사용 샘플 명령:**epmautomate calculateModel &currYear::Jan::Actual::Working "10 Actuals Allocation Process" &execType

```
clearCalculatedData=true executeCalculations=true optimizeForReporting=true
comment="Running all rules to calculate a POV"
```

① Note

- &(앰퍼샌드) 문자로 시작하는 매개변수 값만 대체 변수로 처리됩니다.
- 대체 변수는 Enterprise Profitability and Cost Management에 있어야 합니다. 존재하지 않거나 철자가 잘못된 대체 변수를 전달할 경우 명령이 실패합니다.

하나의 매개변수에 여러 값 지정

일부 EPM Automate 명령에서는 여러 개의 심표로 구분된 매개변수 값이 허용됩니다. Planning 애플리케이션의 비즈니스 규칙, 규칙 세트 및 템플릿에 있는 멤버 유형의 실행 시간 프롬프트를 예로 들 수 있습니다.

EPM Automate 명령에서 이름이 Entities인 멤버 유형의 실행 시간 프롬프트에 대해 두 개 이상의 멤버를 설정하려면 runbusinessrule 명령 실행에 대한 다음 예에 설명된 것처럼 ,(심표)를 사용합니다.

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

공백 및 심표와 같은 특수 문자가 포함된 멤버 이름은 다음 예에 표시된 것처럼 큰따옴표로 묶고 \ (백슬래시)로 이스케이프되어야 합니다.

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

일별 유지관리 중 동작

환경에 대한 일별 유지관리가 진행 중인 동안에는 EPM Automate 명령을 실행하지 마십시오.

일별 유지관리 중에는 사용자 활동이 허용되지 않습니다. 일별 유지관리가 진행 중인 동안 직접 또는 스크립트를 사용하여 EPM Automate 명령을 실행하려고 시도하면 다음 오류가 표시됩니다.

```
EPMAT-11:Internal server error. Due to the daily maintenance, your
Oracle EPM Cloud Service environment is currently unavailable.
```

EPM Automate 실행

Oracle Fusion Cloud Enterprise Performance Management 인증서를 사용하여 EPM Automate를 통해 로그인합니다. SSO 인증서를 사용하여 로그인할 수는 없습니다.

모든 Cloud EPM 사용자는 해당 ID 도메인 인증서를 사용하여 EPM Automate를 통해 환경에 연결할 수 있습니다. 사용자에게 지정된 사전 정의된 역할 및 애플리케이션 역할에 따라 사용자가 실행할 수 있는 명령이 결정됩니다.

또한 일부 명령은 서비스 관리자만 실행할 수 있으며, 일부 명령은 실행하는 데 ID 도메인 관리자 역할이 필요할 수도 있습니다.

디버그 로그 파일 생성

오라클 고객지원센터에서는 EPM Automate 실행 중 발생한 문제를 해결하기 위해 해당 세션의 디버그 로그 파일을 요청합니다. 디버그 메시지를 생성하는 -d 옵션을 지원합니다. 그러면 >

지시어를 사용하여 파일로 메시지를 리디렉션할 수 있습니다. 하나의 명령에 대해 디버그 파일을 생성하거나 여러 명령이 포함된 배치 실행 파일 또는 스크립트를 생성할 수 있습니다.

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

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

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

Windows

EPM Automate를 실행하기 전에 실행 중인 컴퓨터에서 환경에 액세스할 수 있는지 확인합니다.

EPM Automate는 사용자 정보가 포함된 `.prefs` 파일 및 로그 파일을 현재 디렉토리에 생성합니다. Windows 컴퓨터에서는 `.prefs` 파일을 생성한 사용자 및 Windows 관리자에게만 이 파일의 콘텐츠가 표시됩니다. Linux, UNIX, macOSX 환경에서는 권한 600으로 `.prefs` 파일이 생성되므로 소유자에게만 이 파일에 대한 읽기 및 쓰기가 허용됩니다.

사용자가 명령을 실행하는 Windows 디렉토리에서 쓰기 권한이 없는 경우 EPM Automate는 Windows 환경에서 `FileNotFoundException: .prefs`(액세스가 거부됨) 오류를 표시합니다. 이 오류를 해결하려면 현재 사용자의 Windows 계정에 EPM Automate가 실행되는 디렉토리에 대한 읽기/쓰기 액세스 권한이 있어야 합니다. 또한 이 사용자는 파일에 액세스하거나(예: `uploadFile` 명령을 실행하는 경우) 파일을 쓰는(예: `downloadFile` 명령을 실행하는 경우) 다른 디렉토리에서도 적절한 권한이 있어야 합니다.

① 주

이름에 `&`가 포함된 폴더(예: `C:\Oracle\A&B`)에서는 EPM Automate를 실행할 수 없습니다.

Windows 클라이언트에서 EPM Automate를 실행하려면 다음을 수행합니다.

- 실행, 모든 프로그램, EPM Automate, EPM Automate 실행** 순으로 누릅니다. EPM Automate 명령 프롬프트가 표시됩니다.
- 선택사항:** 작업을 수행하려는 디렉토리로 이동합니다.
- 선택 사항:** 비밀번호 암호화 파일을 생성합니다. 암호화된 비밀번호를 전달하여 세션을 시작하도록 암호화 파일을 사용합니다.

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

- 서비스 관리자로 세션을 시작합니다. 다음과 같은 명령을 사용합니다.

- 암호화되지 않은 비밀번호 사용:

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

- 암호화된 비밀번호 사용:

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

- 완료하려는 태스크를 실행할 명령을 입력합니다.

명령 실행 상태에 대한 자세한 내용은 [종료 코드](#)를 참조하십시오.

6. 환경에서 사인아웃합니다. 다음 명령을 사용합니다.

```
epmautomate logout
```

Linux

① 주

`JAVA_HOME`이 `.profile` 파일의 `PATH` 변수에 설정되어 있거나 쉘 환경 변수로 설정되어 있는지 확인합니다. 지원되는 JRE(버전 17)가 필요합니다.

Linux 클라이언트에서 EPM Automate를 실행하려면 다음을 수행합니다.

1. 터미널 창을 열고 EPM Automate를 설치한 디렉토리로 이동합니다.
2. **선택 사항:** 비밀번호 암호화 파일을 생성합니다. 암호화되지 않은 비밀번호가 아니라 암호화된 비밀번호를 전달하여 세션을 시작하기 위해 암호화 파일을 사용합니다.

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

3. 서비스 관리자로 세션을 시작합니다. 다음과 같은 명령을 사용합니다.

- 암호화되지 않은 비밀번호 사용:

```
./bin/epmautomate.sh login serviceAdmin P@ssword1  
https://test-cloudpln.pbcs.us1.oraclecloud.com
```

- 암호화된 비밀번호 사용:

```
./bin/epmautomate.sh login serviceAdmin ../misc/encrypt/password.epw  
https://test-cloudpln.pbcs.us1.oraclecloud.com
```

4. 완료하려는 태스크를 실행할 명령을 입력합니다.

명령 실행 상태에 대한 자세한 내용은 [종료 코드](#)를 참조하십시오.

5. 환경에서 사인아웃합니다. 다음 명령을 사용합니다.

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

EPM Automate의 여러 인스턴스 실행

EPM Automate의 여러 인스턴스를 동일한 디렉토리의 한 환경에 대해 실행할 수 있습니다. 마찬가지로, 동일한 디렉토리 또는 다른 디렉토리의 서로 다른 환경에 대해 여러 인스턴스를 실행할 수 있습니다.

예를 들어 Planning 애플리케이션 큐브를 <https://cloudpln.pbcs.us1.oraclecloud.com> 및 <https://testcloudpln.pbcs.us1.oraclecloud.com>에서 동시에 새로고쳐야 할 수 있습니다. 이 시나리오에는 두 가지 옵션이 있습니다.

- 동일한 디렉토리에서 두 개의 EPM Automate 인스턴스를 실행하여 서로 다른 환경에서 애플리케이션 큐브를 새로고칩니다.

- 별도 디렉토리에서 EPM Automate를 실행하여 환경에 연결하고 애플리케이션 큐브를 새로고칩니다.

두 시나리오 모두 각 인스턴스는 독립적으로 작동합니다. 한 인스턴스에서 로그아웃하더라도 다른 인스턴스에서는 로그아웃되지 않습니다. EPM Automate를 사용하여 시작된 활동은 다른 인스턴스에서 로그아웃하더라도 해당 환경에서 완료될 때까지 계속 실행됩니다.

이 섹션에는 두 개의 EPM Automate 세션을 생성하여 태스크를 수행하는 데 사용할 수 있는 Windows 및 Unix/Linux 샘플 스크립트(caller 및 multisession)가 있습니다. 여러 개의 동시 세션을 실행하려면 multisession 스크립트를 호출하여 login, uploadfile, listfiles 및 logout 명령을 실행하는 caller 스크립트에 다음 연결 정보를 추가해야 합니다. 다른 태스크를 수행하도록 multisession 스크립트를 수정할 수 있습니다. 이러한 스크립트는 둘 다 동일한 디렉토리에 저장되어 있어야 합니다.

- EPM Automate에서는 EPM_SID 환경 변수를 사용하여 여러 세션을 구별합니다. caller 스크립트에서 이 변수를 각 세션에 고유한 값으로 설정해야 합니다. 샘플 스크립트에서는 다음과 같이 고유한 값으로 설정됩니다.
 - caller.BAT에서 EPM_SID는 고유한 시스템 생성 번호를 지정하는 !RANDOM!으로 설정됩니다. 이 번호는 각 세션의 로그 파일을 생성하는 데에도 사용됩니다. 각 세션의 로그 파일을 추적하려는 경우 !RANDOM! 대신 고유 번호를 지정할 수 있습니다.
 - caller.sh에서 EPM_SID는 고유한 프로세스 ID로 설정됩니다. 각 세션의 로그 파일을 추적하려는 경우 전달된 값을 사용하도록 multisession 스크립트의 export EPM_SID=\$\$ 문을 수정하여 고유한 EPM_SID를 지정한 다음, 각 세션의 caller 스크립트에서 이 매개 변수에 대해 고유한 값을 전달합니다. 예를 들어 caller.sh 스크립트에서 EPM_SID의 값을 다음과 같이 지정합니다.

```
$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: 서비스 관리자의 로그인 ID
- PASSWORD: 서비스 관리자의 비밀번호
- URL: 환경의 연결 URL

샘플 Windows 스크립트

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
```

샘플 Bourne 셸 스크립트

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

```

명령 개요

모든 EPM Automate 명령의 알파벳순 목록입니다.

표 2-1 모든 EPM Automate 명령

명령 이름	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
addUsers	✓	✓	✓	✓	✓	✓	✓	✓
addUsersToGroup	✓	✓	✓	✓	✓	✓	✓	✓
addUsersToTeam		✓	✓			✓		
addUserToGroups	✓	✓	✓	✓	✓	✓	✓	✓
applicationAdminMode	✓	✓	✓		✓			
applyDataGrants				✓				
archiveTmTransactions						✓		
assignRole	✓	✓	✓	✓	✓	✓	✓	✓
autoPredict * 각주 참조	✓							
calculateModel					✓			
clearCube	✓				✓			
clearDataByPointOfView					✓			
clearDataByProfile		✓	✓					
clearPOV				✓				
cloneEnvironment	✓	✓	✓	✓	✓	✓	✓	✓
compactCube	✓				✓			
copyDataByPointOfView					✓			
copyDataByProfile		✓	✓					
copyFileFromInstance	✓	✓	✓	✓	✓	✓	✓	✓
copyFromObjectStorage	✓	✓	✓	✓	✓	✓	✓	✓
copyFromSFTP	✓	✓	✓	✓	✓	✓	✓	✓
copyOwnershipDataToNextYear		✓	✓					
copyPOV				✓				
copySnapshotFromInstance	✓	✓	✓	✓	✓	✓	✓	
copyToObjectStorage	✓	✓	✓	✓	✓	✓	✓	✓
copyToSFTP	✓	✓	✓	✓	✓	✓	✓	✓
createGroups	✓	✓	✓	✓	✓	✓	✓	✓
createNRSnapshot								✓
createReconciliations						✓		
deleteFile	✓	✓	✓	✓	✓	✓	✓	✓
deleteGroups	✓	✓	✓	✓	✓	✓	✓	✓
deletePointOfView					✓			
deletePOV				✓				

표 2-1 (계속) 모든 EPM Automate 명령

명령 이름	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
deployCube				✓				
deployEJTemplates		✓						
deployFormTemplates		✓	✓					
deployTaskManagerTemplate		✓			✓			
dismissIPMInsights**	✓							
downloadFile	✓	✓	✓	✓	✓	✓	✓	✓
enableApp				✓				
enableQueryTracking	✓				✓			
encrypt	✓	✓	✓	✓	✓	✓	✓	✓
essbaseBlockAnalysisReport	✓	✓	✓					
executeAggregationProcess	✓				✓			
executeBurstDefinition								✓
executeReportBurstingDefinition	✓	✓	✓		✓			
exportAccessControl						✓		
exportAppAudit	✓	✓	✓		✓			
exportAppSecurity	✓	✓	✓		✓			
exportARApplicationProperties						✓		
exportBackgroundImage						✓		
exportCellLevelSecurity	✓		✓		✓			
exportData	✓	✓	✓		✓			
exportConsolidationJournals		✓						
exportDataManagement	✓	✓	✓	✓	✓			
exportDimension							✓	
exportDimensionMapping							✓	
exportEJJournals		✓						
exportEssbaseData	✓	✓	✓		✓			
exportJobConsole	✓	✓	✓		✓			
exportLibraryArtifact								✓
exportLibraryDocument	✓	✓	✓		✓			
exportLogoImage						✓		
exportMapping	✓	✓	✓	✓	✓	✓		
exportMetadata	✓	✓	✓		✓			
exportOwnershipData		✓	✓					
exportQueryResults				✓				
exportSnapshot	✓	✓	✓	✓	✓	✓	✓	
exportTemplate				✓				
exportTaskManagerAccessControl		✓	✓		✓			
exportValidIntersections	✓	✓	✓		✓			
extractDimension							✓	
extractPackage							✓	

표 2-1 (계속) 모든 EPM Automate 명령

명령 이름	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
feedback	✓	✓	✓	✓	✓	✓	✓	✓
getApplicationAdminMode	✓	✓	✓		✓	✓		
getDailyMaintenanceStartTime	✓	✓	✓	✓	✓	✓	✓	✓
getEssbaseQryGovExecTime	✓	✓	✓	✓	✓			
getIdleSessionTimeout	✓	✓	✓	✓	✓	✓	✓	✓
getIPAllowlist(사용되지 않음)	✓	✓	✓	✓	✓	✓	✓	✓
getRestrictedDataAccess	✓	✓	✓	✓	✓	✓	✓	✓
getSubstVar	✓	✓	✓		✓			
getVirusScanOnFileUploads	✓	✓	✓	✓	✓	✓	✓	✓
groupAssignmentAuditReport	✓	✓	✓	✓	✓	✓	✓	✓
help	✓	✓	✓	✓	✓	✓	✓	✓
importAppAudit	✓				✓			
importAppSecurity	✓	✓	✓		✓			
importARApplicationProperties						✓		
importBackgroundImage						✓		
importBalances						✓		
importCellLevelSecurity	✓		✓		✓			
importConsolidationJournals		✓						
importData	✓	✓	✓	✓	✓			
importDataManagement	✓	✓	✓	✓	✓			
importDimension							✓	
importEJDimensionMembers		✓						
importJobConsole	✓	✓	✓		✓			
importLibraryArtifact								✓
importLibraryDocument	✓	✓	✓		✓			
importLogoImage						✓		
importMapping	✓	✓	✓	✓	✓	✓		
importMetadata	✓	✓	✓		✓			
importOwnershipData		✓	✓					
importPreMappedBalances						✓		
importPreMappedTransactions						✓		
importProfiles						✓		
importRates						✓		
importRCAttributeValues						✓		
importReconciliationAttributes						✓		
importSDMDimensionMembers		✓	✓					
importSnapshot	✓	✓	✓	✓	✓	✓	✓	
importSupplementalCollectionData		✓	✓					
importSupplementalData		✓	✓					
importTemplate				✓				

표 2-1 (계속) 모든 EPM Automate 명령

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

표 2-1 (계속) 모든 EPM Automate 명령

명령 이름	PLN, SWP, SP, FF	FCC	TR	PCM	EPCM	AR	EDM	NR
runDailyMaintenance	✓	✓	✓	✓	✓	✓	✓	✓
runDataRule	✓	✓	✓	✓	✓	✓		
runDMReport	✓	✓	✓	✓	✓	✓		
runIntegration	✓	✓	✓	✓	✓	✓		
runIntercompanyMatchingReport		✓						
runMatchingReport						✓		
runPipeline	✓	✓	✓		✓	✓		
runPlanTypeMap	✓							
runRuleSet	✓	✓	✓		✓			
runSupplementalDataReport		✓	✓					
runTaskManagerReport		✓	✓		✓			
sendMail	✓	✓	✓	✓	✓	✓	✓	✓
setApplicationAdminMode	✓	✓	✓		✓	✓		
setDailyMaintenanceStartTime	✓	✓	✓	✓	✓	✓	✓	✓
setDemoDates		✓	✓		✓	✓		
setEJJournalStatus		✓						
setEncryptionKey	✓	✓	✓	✓	✓	✓	✓	✓
setEssbaseQryGovExecTime	✓	✓	✓	✓	✓			
setIdleSessionTimeout	✓	✓	✓	✓	✓	✓	✓	✓
setIPAllowlist(사용되지 않음)	✓	✓	✓	✓	✓	✓	✓	✓
setManualDataAccess	✓	✓	✓	✓	✓	✓	✓	✓
setPeriodStatus						✓		
setRestrictedDataAccess	✓	✓	✓	✓	✓	✓	✓	✓
setSubstVars	✓	✓	✓		✓			
setVirusScanOnFileUploads	✓	✓	✓	✓	✓	✓	✓	✓
simulateConcurrentUsage	✓	✓	✓					
skipUpdate	✓	✓	✓	✓	✓	✓	✓	✓
snapshotCompareReport	✓	✓	✓		✓			
sortMember	✓				✓			
unassignRole	✓	✓	✓	✓	✓	✓	✓	✓
updateGuidedLearningSettings	✓	✓	✓		✓			
updateUsers	✓	✓	✓	✓	✓	✓	✓	✓
upgrade	✓	✓	✓	✓	✓	✓	✓	✓
uploadFile	✓	✓	✓	✓	✓	✓	✓	✓
userAuditReport(사용되지 않음)	✓	✓	✓	✓	✓	✓	✓	✓
userGroupReport	✓	✓	✓	✓	✓	✓	✓	✓
userLoginReport	✓	✓	✓	✓	✓	✓	✓	✓
validateConsolidationMetadata		✓						
validateModel					✓			

- * 이 명령은 애플리케이션에서 하이브리드 Oracle Essbase 큐브가 사용으로 설정된 경우에만 지원됩니다. 전략적 Workforce Planning 및 Sales Planning은 하이브리드 Essbase를 지원하지 않습니다. 무형식에는 이 명령이 지원되지 않습니다.
- ** 무형식에는 이 명령이 지원되지 않습니다.

약어

- PLN: Planning(Planning 모듈 포함)
- FF: 무형식
- SWP: 전략적 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 Fusion Cloud Enterprise Data Management
- NR: Narrative Reporting

EPM Automate 명령

이 섹션에서는 각 EPM Automate 명령에 대해 자세히 설명합니다. 각 명령에 대해 제공되는 정보에는 명령을 사용할 수 있는 서비스, 명령 사용법 및 예제가 포함됩니다.

addUsers

환경에 업로드된 ANSI 또는 UTF-8 인코딩된 CSV(Comma Separated Value) 파일을 사용하여 ID 도메인에 사용자 बै치를 생성합니다. 사용자 이름 및 임시 비밀번호도 새 사용자에게 알려 줍니다.

서비스 관리자는 [uploadFile](#) 명령을 사용하여 파일을 환경에 업로드합니다. CSV 파일의 모든 열은 필수입니다. 이 명령은 정의의 각 열에 있는 값을 검증하고, 누락되었거나 부적합한 각각의 값을 확인할 수 있는 오류 메시지를 표시합니다. CSV 파일 형식은 다음과 같습니다.

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

CSV 파일 형식에 대한 자세한 설명은 *Oracle Cloud 시작하기*에서 사용자 계정 बै치 임포트를 참조하십시오.

임포트 파일에서 지정된 User Login 값은 대소문자를 구분하지 않습니다. 예를 들어 John.doe@example.com 값은 John.Doe@example.com 또는 이 값의 모든 대소문자 변형과 동일한 것으로 처리됩니다.

CSV 파일의 사용자 정의가 ID 도메인에 있는 사용자 계정과 일치하면 기존 사용자 계정이 변경되지 않습니다. 이 명령은 계정 정보가 파일에 포함된 새 사용자에게 대해서만 계정을 생성합니다. 사용자 계정은 ID 도메인이 지원하는 모든 환경에 공통적으로 적용되므로 ID 도메인을 공유하는 모든 환경에서 새 사용자를 사용할 수 있습니다.

완료되면 명령이 실패한 각 항목에 대한 정보를 콘솔에 인쇄합니다. 이 정보를 검토하여 CSV 파일의 일부 항목에서 명령 실행이 실패한 이유를 알아봅니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

ID 도메인 관리자 및 사전 정의된 역할

사용법

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

설명:

- *FILE_NAME*은 사용자 정보를 포함하는 CSV 파일의 이름입니다. 멀티바이트 문자가 포함된 입력 파일은 UTF-8 문자 인코딩을 사용해야 합니다.
- *userPassword*는 선택사항으로, ID 도메인에 생성된 모든 새 사용자의 기본 비밀번호를 나타냅니다. 지정되는 경우, 이 비밀번호는 최소 ID 도메인 비밀번호 요구사항을 충족해야 합니다. 매개변수가 지정되지 않는 경우, 각 사용자에게 고유한 임시 비밀번호가 지정됩니다. 지정되는 경우, *userPassword* 매개변수 값이 CSV 파일에 지정된 모든 사용자의 비밀번호로 사용됩니다. 사용자를 테스트 목적으로만 생성하는 경우 모든 사용자에게 동일한 비밀번호를 지정하는 것이 좋습니다. 실제 Oracle Fusion Cloud Enterprise Performance Management 사용자를 생성하고 있으며 각 사용자에게 특정 비밀번호를 지정하려는 경우, *userPassword* 선택적 매개변수에 값을 지정하지 않고 이 명령을 사용합니다.
- *resetPassword*는 선택사항으로, 새 사용자가 처음 로그인 시 비밀번호를 변경해야 하는지 여부를 나타냅니다. 기본값은 true입니다. 이 매개변수가 false로 설정되지 않은 경우 새 사용자가 처음 로그인 시 비밀번호를 변경해야 합니다.
*resetPassword*가 true로 설정된 경우 이 명령은 각각의 새 사용자에게 계정에 대한 세부정보 (사용자 이름 및 비밀번호)가 포함된 전자메일을 보냅니다. *resetPassword*가 false로 설정되면 전자메일을 보내지 않습니다. *resetPassword*를 false로 설정하는 경우 *userPassword*를 지정해야 합니다. 지정하지 않으면 각 사용자에게 고유 임시 비밀번호가 지정되지만 전자메일을 보내지 않으므로 사용자가 비밀번호를 몰라서 로그인할 수 없습니다.

예

- 동일한 비밀번호를 사용하여 ID 도메인에 테스트 사용자를 추가하고 비밀번호를 변경하도록 요구하지 않음:

```
epmautomate addUsers user_file.CSV userPassword=Example@Pwd12  
resetPassword=false
```
- 임시 비밀번호를 사용하여 사용자를 ID 도메인에 추가하고 사용자에게 비밀번호를 변경하도록 요구함:

```
epmautomate addUsers user_file.CSV
```

addUsersToGroup

환경에 업로드된 ANSI 또는 UTF-8 인코딩 CSV 파일을 사용하여 액세스 제어의 기존 그룹에 사용자 배치를 추가합니다.

서비스 관리자는 `uploadFile` 명령을 사용하여 파일을 환경에 업로드합니다. 사용자 로그인 값은 대소문자를 구분하지 않습니다. 파일 형식은 다음과 같습니다.

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

① 주

다음 두 조건이 모두 충족되는 경우에만 사용자가 그룹에 추가됩니다.

- 파일에 포함된 사용자 로그인 값이 환경을 서비스하는 ID 도메인에 있습니다. 사용자 로그인 값은 대소문자를 구분하지 않습니다.
- 사용자가 ID 도메인에서 사전 정의된 역할에 지정되었습니다.

완료되면 명령이 실패한 각 항목에 대한 정보를 콘솔에 인쇄합니다. 이 정보를 검토하여 CSV 파일의 일부 항목에서 명령 실행이 실패한 이유를 알아봅니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Enterprise Profitability and Cost Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할

사용법

`epmautomate addUsersToGroup FILE_NAME GROUP_NAME` 설명:

- `FILE_NAME`은 액세스 제어에서 그룹에 지정할 사용자의 로그인 이름을 포함하는 CSV 파일의 이름입니다.
- `GROUP_NAME`은 액세스 제어에 있는 그룹의 이름입니다. 이 값은 대소문자를 구분하지 않습니다.

예

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

addUsersToTeam

CSV 파일에 나열된 Oracle Fusion Cloud Enterprise Performance Management 사용자를 기존 팀에 추가합니다.

CSV 파일에 포함된 사용자가 이미 팀 멤버인 경우 이 명령은 해당 사용자를 무시합니다. 이 파일의 값은 대소문자를 구분하지 않습니다. 서비스 관리자는 [uploadFile](#) 명령을 사용하여 이 파일을 환경에 로드합니다. CSV 파일 형식은 다음과 같습니다.

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

① 주

기본 사용자는 기본적으로 팀에 지정된 태스크를 수행하도록 지정됩니다.

적용 대상

Financial Consolidation and Close, Tax Reporting, Account Reconciliation

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 팀 - 애플리케이션 역할 관리
- 사전 정의된 역할 및 사용자 - 애플리케이션 역할 관리

사용법

`epmautomate addUsersToTeam FILE TEAM_NAME` 설명:

- `FILE`은 팀에 추가할 사용자의 로그인 ID가 나열된 UTF-8 형식 CSV 파일을 확인합니다. 이 명령을 실행하기 전에 [uploadFile](#) 명령을 사용하여 파일을 환경으로 업로드합니다.
- `TEAM_NAME`은 액세스 제어에 정의된 팀 이름을 확인합니다. 이 값은 대소문자를 구분하지 않습니다.

예

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

addUserToGroups

사용자를 ANSI 또는 UTF-8 인코딩된 CSV 파일에서 확인된 액세스 제어 그룹의 멤버로 추가합니다.

서비스 관리자는 [uploadFile](#) 명령을 사용하여 파일을 환경에 업로드합니다. 파일 형식은 다음과 같습니다.

```
Group Name
Group1
Group2
```

그룹 이름 값은 대소문자를 구분하지 않습니다.

완료되면 명령이 실패한 각 항목에 대한 정보를 콘솔에 인쇄합니다. 이 정보를 검토하여 CSV 파일의 일부 항목에서 명령 실행이 실패한 이유를 알아봅니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할

사용법

`epmautomate addUserToGroups FILE_NAME User_Login`, 설명:

- `FILE_NAME`은 사용자를 지정하려는 액세스 제어 그룹 이름이 포함된 CSV 파일의 이름입니다.
- `User_Login`은 액세스 제어 그룹에 지정할 Oracle Fusion Cloud Enterprise Performance Management 사용자의 로그인 ID입니다. 이 사용자 로그인 ID는 대소문자를 구분하지 않으며, 환경을 서비스하는 ID 도메인에 있어야 하고 사전 정의된 역할에 지정되어야 합니다.

예

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

applicationAdminMode

애플리케이션에 대한 액세스가 서비스 관리자만으로 제한되도록 애플리케이션을 관리 모드로 전환합니다.

이 명령은 서비스 관리자가 관리 작업을 수행할 때 사용자가 애플리케이션에서 작업하지 못하도록 하는 데 유용합니다. 사용자가 모드를 다시 변경하여 모든 사용자가 액세스할 수 있도록 할 때까지 애플리케이션은 관리 모드로 유지됩니다.

① 주

이 명령은 더 이상 사용되지 않지만 EPM Automate에서 제거되지는 않았습니다. 대신 [setApplicationAdminMode](#) 명령을 사용하는 것이 좋습니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

`epmautomate applicationAdminMode VALUE` 설명: `VALUE`는 애플리케이션을 관리 모드로 전환할지 여부를 지정합니다. 허용되는 값은 다음과 같습니다.

- `true` - 애플리케이션을 관리 모드로 전환합니다.

- `false` - 모든 사용자가 액세스할 수 있도록 애플리케이션을 일반 모드로 되돌립니다.

예

- 애플리케이션을 관리 모드로 전환: `epmautomate applicationAdminMode true`
- 애플리케이션을 일반 운영으로 되돌리기: `epmautomate applicationAdminMode false`

applyDataGrants

Profitability and Cost Management 애플리케이션에서 정의된 데이터 권한부여와 일치하도록 Oracle Essbase 데이터 슬라이스에 대한 액세스를 제어하는 데이터 권한부여를 새로고칩니다.

Profitability and Cost Management 애플리케이션에서 지정한 사용자 및 그룹 레벨 데이터 권한부여는 Essbase에서 자동으로 동기화됩니다. 애플리케이션의 데이터 권한부여와 Essbase의 필터 간에 불일치가 의심되는 경우 이 명령을 사용하여 Essbase 데이터에 대한 액세스 권한을 동기화합니다.

이 작업을 완료하는 데 필요한 시간은 애플리케이션의 크기에 따라 다릅니다. 다음 유지 관리 기간에 애플리케이션이 백업되기 전에 데이터 권한부여 새로고침 작업이 완료되었는지 확인합니다. 이 작업을 진행하는 동안 애플리케이션을 사용하면 안 되기 때문에 사용자가 애플리케이션을 사용하지 않는 시간에 이 작업을 예약하는 것이 좋습니다.

적용 대상

Profitability and Cost Management,

필수 역할

서비스 관리자, 고급 사용자

사용법

`epmautomate applyDataGrants APPLICATION_NAME` 설명: `APPLICATION_NAME`은 데이터 슬라이스 세트를 재생성할 Profitability and Cost Management 애플리케이션의 이름입니다.

예

```
epmautomate applyDataGrants BksML12
```

archiveTmTransactions

지정된 기간의 일수보다 오래되거나 일수가 같은 일치 트랜잭션을 지원 및 조정 세부정보와 함께 아카이브합니다. 일치 트랜잭션은 ZIP 파일로 기록됩니다.

이 명령을 사용하여 조직의 트랜잭션 보존 정책에 따라 오래된 일치 트랜잭션을 아카이브한 후 제거하면 Account Reconciliation 애플리케이션 크기를 최적화된 상태로 유지할 수 있습니다.

적용 대상

Account Reconciliation

필수 역할

서비스 관리자, 고급 사용자, 사용자, 조회자
사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

```
epmautomate archiveTmTransactions matchType age [filterOperator=VALUE]
[filterValue=VALUE] [logFilename=FILE_NAME] [filename=FILE_NAME] 설명:
```

- matchType은 일치 트랜잭션을 아카이브할 일치 유형의 ID(TextID)입니다.
- age는 트랜잭션이 일치된 이후의 일수를 나타냅니다. 이 값의 일수보다 오래되거나 일수가 같은 일치 트랜잭션은 아카이브됩니다.
- filterOperator는 선택사항으로, 아카이브할 일치 트랜잭션이 포함된 계정을 확인하는 다음 필터 조건 중 하나입니다. 이 값과 filterValue 값을 조합하여 일치 트랜잭션을 아카이브할 계정을 확인합니다.
 - 같음
 - not_equals
 - starts_with
 - ends_with
 - contains
 - not_contains
- filterValue는 선택사항으로, 아카이브할 트랜잭션을 확인하는 필터 값입니다. filterOperator가 equals 또는 not_equals인 경우 공백으로 구분된 목록을 사용하여 여러 값을 지정할 수 있습니다(예: filterValue=101-120 filterValue=102-202). 여러 값이 지정되면 필터 연산자 및 필터 값 조합과 일치하는 계정의 트랜잭션이 아카이브용으로 선택됩니다.

① Note

filterOperator 및 filterValue가 지정되지 않으면 age의 일수보다 오래되거나 일수가 같은 모든 일치 트랜잭션이 matchType이 지정된 모든 계정에서 아카이브됩니다.

- logFilename은 선택사항으로, 명령 활동에 대한 정보를 기록할 로그 파일의 이름입니다. 파일 이름을 지정하지 않으면 이름이 Archive_Transactions_matchType_JOBID.log인 로그 파일이 자동으로 생성됩니다.
- filename은 선택사항으로, 아카이브된 트랜잭션이 포함되는 .ZIP 파일의 이름입니다. 지정되지 않으면 명령은 기본적으로 Archived_Transactions_matchType_JOBID.zip을 생성합니다. [downloadFile](#) 명령을 사용하여 이 파일을 로컬 컴퓨터로 다운로드할 수 있습니다.

① Note

이 명령은 지정된 매개변수를 사용하여 TM 트랜잭션 아카이브 작업을 실행합니다. 작업 ID가 명령 출력에 반환되므로 [purgeArchivedTmTransactions](#) 명령에서 편리하게 사용할 수 있습니다. 작업 콘솔에서 작업을 모니터링할 수 있습니다.

예

- 필터를 사용하지 않고 사용자정의 로그 및 .ZIP 파일 이름을 사용하여 오래된 일치 트랜잭션 아카이브:


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

- 필터를 사용하여 오래된 일치 트랜잭션 아카이브:
 - `epmautomate archiveTmTransactions cashrecon 180 filterOperator>equals filterValue=101-120 FilterValue=102-202`
 - `epmautomate archiveTmTransactions cashrecon 180 filterOperator=contains filterValue=11`

assignRole

ANSI 또는 UTF-8 인코딩 CSV 파일에 로그인 ID가 포함된 사용자(이 명령을 실행하는 사용자 포함)에게 역할을 지정합니다. 이 명령을 사용하여 사용자를 사전 정의된 역할 또는 애플리케이션 역할에 지정합니다.

이 명령을 사용하기 전에 서비스 관리자는 [uploadFile](#) 명령을 사용하여 입력 파일을 환경에 업로드합니다. 파일 형식은 다음과 같습니다.

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

Oracle Cloud 시작하기에서 하나의 역할을 여러 사용자에게 지정을 참조하십시오.

① 주

- 파일에 포함된 사용자 로그인 값은 대소문자를 구분하지 않습니다.
- 큰따옴표를 사용하여 공백 문자를 포함하는 역할 이름을 묶습니다.

완료되면 명령이 실패한 각 항목에 대한 정보를 콘솔에 인쇄합니다. 이 정보를 검토하여 CSV 파일의 일부 항목에서 명령 실행이 실패한 이유를 알아봅니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

사전 정의된 역할을 지정하는 경우:

- 사전 정의된 역할을 지정하는 경우:
 - 서비스 관리자
 - ID 도메인 관리자 및 사전 정의된 역할
- 애플리케이션 역할을 지정하려면 다음을 수행합니다.
 - 서비스 관리자
 - 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할

사용법

`epmautomate assignRole FILE_NAME ROLE 설명:`

- `FILE_NAME`은 사용자 로그인 ID가 포함된 CSV 파일의 이름입니다. CSV 확장자는 소문자로 지정하십시오.
- `ROLE`은 다음 중 하나입니다. 이 값은 대소문자를 구분하지 않습니다.
 - 사전 정의된 ID 도메인 역할에 사용자를 지정하는 경우 `ROLE`은 서비스에 적용 가능한 사전 정의된 역할을 확인해야 합니다. *관리자용 시작 가이드*에서 사전 정의된 역할 이해를 참조하십시오.
이러한 역할에 대한 설명은 *액세스 제어 관리*에서 애플리케이션 레벨에서 역할 지정 관리를 참조하십시오
 - 애플리케이션 역할에 사용자를 지정하는 경우 `ROLE`은 현재 환경에서 애플리케이션에 속하는 역할을 확인해야 합니다. 애플리케이션 역할은 액세스 제어의 **역할** 탭에 나열됩니다. 각 비즈니스 프로세스의 애플리케이션 역할에 대한 설명은 *Oracle Enterprise Performance Management Cloud용 Access Control 관리*에서 다음 항목을 참조하십시오.
 - * Account Reconciliation
 - * Enterprise Profitability and Cost Management
 - * Planning, FreeForm, Financial Consolidation and Close 및 Tax Reporting
 - * Profitability and Cost Management,
 - * Oracle Enterprise Data Management
 - * Narrative Reporting

예

- 사용자를 사전 정의된 ID 도메인 역할에 지정합니다.
`epmautomate assignRole admin_role_file.csv "Service Administrator"`
- 사용자를 애플리케이션 역할에 지정합니다.
`epmautomate assignRole example_file.csv "Task List Access Manager"`

autoPredict

Planning 또는 Planning 모듈의 기존 자동 예측 정의를 기준으로 향후 성과 예측을 생성합니다.

이 명령은 애플리케이션에 지정된 자동 예측 정의에서 확인된 각 멤버의 과거 데이터를 사용하는 작업을 시작합니다. 자동 예측 기능을 사용하는 애플리케이션 및 예측 설정에 대한 자세한 내용은 *Planning 관리*의 자동 예측을 사용하여 자동으로 실행되도록 예측 설정을 참조하십시오.

적용 대상

애플리케이션에서 하이브리드 Oracle Essbase 큐브가 사용으로 설정된 경우 Planning 및 Planning Modules.

필수 역할

서비스 관리자

사용법

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

- `PREDICTION_DEFINITION`은 애플리케이션에서 사용할 수 있는 자동 예측 정의의 이름입니다.
- `forceRun`은 초기 실행 이후에 기본 정의가 변경되지 않은 경우 예측을 실행할지 여부를 지정합니다(선택사항). 기본값은 `false`입니다.

작업 정의가 변경되지 않은 경우에도 자동 예측 작업을 실행하려면 이 매개변수의 값을 true로 설정합니다. 작업을 처음 실행할 때 기본값(false)을 사용하여 예측을 한 번 실행합니다.

- paginatedDim은 별도의 스레드에서 예측을 병렬 실행하여 자동 예측 작업 속도를 높이는 데 사용되는 차원을 지정합니다(선택사항). 해당 병렬 스레드를 효율적으로 실행하려면 각 예측 스레드에 데이터가 균등하게 분산되는 차원을 선택합니다.

예

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

calculateModel

Enterprise Profitability and Cost Management 애플리케이션에서 계산 프로세스를 실행합니다.

적용 대상

Enterprise Profitability and Cost Management

필수 역할

서비스 관리자

사용법

① Note

대체 변수가 있으면 실제 매개변수 값 대신 대체 변수를 사용할 수 있습니다. 자세한 내용은 [일부 Enterprise Profitability and Cost Management 명령에서 대체 변수 사용](#)을 참조하십시오.

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

- **POV_NAME**은 계산할 데이터 POV의 이름입니다. 여러 POV를 계산하려면 구분자인 쉼표로 구분하여 POV 이름을 나열합니다. POV 이름을 구분하는 데 다른 구분자를 사용하지 마십시오. 멤버 이름에 공백이 있는 경우 POV 이름 목록을 큰 따옴표로 묶습니다.
- **MODEL_NAME**은 계산할 모델의 이름입니다. 이름에 공백이 포함된 경우 모델 이름을 큰 따옴표로 묶습니다.
- **EXECUTION_TYPE**은 다음 중 하나로, 규칙 실행 유형을 나타냅니다.
 - **ALL_RULES** - 모든 규칙을 사용하여 POV를 계산합니다.
이 값을 지정하는 경우 규칙 서브세트 또는 단일 규칙 관련 런타임 매개변수(예: rulesetSeqNumStart, rulesetSeqNumEnd, ruleName)를 지정하지 마십시오.
 - **RULESET_SUBSET** - 규칙 세트의 서브세트를 사용하여 POV를 계산합니다.
이 값을 사용하는 경우 rulesetSeqNumStart 및 rulesetSeqNumEnd 값을 런타임 매개변수로 지정해야 합니다.
 - **SINGLE_RULE** - 특정 규칙을 실행하여 POV를 계산합니다.
이 값을 사용하는 경우 ruleName만 런타임 매개변수로 지정해야 합니다.
 - **RUN_FROM_RULE** - 특정 규칙부터 시작하여 POV에서 계산을 실행합니다.
이 값을 사용하는 경우 ruleName만 런타임 매개변수로 지정해야 합니다.
 - **STOP_AFTER_RULE** - 특정 규칙이 계산을 완료한 후 POV 계산을 중지합니다.

이 값을 사용하는 경우 ruleName만 런타임 매개변수로 지정해야 합니다.

- povDelimiter는 선택사항으로, POV 값에서 사용되는 구분자입니다. 기본 구분자는 ::(이중 콜론)입니다. 구분자는 큰따옴표로 묶어야 합니다. 다음 구분자만 지원됩니다.
 - _(밑줄)
 - #(해시)
 - ~(물결표)
 - %(퍼센트)
 - ;(세미콜론)
 - :(콜론)
 - -(대시)
- optimizeForReporting=true|false는 선택사항으로, 계산을 실행할 때 보고에 최적화를 사용할지 여부를 지정합니다. 기본값은 false입니다. 단일 규칙 또는 순차적인 일련의 POV를 실행하는 등의 경우에 집계 생성 단계를 건너뛰어 처리 시간을 절약하려면 이 값을 false로 설정합니다. 동시 계산 작업을 여러 개 실행하는 경우 모든 작업에 대해 optimizeForReporting=true를 설정합니다. 그러면 마지막으로 완료되는 작업만 집계를 수행하므로 중복 처리가 방지되고 실행 중인 작업이 느려지지 않습니다.
- captureDebugScripts=true|false는 선택사항으로, 인박스에 디버그 스크립트를 생성할지 여부를 나타냅니다. 오라클에서 계산 이슈 문제를 해결하는 데 이러한 스크립트가 필요할 수 있습니다. 기본값은 false입니다.
- comment="COMMENT"는 선택사항으로, 프로세스 설명을 큰따옴표로 묶어서 지정합니다.
- PARAMETER=VALUE는 선택사항으로, 계산을 실행할 런타임 매개변수와 해당 값을 나타냅니다. 프로세스에 필요한 개수만큼 매개변수와 값 쌍을 지정합니다. 적합한 매개변수 및 해당 값은 다음과 같습니다.
 - rulesetSeqNumStart는 실행할 규칙 세트에서 첫번째 규칙의 순번을 지정합니다. EXECUTION_TYPE=RULESET_SUBSET가 사용되는 경우에만 적합합니다.
 - rulesetSeqNumEnd는 실행할 규칙 세트에서 마지막 규칙의 순번을 지정합니다. EXECUTION_TYPE=RULESET_SUBSET가 사용되는 경우에만 적합합니다.
 - ruleName은 실행할 규칙의 이름을 지정합니다. 공백 문자가 포함된 경우 값을 큰 따옴표로 묶습니다. EXECUTION_TYPE 값이 SINGLE_RULE, RUN_FROM_RULE 또는 STOP_AFTER_RULE로 설정된 경우에만 적합합니다.
 - clearCalculatedData=true|false는 기존 계산을 지울지 여부를 지정합니다. 기본값은 false입니다.
 - executeCalculations=true|false는 계산을 실행할지 여부를 지정합니다. 기본값은 false입니다.

① Note

매개변수 값(true 및 false)은 모두 소문자여야 합니다.

예

- 모든 규칙을 실행하여 단일 POV 계산:


```
epmautomate calculateModel FY22::Jan::Actual::Working "10 Actuals Allocation Process" ALL_RULES clearCalculatedData=true executeCalculations=true optimizeForReporting=true comment="Running all rules to calculate a POV"
```

- 모든 규칙을 실행하여 여러 POV 계산(사용자정의 구분자 사용):

```
epmautomate calculateModel
"FY22_Jan_Actual_Working,FY22_Feb_Actual_Working,FY22_Mar_Actual_Working" "10
Actuals Allocation Process" ALL_RULES clearCalculatedData=true
executeCalculations=true optimizeForReporting=true captureDebugScripts=true
comment="Runing calculation for multiple POVs" povDelimiter="_"
```
- 다양한 규칙 세트를 실행하여 POV 계산:

```
epmautomate calculateModel FY22::Jan::Actual::Working "10 Actuals Allocation
Process" RULESET_SUBSET rulesetSeqNumStart=10 rulesetSeqNumEnd=20
clearCalculatedData=true executeCalculations=true comment="Running a subset of
rule sets"
```
- 특정 규칙을 실행하여 POV 계산:

```
epmautomate calculateModel FY22::Jan::Actual::Working "10 Actuals Allocation
Process" SINGLE_RULE ruleName="Rent and Utilities Reassignment"
clearCalculatedData=true executeCalculations=true comment="Running a specific
rule"
```

clearCube

`clear cube` 유형의 작업에 지정된 설정을 사용하여 입력 및 보고 큐브의 특정 데이터를 삭제합니다.

이 명령은 애플리케이션의 관계형 테이블에 있는 애플리케이션 정의는 삭제하지 않습니다. *Planning 관리*에서 큐브 지우기를 참조하십시오.

적용 대상

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

필수 역할

서비스 관리자

사용법

`epmautomate clearCube JOB_NAME` 설명: `JOB_NAME`은 애플리케이션에서 정의된 작업의 이름입니다.

예

```
epmautomate clearCube ClearPlan1
```

clearDataByPointOfView

Enterprise Profitability and Cost Management 큐브의 특정 POV 데이터를 지웁니다.

적용 대상

Enterprise Profitability and Cost Management

필수 역할

서비스 관리자

사용법

epmAutomate clearDataByPointOfView *POV_NAME* [cubeName=*CUBE_NAME*]
[*PARAMETER=VALUE*] 설명:

- *POV_NAME*은 애플리케이션의 POV 이름입니다.
- cubeName은 선택사항으로, 데이터를 지울 큐브의 이름입니다. 기본값은 PCM_CLC입니다.
- *PARAMETER=VALUE*는 선택적 런타임 매개변수와 해당 값을 나타냅니다. 프로세스에 필요한 개수만큼 매개변수와 값 쌍을 지정합니다. 적합한 매개변수 및 해당 값은 다음과 같습니다.
 - povDelimiter는 POV 값에서 사용되는 구분자입니다. 기본값은 ::(이중 콜론)입니다. 이 값은 큰따옴표로 묶어야 합니다. 예: povDelimiter="_" . 기본값 외에는 다음 구분자만 지원됩니다. _(밑줄), #(해시), ~(물결표), %(퍼센트), ; (세미콜론), :(콜론), -(대시).
 - clearInput=true|false는 입력 데이터를 지울지 여부를 지정합니다. 기본값은 false입니다.
 - clearAllocatedValues=true|false는 할당된 값을 지울지 여부를 지정합니다. 기본값은 false입니다.
 - clearAdjustmentValues=true|false는 조정 값을 지울지 여부를 지정합니다. 기본값은 false입니다.

Note

- * 매개변수 값(true 또는 false)은 모두 소문자여야 합니다.
- * clearInput, clearAllocatedValues 또는 clearAdjustmentValues 매개변수 중 하나 이상을 true로 설정해야 합니다.

예

- 기본 POV 구분자를 사용하는 기본 PCM_CLC 큐브의 POV에서 데이터 지우기:
epmAutomate clearDataByPointOfView FY21::Jan::Actual::Working clearInput=true
clearAllocatedValues=true clearAdjustmentValues=true
- 사용자정의 POV 구분자를 사용하는 특정 큐브의 POV에서 입력 데이터와 할당된 값 지우기:
epmAutomate clearDataByPointOfView FY21_Jan_Actual_Working cubeName=PCM_REP
povDelimiter="_" clearInput=true clearAllocatedValues=true
- 사용자정의 POV 구분자를 사용하는 특정 큐브의 POV에서 입력 데이터 지우기:
epmAutomate clearDataByPointOfView FY21:Jan:Actual:Working cubeName=PCM_REP
povDelimiter=":" clearInput=true

clearDataByProfile

Financial Consolidation and Close 및 Tax Reporting에 정의된 데이터 지우기 프로파일에서 확인된 항목(예: 영역)의 데이터를 지웁니다.

적용 대상

Financial Consolidation and Close, Tax Reporting

필수 역할

서비스 관리자

사용법

`epmautomate clearDataByProfile PROFILE_NAME` 설명: `PROFILE_NAME`은 데이터 지우기 프로파일의 이름입니다.

예

```
epmautomate clearDataByProfile clearDataProfile_01
```

clearPOV

Profitability and Cost Management 애플리케이션에서 POV(Point of View) 조합의 모델 아티팩트 및 데이터와 POV 내의 데이터 영역을 지웁니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

① 주

대체 변수가 있으면 실제 매개변수 값 대신 대체 변수를 사용할 수 있습니다. 자세한 내용은 [일부 Enterprise Profitability and Cost Management 명령에서 대체 변수 사용](#)을 참조하십시오.

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

- `APPLICATION_NAME`은 Profitability and Cost Management 애플리케이션의 이름입니다.
- `POV_NAME`은 애플리케이션의 POV입니다. 이 값은 필수입니다.
- `QUERY_NAME`은 선택사항으로, Profitability and Cost Management에서 정의된 쿼리의 이름입니다. 지정된 경우 이 쿼리는 POV 내에서 데이터 영역을 지우는 데 사용됩니다.

① 주

쿼리 이름을 지정하는 경우, 모든 런타임 매개변수(아래 참조)의 값을 `false`로 설정해야 합니다.

- `PARAMETER=VALUE`는 POV를 지울 런타임 매개변수와 해당 값을 나타냅니다. 프로세스에 필요한 개수만큼 매개변수와 값 쌍을 지정합니다. 적합한 매개변수가 하나 이상 필요하며 해당 값은 다음과 같습니다.
 - `isManageRule=true|false`는 규칙을 지울지 여부를 지정합니다.

- `isInputData=true|false`는 입력 데이터를 지울지 여부를 지정합니다.
- `isAllocatedValues=true|false`는 할당 값을 지울지 여부를 지정합니다.
- `isAdjustmentValues=true|false`는 조정 값을 지울지 여부를 지정합니다.

① 주

매개변수 값(`true` 또는 `false`)은 모두 소문자여야 합니다.

POV의 데이터 영역을 지우려면(`QUERY_NAME`이 지정된 경우), 런타임 매개변수 (`isManageRule`, `isInputData`, `isAllocatedValues` 및 `isAdjustmentValues`)의 값을 `false`로 설정해야 합니다.

- `stringDelimiter="DELIMITER"`는 POV 값에서 사용되는 구분자를 지정합니다. 구분자는 큰따옴표로 묶어야 합니다. 기본값은 `_`(밑줄)입니다.

예

- POV에서 모든 모델 아티팩트 및 데이터 지우기: `epmautomate clearPOV BksML12 2012_Jan_Actual isManageRule=true isInputData=true isAllocatedValues=true isAdjustmentValues=true stringDelimiter="_"`
- POV 내에서 데이터 영역 지우기: `epmautomate clearPOV BksML12 2012_Jan_Actual queryName=BksML12_2012_Jan_clear_query isManageRule=false isInputData=false isAllocatedValues=false isAdjustmentValues=false stringDelimiter="_"`

cloneEnvironment

현재 환경을 복제하고 선택적으로 ID 도메인 아티팩트(사용자 및 사전 정의된 역할 지정), Data Management 레코드, 감사 레코드, 작업 콘솔 레코드, 애플리케이션 등록정보, 인박스 및 아웃박스 콘텐츠, 저장된 스냅샷을 복제합니다. 이 명령은 마이그레이션의 환경 복제 기능 대신 사용할 수 있습니다.

소스 및 타겟 환경의 스케줄링된 일별 유지관리 후 복제를 시작합니다. 복제가 진행되는 동안 소스 환경의 일별 유지관리가 시작되면 복제 프로세스가 종료됩니다. 일별 유지관리 시작 시간에 복제가 진행 중이더라도 타겟 환경의 복제 프로세스는 영향을 받지 않습니다. 이 시나리오에서 일일 유지관리는 복제가 완료된 후 실행됩니다.

환경 복제에 오랜 시간이 걸리는 경우에는 일일 유지관리 시작 시간을 다시 스케줄링하여 복제 프로세스가 종료되지 않도록 합니다. 일별 유지관리 시작 시간을 재설정하는 방법에 대한 자세한 내용은 다음 정보 소스를 참조하십시오.

- [setDailyMaintenanceStartTime](#)
- *관리자용 시작 가이드*의 일별 유지관리 관리.
- *Oracle Fusion Cloud EPM용 REST API 가이드*의 일일 유지관리 기간 시간 보기 및 설정.

Note

- **Data Management:** 스테이지 테이블에 매우 많은 수의 레코드가 있는 경우 Data Management 레코드를 복제하는 데 시간이 오래 걸릴 수 있습니다. 마찬가지로, 인박스 및 아웃박스 콘텐츠와 저장된 스냅샷 복제도 많은 양의 데이터가 포함되어 있는 경우 특히 상당한 시간이 걸릴 수 있습니다.
- **Planning:** Planning 비즈니스 프로세스에 사용자정의 기간 멤버로 대체된 이름이 변경된 시드된 기간 멤버가 포함된 경우 복제가 실패할 수 있습니다. 예를 들어, 시드된 *YearTotal* 기간 멤버를 *unused_YearTotal*로 이름을 변경한 다음 대체 유형 기간 멤버를 원래 시드된 멤버 이름(이 예에서 *YearTotal*)로 추가했습니다. 이 시나리오에서는 환경 복제가 실패할 수 있습니다.

이러한 항목에 대한 자세한 내용은 *마이그레이션 관리*에서 Cloud EPM 환경 복제를 참조하십시오.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

```
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]
[ApplicationProperties=true|false] 설명:
```

Note

- `dataManagement` 매개변수는 Cloud EDM 및 Narrative Reporting 환경에 적용되지 않습니다.
데이터 관리 레코드는 소스 환경과 타겟 환경 둘 다 월별 업데이트 버전이 동일하거나 타겟 환경이 소스 환경보다 업데이트 버전이 하나 더 최신인 경우에만 복제합니다. 예를 들어 22.01 Data Management 레코드는 다른 22.01 환경 또는 22.02 환경에만 복제할 수 있습니다.
- `jobConsole` 매개변수는 Planning, Planning 모듈, 무형식, Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management, Sales Planning, 전략적 Workforce Planning에만 적용됩니다.
- `appAudit` 매개변수는 Planning, Planning 모듈, 무형식, Enterprise Profitability and Cost Management, Sales Planning, 전략적 Workforce Planning에만 적용됩니다. Financial Consolidation and Close 및 Tax Reporting에 대한 감사 정보는 기본적으로 스냅샷에 포함되어 있습니다.
- `dataManagement`, `jobConsole` 또는 `appAudit` 매개변수를 환경에 적용할 수 없는 경우 명령은 지정된 값을 무시합니다.
- `ApplicationProperties` 매개변수는 Account Reconciliation에만 적용됩니다.

- `TARGET_USERNAME`은 타겟 환경의 서비스 관리자 ID입니다. 타겟 ID 도메인 사용자 이름(SSO 사용자 이름이 아님)을 사용해야 합니다.

Note

사용자 및 역할 지정을 복제하려면 이 사용자는 ID 도메인 관리자 역할에 지정된 서비스 관리자여야 합니다.

- `TARGET_PASSWORD`는 `TARGET_USERNAME`에 의해 확인된 사용자에 대한 암호화된 비밀번호 또는 OAuth 2.0 새로고침 토큰의 이름 및 위치입니다.
- `TARGET_URL`은 복제된 환경이 될 환경의 `CLOUD-EPM_BASE_URL`입니다.
- `SnapshotName`은 선택사항으로, 복제에 사용해야 하는 스냅샷의 이름입니다. 이 스냅샷은 소스 환경에 있어야 합니다. 기본값은 Artifact Snapshot이며, 최종 유지관리 스냅샷을 사용하여 환경을 복제합니다.

Note

이는 일별 유지관리 프로세스에서 생성된 스냅샷이어야 합니다. [restoreBackup](#) 명령을 사용하여 소스 환경에 복사되는 스냅샷일 수 있습니다.

- `UsersAndPreDefinedRoles`는 선택사항으로, 사용자 및 사전 정의된 역할 지정을 복제할지 여부를 나타냅니다(액세스 제어 그룹은 항상 복제됨). 기본값은 `false`입니다. 이 옵션을 사용하려면 `TARGET_USER_NAME`으로 확인된 사용자에게 타겟 환경에서 ID 도메인 관리자 역할이 있어야 합니다.

이 확인란을 선택한 다음 ID 도메인 관리자가 아닌 사용자가 환경을 복제하는 경우 사용자 및 사전 정의된 역할 임포트가 실패합니다. 마이그레이션 상태 보고서에 다음 오류가 기록됨: 외부 디렉토리 아티팩트 <artifact_name> 임포트 실패. <user_name> 사용자는 이 작업을 수행할 수 있는 권한이 없습니다. 이 작업을 수행하려면 사용자에게 ID 도메인 관리자 역할이 있어야 합니다.

- 사용자를 임포트하지 않고 소스 스냅샷의 사용자가 타겟 환경의 사전 정의된 역할에 지정되지 않은 경우 오류(EPMIE-00070: 지정된 역할 임포트 중 사용자를 찾지 못했습니다.)가 표시됩니다.
- ID 도메인 관리자 역할 지정은 복제되지 않습니다. ID 도메인 관리자 역할만 지정된 사용자는 타겟 환경으로 복제되지 않습니다. 소스 환경에서 ID 도메인 관리자 역할 및 사전 정의된 역할 조합에 지정된 사용자는 복제되지만 타겟 환경의 해당 사전 정의된 역할에만 지정됩니다. 이러한 사용자는 타겟 환경에서 ID 도메인 관리자 역할에 지정되지 않습니다.
- 사용자의 사전 정의된 역할 변경사항은 소스 스냅샷에서 지정된 역할에 따라 업데이트됩니다. 그러나 타겟의 역할 지정이 소스 스냅샷의 역할 지정과 일치하도록 제거되지는 않습니다. 예를 들어 jdoe가 타겟 환경에서는 고급 사용자 사전 정의된 역할에 지정되었지만 소스 스냅샷에서는 사용자 역할만 있다고 가정합니다. 이런 상황에서 이 명령은 사용자 역할에 jdoe를 지정하고 타겟 환경의 고급 사용자 역할 지정을 제거하지 않습니다.
- 이 명령은 소스 스냅샷에 없는 경우에도 타겟 환경의 기존 사용자를 삭제하지 않습니다. 예를 들어 jdoe는 타겟 환경에 계정이 있지만 소스 스냅샷에는 이 계정이 없습니다. 이런 상황에서 타겟 환경의 jdoe 계정은 삭제되지 않습니다.
- 이 명령은 타겟 환경에 없는 사용자를 추가합니다. 소스 스냅샷과 다른 경우에도 타겟 환경의 현재 사용자 등록정보를 업데이트하지 않습니다. 예를 들어 소스 스냅샷에 있는 jdoe의 성 철자가 타겟 환경과 다른 경우에도 타겟 환경에서 변경되지 않습니다. 타겟 환경의 새 사용자에게 무작위 비밀번호가 지정됩니다. 새 사용자는 비밀번호를 변경하도록 요청하는 계정 활성화 전자메일을 받게 됩니다.
- 이 명령은 소스 스냅샷과 다른 경우에도 타겟 환경의 기존 사용자 비밀번호를 변경하지 않습니다.
- `dataManagement=true|false`는 선택사항으로, 소스 환경의 Data Management 레코드를 타겟 환경으로 복제합니다. 기본값은 `true`이며, Data Management 레코드를 복제합니다. Data Management 레코드를 복제하지 않으려면 이 값을 `false`로 설정하십시오.
- `appAudit=true|false`는 선택사항으로, 소스 환경의 감사 레코드를 타겟 환경으로 복제합니다. 기본값은 `true`이며, 애플리케이션 감사 데이터를 복제합니다. 애플리케이션 감사 데이터를 타겟 환경에 복제하지 않으려면 이 값을 `false`로 설정하십시오.
- `jobConsole=true|false`는 선택사항으로, 소스 환경의 작업 콘솔 레코드를 타겟 환경으로 복제합니다. 기본값은 `true`입니다. 작업 콘솔 레코드를 복제하지 않으려면 이 값을 `false`로 설정하십시오.
- `storedSnapshotsAndFiles`는 선택사항으로, 이 명령에서 인박스 및 아웃박스 콘텐츠와 저장된 스냅샷을 복제하는지 여부를 나타냅니다. 기본값은 `false`입니다.

① Note

인박스 및 아웃박스의 최상위 폴더만 복제되며 하위 폴더는 복제되지 않습니다. 하위 폴더의 콘텐츠를 유지해야 하는 경우 로컬 컴퓨터에 백업한 다음 타겟 환경에 업로드합니다.

- `DailyMaintenanceStartTime`는 선택사항으로, 복제된 타겟 환경의 유지관리 시작 시간을 소스 환경의 해당 시간으로 재설정합니다. 기본값은 `true`입니다. 타겟 환경의 현재 유지관리 시작 시간을 유지하려면 이 값을 `false`로 설정하십시오.
- `ApplicationProperties`는 선택사항으로, Account Reconciliation 애플리케이션 설정 (전자메일 통지 설정/해제 설정, Redwood 환경, 테마, 비즈니스 프로세스 이름, 로고 이미지 및 배경 이미지)을 복제합니다. 기본값은 `true`입니다. 현재 애플리케이션 설정을 유지하려는 타겟 환경의 경우 이 값을 `false`로 설정합니다.

예

- 환경, 사용자 및 사전 정의된 역할 지정, 감사 데이터, 작업 콘솔 레코드, Data Management 레코드를 복제합니다. 또한, 타겟 환경의 유지관리 시작 시간을 소스 환경의 해당 시간으로 변경합니다.

```
epmAutomate cloneEnvironment serviceAdmin Password.epw https://test-cloudpln.pbcs.us1.oraclecloud.com UsersAndPreDefinedRoles=true
```
- 타겟 환경의 유지관리 시작 시간을 변경하지 않고 인박스 및 아웃박스 콘텐츠, 저장된 스냅샷은 포함되지만 사용자 및 사전 정의된 역할 지정, Data Management 레코드, 감사 데이터, 작업 콘솔 레코드는 포함되지 않은 환경을 복제합니다.

```
epmAutomate cloneEnvironment serviceAdmin Password.epw https://test-cloudpln.pbcs.us1.oraclecloud.com DataManagement=false appAudit=false jobConsole=false storedSnapshotsAndFiles=true DailyMaintenanceStartTime=false
```
- 사용자정의 스냅샷을 사용하여 전체 환경(사용자 및 사전 정의된 역할 지정, 감사 데이터, 작업 콘솔 레코드, 인박스 및 아웃박스 콘텐츠, 저장된 스냅샷, Data Management 레코드)을 복제합니다. 또한, 타겟 환경의 유지관리 시작 시간을 소스 환경의 해당 시간으로 변경합니다.

```
epmAutomate cloneEnvironment serviceAdmin Password.epw https://test-cloudpln.pbcs.us1.oraclecloud.com UsersAndPreDefinedRoles=true storedSnapshotsAndFiles=true SnapshotName=SampleSnapshot
```
- **Account Reconciliation만 해당:** 환경, 사용자, 사전 정의된 역할 지정 및 Data Management 레코드를 복제하지만 타겟 환경의 애플리케이션 설정은 유지합니다.

```
epmAutomate cloneEnvironment serviceAdmin Password.epw https://test-cloudarcs.arcs.epm.us.oraclecloud.com UsersAndPreDefinedRoles=true ApplicationProperties=false
```

compactCube

큐브 압축 유형의 작업에 정의된 설정을 사용하여 집계 저장영역(ASO) 큐브의 아웃라인 파일을 압축합니다. 아웃라인을 압축하면 큐브 아웃라인 파일을 최적의 크기로 유지하는 데 도움이 됩니다. 압축은 큐브의 데이터를 지우거나 비즈니스 프로세스의 변경사항을 Oracle Essbase로 푸시하지 않습니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

`epmautomate compactCube ASO_CUBE_NAME` 설명: `ASO_CUBE_NAME`은 압축할 아웃라인, ASO 큐브의 이름입니다.

예

```
epmautomate compactCube Vis1Aso
```

copyDataByPointOfView

큐브의 소스 POV 데이터를 동일하거나 다른 Enterprise Profitability and Cost Management 큐브의 대상 POV에 복사합니다.

적용 대상

Enterprise Profitability and Cost Management

필수 역할

서비스 관리자

사용법

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

- *SOURCE_POV_NAME*은 데이터를 복사할 소스 POV의 이름입니다.
- *TARGET_POV_NAME*은 소스의 데이터를 복사할 적합한 타겟 POV의 이름입니다.
- *copyType*은 소스 POV에서 복사할 데이터를 나타냅니다. 적합한 값은 다음과 같습니다.
 - *ALL_DATA* - 모든 입력과 계산된 데이터를 대상 POV에 복사합니다.
 - *INPUT* - 드라이버 데이터를 비롯한 모든 입력 데이터를 대상 POV에 복사합니다.
- *SOURCE_CUBE_NAME*은 소스 POV가 포함된 큐브의 이름입니다.
- *TARGET_CUBE_NAME*은 타겟 POV가 포함된 큐브의 이름입니다.
- *PARAMETER=VALUE*는 선택적 런타임 매개변수와 해당 값을 나타냅니다. 프로세스에 필요한 개수만큼 매개변수와 값 쌍을 지정합니다. 적합한 매개변수 및 해당 값은 다음과 같습니다.
 - *povDelimiter*는 선택사항으로, POV 값에서 사용되는 구분자입니다. 기본값은 ::(이중 콜론)입니다. 이 값은 큰따옴표로 묶어야 합니다. 예: *povDelimiter*="_" . 기본값 외에는 다음 구분자만 지원됩니다. _(밑줄), #(해시), ~(물결표), %(퍼센트), ; (세미콜론), :(콜론), -(대시).
 - *createDestPOV=true|false*는 타겟 POV가 없을 경우 생성할지 여부를 지정합니다. 기본값은 *false*입니다. 대상 POV가 없는 경우 이 매개변수 값을 *true*로 설정해야 합니다.

예

- 모든 데이터를 동일한 큐브의 다른 POV에 복사:


```
epmAutomate copyDataByPointOfView FY21_Jan_Actual_Working
FY22_Jan_Actual_Working ALL_DATA PCM_CLC PCM_CLC povDelimiter="_"
createDestPOV=true
```
- 모든 데이터를 다른 큐브의 다른 POV에 복사:


```
epmAutomate copyDataByPointOfView FY21_Jan_Actual_Working
FY22_Jan_Actual_Working ALL_DATA PCM_CLC PCM_REP povDelimiter="_"
createDestPOV=true
```
- 입력 데이터를 동일한 큐브의 다른 POV에 복사:


```
epmAutomate copyDataByPointOfView FY21_Jan_Actual_Working
FY22_Jan_Actual_Working INPUT PCM_CLC PCM_CLC povDelimiter="_"
createDestPOV=true
```
- 입력 데이터를 다른 큐브의 다른 POV에 복사:

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

copyDataByProfile

데이터 복사 프로파일에서 확인된 항목(예: 영역)의 데이터를 복사합니다.

적용 대상

Financial Consolidation and Close, Tax Reporting

필수 역할

서비스 관리자

사용법

epmautomate copyDataByProfile *PROFILE_NAME*, 설명: *PROFILE_NAME*은 Financial Consolidation and Close 및 Tax Reporting에서 정의된 데이터 복사 프로파일의 이름입니다.

예

```
epmautomate copyDataByProfile copyDataProfile_01
```

copyFileFromInstance

소스 환경에서 이 명령을 실행하는 환경으로 파일을 복사합니다.

이 명령을 실행하기 전에 EPM Automate를 사용하여 파일을 복사할 환경에 로그인합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

```
epmautomate copyFileFromInstance SOURCE_FILE_NAME USERNAME PASSWORD_FILE URL
TARGET_FILE_NAME 설명:
```

- *SOURCE_FILE_NAME*은 소스 환경에서 복사할 파일의 이름(확장자 포함)입니다.
- *USERNAME*은 소스 환경 서비스 관리자의 사용자 이름입니다.
- *PASSWORD_FILE*은 소스 환경 서비스 관리자의 암호화된 비밀번호가 포함된 파일 이름과 위치입니다.
- *URL*은 소스 환경의 URL입니다.
- *TARGET_FILE_NAME*은 이 명령을 실행하는 환경 내 파일의 고유 이름(확장자 포함)입니다.

예

```
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

Oracle Object Storage 버킷의 파일 또는 백업 스냅샷을 현재 환경에 복사합니다.

백업 스냅샷을 복사하는 경우 이 명령은 Object Storage 버킷에서 스냅샷을 복사한 다음, Oracle Fusion Cloud Enterprise Performance Management에서 스냅샷 콘텐츠를 추출합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

서비스 관리자

사용법

```
epmautomate copyFromObjectStorage USERNAME PASSWORD URL TARGET_FILE_NAME 설명:
```

- *USERNAME*은 Oracle Object Storage Cloud에서 필요한 액세스 권한이 있는 사용자의 ID입니다. 페더레이션된 ID 제공자에서 생성된 사용자의 경우 사용자의 전체 이름을 지정합니다(예: exampleIdP/jdoe 또는 exampleIdP/john.doe@example.com, 설명: exampleIdP는 페더레이션된 ID 제공자의 이름). 다른 사용자의 경우 사용자 ID를 제공합니다.
- *PASSWORD*는 사용자와 연계된 Swift 비밀번호 또는 인증 토큰입니다. 이 비밀번호는 Object Storage Console에 로그인하는 데 사용하는 비밀번호와 다릅니다. 인증 토큰은 Swift 클라이언트로 인증하는 경우와 같이 타사 API로 인증하는 데 사용하는 Oracle에서 생성된 토큰입니다. 이 토큰 생성에 대한 지침은 *Oracle Cloud Infrastructure 설명서*에서 [인증 토큰을 생성하려면](#)을 참조하십시오.
- *URL*은 복사할 객체의 이름과 버킷 이름을 포함하는 Oracle Object Storage Cloud 버킷의 URL입니다.
URL 형식은 다음과 같습니다.

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

이 URL의 구성요소는 다음과 같습니다.

- *region_identifier*는 Oracle Cloud 지역입니다.
- *namespace*는 모든 버킷 및 객체에 대한 최상위 레벨 컨테이너입니다. 각 Oracle Cloud Infrastructure 테넌트는 계정 생성 시에 변경이 불가능하며 고유한 시스템 생성 Object Storage 네임스페이스 이름이 지정됩니다. 테넌시 네임스페이스 이름(예: axaxnprorw5)은 모든 영역에서 유효합니다.
- *bucket_name*은 데이터 및 파일을 저장하는 논리 컨테이너의 이름입니다. 버킷은 구획에서 구성되고 유지관리됩니다. 시스템 생성 버킷 이름(예: bucket-20210301-1359)에는 현재 연도, 월, 일 및 시간이 반영됩니다.

- `object_name`은 Oracle Object Storage Cloud에서 복사할 스냅샷 또는 파일의 이름입니다. 이 값은 Object Storage Cloud에 있는 객체의 전체 이름과 정확히 일치해야 합니다. 객체 이름에 포함되어 있지 않은 경우 `.zip` 같은 확장명을 사용하지 마십시오.

자세한 내용은 *Oracle Cloud Infrastructure 설명서*의 다음 항목을 참조하십시오.

- [지역 및 가용성 도메인](#)
- [Object Storage 네임스페이스 이해](#)
- [버킷 관리](#)
- `TARGET_FILE_NAME`은 Cloud EPM 환경에 있는 파일 또는 스냅샷의 고유 이름입니다. 스냅샷을 복사할 때 이 파일 이름을 [importSnapshot](#) 명령과 함께 사용할 수 있도록 ZIP 확장자를 지정하지 마십시오.
100MB보다 큰 파일은 해당 세그먼트를 확인하는 매니페스트 파일과 함께 논리적 디렉토리 내의 Oracle Object Storage에 저장됩니다. 논리 디렉토리의 이름을 `TARGET_FILE_NAME`으로 지정합니다.

예

아래 예에서 `URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET`을 이 형식의 작동하는 URL인 `https://swiftobjectstorage.region_identifier.oraclecloud.com/v1/namespace/bucket_name/`으로 바꿉니다.

- Oracle Object Storage 버킷의 `backup_Snapshot_12_05_20.zip` 스냅샷을 Cloud EPM으로 복사하고 이름을 바꿉니다.

```
epmautomate copyFromObjectStorage oracleidentitycloudservice/jDoe example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/backup_Snapshot_12_05_20.zip snapshot_from_osc
```
- Oracle Object Storage 버킷의 `backup_Snapshot_12_05_20` 스냅샷을 Cloud EPM으로 복사하고 이름을 바꿉니다.

```
epmautomate copyFromObjectStorage oracleidentitycloudservice/jDoe example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/backup_Snapshot_12_05_20 snapshot_from_osc
```
- Oracle Object Storage 버킷의 `backup_Snapshot_12_05_20` 스냅샷을 이름을 바꾸지 않고 Cloud EPM으로 복사합니다.

```
epmautomate copyFromObjectStorage oracleidentitycloudservice/jDoe example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/backup_snapshot_12_05_20 backup_snapshot_12_05_20
```
- Oracle Object Storage 버킷에서 Cloud EPM으로 파일을 복사합니다.

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

copyFromSFTP

보안 SFTP 서버에서 Oracle Fusion Cloud Enterprise Performance Management 환경으로 직접 파일을 복사합니다. 이 명령을 사용하면, SFTP 클라이언트를 사용하여 다른 서버로 파일을 먼저 다운로드한 후 EPM Automate 또는 REST API를 사용하여 Cloud EPM에 해당 파일을 업로드하지 않고도 SFTP 서버에서 직접 파일을 복사할 수 있습니다.

IP 허용 목록이 SFTP 서버에 대해 사용으로 설정된 경우 Cloud EPM 환경을 호스트하는 OCI 지역의 아웃바운드 IP 주소를 SFTP 서버의 IP 허용 목록에 추가해야 합니다. OCI 지역의 아웃바운드 IP 주소는 *작업 가이드*의 Cloud EPM 데이터 센터 및 지역의 아웃바운드 IP 주소를 참조하십시오.

Note

이 명령은 포트 22에서 실행 중인 SFTP 서버만 지원합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

```
epmautomate copyFromSftp SFTP_SERVER_URL FILE_NAME username=USERNAME
password=PASSWORD:
```

- *SFTP_SERVER_URL*은 서버에서 복사할 파일의 위치 및 이름이 포함된 SFTP 서버의 URL입니다.
- *FILE_NAME*은 Cloud EPM 환경에 있는 파일의 파일 이름입니다. 예를 들어 기본 업로드 위치에 복사하지 않는 경우 경로를 지정합니다.
- *USERNAME*은 SFTP 서버에서 필요한 액세스 권한이 있는 사용자의 ID입니다.
- *PASSWORD*는 SFTP 사용자 비밀번호입니다.

예

```
epmautomate copyFromSftp sftp://mySFTP_Server/someDirectory/someFile.csv
myFile.csv username=jdoe password=Sftpwelcome
```

copyOwnershipDataToNextYear

소유권 데이터를 연도의 마지막 기간에서 다음 연도의 첫번째 기간으로 복사합니다.

초기 기본값 및 소유권 대체 설정은 동일한 연도 내의 기간 간에는 자동으로 차기 이월되지만 후속 연도의 기간으로는 자동으로 차기 이월되지 않습니다. 연도의 마지막 기간에서 다음 연도의 첫번째 기간으로 가장 최근 소유권 설정을 차기 이월하려면 POV에서 연도의 마지막 기간 소유권 설정을 다음 연도의 첫번째 기간으로 복사해야 합니다.

적용 대상

Financial Consolidation and Close 및 Tax Reporting.

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자

사용법

```
epmautomate copyOwnershipDataToNextYear Scenario Year 설명:
```

- Scenario는 소유권 데이터를 복사할 출처 시나리오의 이름입니다.
- Year는 다음 연도의 첫번째 기간으로 복사할 소유권 데이터의 출처 연도입니다.

예

```
epmautomate copyOwnershipDataToNextYear FCCS_total_Actual FY18
```

copyPOV

소스 POV의 모델 아티팩트와 Oracle Essbase 큐브 데이터를 대상 POV에 복사합니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

① 주

대체 변수가 있으면 실제 매개변수 값 대신 대체 변수를 사용할 수 있습니다. 자세한 내용은 [일부 Enterprise Profitability and Cost Management 명령에서 대체 변수 사용](#)을 참조하십시오.

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

- *APPLICATION_NAME*은 소스 POV가 포함된 Profitability and Cost Management 애플리케이션의 이름입니다.
- *SOURCE_POV_NAME*은 지정한 애플리케이션의 소스 POV 이름입니다.
- *TARGET_POV_NAME*은 초안 상태의 적합한 대상 POV 이름입니다.
- *PARAMETER=VALUE*는 POV를 복사할 런타임 매개변수와 해당 값을 나타냅니다. 프로세스에 필요한 개수만큼 매개변수와 값 쌍을 지정합니다. 적합한 매개변수 및 해당 값은 다음과 같습니다.
 - *isManageRule=true|false*는 규칙을 복사할지 여부를 지정합니다.
 - *isInputData=true | isAllData=true | isAllInputData=true(선택사항)*는 데이터를 복사하는 방법을 지정합니다. 다음 매개변수의 경우 기본값은 *false*입니다. 다음 중 하나만 *true*로 지정하십시오.
 - * 입력 데이터를 대상 POV로 복사하려면 *isInputData=true*를 지정합니다.
 - * 모든 입력 및 계산된 데이터를 대상 POV로 복사하려면 *isAllData=true*를 지정합니다.
 - * 드라이버 데이터를 포함하여 모든 입력 데이터를 대상 POV로 복사하려면 *AllInputData=true*를 지정합니다.
 - *modelViewName=NAME*은 소스 POV에서 타겟 POV로 복사할 데이터 슬라이스의 이름을 지정합니다.

- createDestPOV=true|false는 타겟 POV가 없을 경우 생성할지 여부를 지정합니다.
- nonEmptyTupleEnabled=true|false는 데이터가 있는 교차점만 명령에서 고려하도록 NET(비어 있지 않은 튜플)를 사용으로 설정할 것인지 여부를 지정합니다. 기본값은 true입니다. 이 값을 사용하면 드물지만 이 명령 때문에 Essbase 데이터 복사가 잘 수행되지 않을 수 있습니다. 이런 경우 성능이 향상되도록 nonEmptyTupleEnabled=false를 사용하여 기본값을 대체합니다.

① 주

매개변수 값(true 또는 false)은 모두 소문자여야 합니다.

- stringDelimiter="DELIMITER"는 POV 값에서 사용되는 구분자를 지정합니다. 구분자는 큰따옴표로 묶어야 합니다.

예

- `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

소스 환경에서 이 명령을 실행하는 환경(타겟)으로 현재 스냅샷을 복사합니다.

이 명령은 주로 다른 환경에서 현재 스냅샷을 복사(예: 테스트 환경에서 프로덕션 환경으로 복사)하여 환경을 마이그레이션하는 데 첫번째 단계로 사용됩니다. [importSnapshot](#) 명령을 사용하여 마이그레이션 프로세스를 완료합니다.

이 명령을 실행하기 전에 EPM Automate 세션을 시작하고 타겟 환경에 로그인합니다.

소스 환경의 스냅샷이 생성되는 동안(예: 일별 유지관리 중) 이 명령을 실행하여 현재 스냅샷을 복사하는 경우 File not found 또는 Invalid File 오류가 표시됩니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Strategic Workforce Planning, Sales Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

```
epmautomate copySnapshotFromInstance SNAPSHOT_NAME USERNAME PASSWORD_FILE URL
```

설명:

- SNAPSHOT_NAME은 소스 환경의 기존 스냅샷 이름입니다.

- `USERNAME`은 소스 환경 서비스 관리자의 사용자 이름입니다.
- `PASSWORD_FILE`은 소스 환경 서비스 관리자의 암호화된 비밀번호가 포함된 파일 이름과 위치입니다.
- `URL`은 소스 환경의 URL입니다.

예

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

copyToObjectStorage

현재 환경의 파일 또는 스냅샷을 Oracle Object Storage Cloud 버킷에 복사합니다.

스냅샷을 복사하는 경우 이 명령은 스냅샷의 콘텐츠를 zip으로 압축한 후 Oracle Object Storage에 복사합니다.

파일을 빠르고 원활하게 복사할 수 있도록 이 명령은 큰 파일(100MB보다 큼)을 10MB 세그먼트로 쪼개고(`FILE_NAME/FILE_NAME_object_store_bytes_seg_0 ~ FILE_NAME/FILE_NAME_object_store_bytes_seg_n`으로 이름 지정) 매니페스트 파일(`FILE_NAME/FILE_NAME.manifest`로 이름 지정)을 생성합니다. 파일 세그먼트는 매니페스트 파일과 함께 Oracle Object Storage에 저장됩니다. Object Storage Console에서 파일은 파일 세그먼트와 매니페스트 파일이 포함된 논리적 디렉토리로 표시됩니다.

100MB보다 작은 파일은 세그먼트로 분할되지 않고 원래 파일 이름으로 저장됩니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

서비스 관리자

사용법

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

- `SOURCE_FILE_NAME`은 Oracle Fusion Cloud Enterprise Performance Management에 있는 파일 또는 스냅샷의 이름입니다. 스냅샷을 복사하는 경우 ZIP 확장자를 지정하지 마십시오.
- `USERNAME`은 Oracle Object Storage Cloud에 쓰는 데 필요한 액세스 권한이 있는 사용자의 ID입니다.
페더레이션된 ID 제공자에서 생성된 사용자의 경우 사용자의 전체 이름을 지정합니다(예: `exampleIdP/jdoe` 또는 `exampleIdP/john.doe@example.com`, 설명: `exampleIdP`는 페더레이션된 ID 제공자의 이름). 다른 사용자의 경우 사용자 ID를 제공합니다.
- `PASSWORD`는 사용자와 연계된 Swift 비밀번호 또는 인증 토큰입니다. 이 비밀번호는 Object Storage Console에 로그인하는 데 사용하는 비밀번호와 다릅니다. 인증 토큰은 Swift 클라이언트로 인증하는 경우와 같이 타사 API로 인증하는 데 사용하는 Oracle에서 생성된 토큰입니다. 이 토큰 생성에 대한 지침은 [Oracle Cloud Infrastructure 설명서](#)에서 [인증 토큰을 생성하려면](#)을 참조하십시오.
- `URL`은 객체 이름 선택사항이 추가된 Oracle Object Storage Cloud 버킷의 URL입니다. 객체 이름이 없는 URL 형식은 다음과 같습니다.

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

객체 이름이 있는 URL 형식은 다음과 같습니다.

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

이 URL의 구성요소는 다음과 같습니다.

- region_identifier는 Oracle Cloud 지역입니다.
- namespace는 모든 버킷 및 객체에 대한 최상위 레벨 컨테이너입니다. 각 Oracle Cloud Infrastructure 테넌트는 계정 생성 시에 변경이 불가능하며 고유한 시스템 생성 Object Storage 네임스페이스 이름이 지정됩니다. 테넌트 네임스페이스 이름(예: axaxnpcrorw5)은 모든 영역에서 유효합니다.
- bucket_name은 데이터 및 파일을 저장하는 논리 컨테이너의 이름입니다. 버킷은 구획에서 구성되고 유지관리됩니다. 시스템 생성 버킷 이름(예: bucket-20210301-1359)에는 현재 연도, 월, 일 및 시간이 반영됩니다.
- object_name은 선택사항으로, Oracle Object Storage Cloud의 파일에 사용할 이름입니다. 객체 이름을 지정하지 않으면 파일이 원래 이름으로 복사됩니다.

자세한 내용은 *Oracle Cloud Infrastructure 설명서*의 다음 항목을 참조하십시오.

- [지역 및 가용성 도메인](#)
- [Object Storage 네임스페이스 이해](#)
- [버킷 관리](#)

예

아래 예에서 `URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET`을 이 형식의 작동하는 URL인 `https://swiftobjectstorage.region_identifier.oraclecloud.com/v1/namespace/bucket_name/`으로 바꿉니다.

- Object Storage 버킷에 스냅샷을 복사하고 이름을 바꿉니다.

```
epmautomate copyToObjectStorage "Artifact Snapshot"
oracleidentitycloudservice/jDoe example_pwd
URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET/Snapshot_04_30_21
```
- Object Storage 버킷에 파일을 복사합니다.

```
epmautomate copyToObjectStorage example_file.txt oracleidentitycloudservice/
jDoe example_pwd URL_OF_THE_ORACLE_OBJECT_STORAGE_BUCKET
```
- Object Storage 버킷에 파일을 복사하고 이름을 바꿉니다.

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

copyToSFTP

Oracle Fusion Cloud Enterprise Performance Management 환경의 위치에서 보안 FTP 서버의 위치로 파일을 복사합니다. 이 명령을 사용하면, EPM Automate 또는 REST API를 사용하여 다른 서버로 파일을 먼저 다운로드한 후 SFTP 클라이언트를 사용하여 SFTP 서버에 해당 파일을 업로드하지 않고도 환경에서 SFTP 서버로 직접 파일을 복사할 수 있습니다.

IP 허용 목록이 SFTP 서버에 대해 사용으로 설정된 경우 Cloud EPM 환경을 호스팅하는 OCI 지역의 아웃바운드 IP 주소를 SFTP 서버의 IP 허용 목록에 추가해야 합니다. OCI 지역의 아웃바운드 IP 주소는 *작업 가이드*의 Cloud EPM 데이터 센터 및 지역의 아웃바운드 IP 주소를 참조하십시오.

Note

이 명령은 포트 22에서 실행 중인 SFTP 서버만 지원합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

서비스 관리자

사용법

```
epmautomate copyToSftp SFTP_SERVER_URL EPM_FILE_NAME username=USERNAME
password=PASSWORD:
```

- *SFTP_SERVER_URL*은 파일을 복사할 디렉토리 및 파일 이름이 포함된 SFTP 서버의 URL입니다.
- *EPM_FILE_NAME*은 Cloud EPM 환경에서 복사할 파일의 이름 및 위치(기본 아웃박스가 아닌 경우)입니다. 스냅샷을 복사하는 경우 .ZIP 확장자를 지정하지 마십시오.
- *USERNAME*은 SFTP 서버에서 필요한 액세스 권한이 있는 사용자의 ID입니다.
- *PASSWORD*는 SFTP 사용자 비밀번호입니다.

예

```
epmautomate copyToSftp sftp://mySFTP_Server/someDirectory/some_file.csv
myFile.csv username=jDoe password=SftpWelcome
```

createGroups

환경에 업로드된 ANSI 또는 UTF-8 인코딩 CSV 파일을 사용하여 액세스 제어에 그룹을 추가합니다.

서비스 관리자는 [uploadFile](#) 명령을 사용하여 파일을 환경에 업로드합니다. 파일 형식은 다음과 같습니다.

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

그룹 이름에는 대/소문자가 구분되지 않습니다. 완료되면 명령이 실패한 각 항목에 대한 정보를 콘솔에 인쇄합니다. 이 정보를 검토하여 CSV 파일의 일부 항목에서 명령 실행이 실패한 이유를 알아봅니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할

사용법

epmautomate createGroups *FILE_NAME* 설명: *FILE_NAME*은 그룹 이름 및 설명을 포함하는 CSV 파일의 이름입니다.

예

```
epmautomate createGroups group_file.CSV
```

createNRSnapshot

Narrative Reporting 환경의 주문형 스냅샷인 `EPRCS_Backup.tar.gz`를 생성합니다.

[downloadFile](#) 명령을 사용하여 `EPRCS_Backup.tar.gz` 및 오류 파일을 로컬 컴퓨터에 다운로드하거나 [copyFileFromInstance](#) 명령을 사용하여 다른 환경에 복사할 수 있습니다.

`EPRCS_Backup.tar.gz`의 애플리케이션 데이터는 최종 일별 유지관리 날짜를 기준으로 합니다. 더 최근 데이터를 백업해야 하는 경우 데이터 익스포트 Narrative Reporting 기능을 사용하십시오.

적용 대상

Narrative Reporting

필수 역할

서비스 관리자

사용법

epmautomate createNRSnapshot [*errorFile=Error_File.txt*] 설명: *errorFile*은 선택사항으로, 이 명령에서 오류가 발생하는 경우 오류를 기록하기 위한 고유 텍스트 파일의 이름입니다.

예

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

createReconciliations

공용 필터를 사용하여 프로파일을 지정된 기간으로 복사합니다.

적용 대상

Account Reconciliation.

필수 역할

- 서비스 관리자
- 사전 정의된 역할(조회자 역할에는 조정 - 준비자 애플리케이션 역할이 필요함)

사용법

epmautomate createReconciliations *PERIOD SAVED_FILTER* 설명:

- *PERIOD*는 기간 이름입니다.
- *SAVED_FILTER*는 저장된 공용 필터의 이름입니다. 저장된 필터를 지정하지 않는 경우 이 명령은 공용 필터를 사용하여 모든 적용 가능한 프로파일을 복사합니다.

① 주

개인 필터의 이름을 지정하는 경우 이 명령은 필터를 무시하고 Filter not found 메시지를 표시합니다.

예

- 기간의 모든 프로파일 복사: epmautomate createReconciliations "January 2015"
- 특정 필터의 프로파일 복사: epmautomate createReconciliations "January 2015" "Corporate Recs"

deleteFile

기본 업로드 위치, 인박스 또는 아웃박스, Data Management 폴더, profitinbox/profitoutbox에서 파일 또는 스냅샷을 삭제합니다.

기본 업로드 위치가 아닌 위치에서 파일을 삭제하려면 파일 위치를 지정해야 합니다.

생성 또는 아카이브 중인 스냅샷을 삭제하기 위해 이 명령을 실행하는 경우 다음 오류 중 하나가 표시됩니다.

- 스냅샷이 생성되는 경우 File not found 오류 또는 Invalid File 오류
- 스냅샷이 아카이브되고 있는 경우 Archive process is in progress. Unable to Rename or Delete 오류

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

epmautomate deleteFile *FILE_NAME*

① 주

해당하는 경우 확장자를 포함한 파일 이름(예: data.csv, data.zip)을 지정해야 합니다. 파일 확장자(.ZIP)를 지정하지 않고 스냅샷을 삭제할 수 있습니다. 그러나 이 용법은 더 이상 사용되지 않습니다. 파일이 기본 위치에 없는 경우 파일의 위치를 지정해야 합니다. 자세한 내용은 [기본 파일 위치](#)를 참조하십시오. 지원되는 위치로는 inbox, profitinbox, outbox, profitoutbox, to_be_imported, inbox/directory_name이 있습니다.

예

- 기본 업로드 위치에서 파일을 삭제합니다.
`epmautomate deleteFile data.csv`
- 인박스에서 파일을 삭제합니다.
`epmautomate deleteFile inbox/data.csv`
- 아웃박스에서 파일을 삭제합니다.
`epmautomate deleteFile outbox/data.csv`
- 마이그레이션을 사용하여 생성한 스냅샷을 삭제합니다.
 - `epmautomate deleteFile "Backup 18-06-12.zip"` 또는
 - `epmautomate deleteFile "Backup 18-06-12"`(사용되지 않음)
- profitinbox에서 삭제합니다(Profitability and Cost Management).
`epmautomate deleteFile profitinbox/data.csv`
- profitoutbox에서 삭제합니다(Profitability and Cost Management).
`epmautomate deleteFile profitoutbox/data.csv`
- Data Management 업로드 폴더에서 삭제합니다.
`epmautomate deleteFile inbox/dm_data/data.csv`
- Data Management 폴더에서 삭제합니다.
`epmautomate deleteFile outbox/dm_data/data.csv`

deleteGroups

환경에 업로드된 ANSI 또는 UTF-8 인코딩 CSV 파일에서 사용가능한 정보를 기준으로 액세스 제어에서 그룹을 제거합니다.

서비스 관리자는 [uploadFile](#) 명령을 사용하여 파일을 환경에 업로드합니다. 파일 형식은 다음과 같습니다.

```
Group Name
Example_grp1
Example_grp2
```

파일의 그룹 이름 값은 대소문자를 구분하지 않습니다. 완료되면 명령이 실패한 각 항목에 대한 정보를 콘솔에 인쇄합니다. 이 정보를 검토하여 CSV 파일의 일부 항목에서 명령 실행이 실패한 이유를 알아봅니다.

적용 대상

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

Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할

사용법

`epmautomate deleteGroups FILE_NAME` 설명: `FILE_NAME`은 액세스 제어에서 제거할 그룹의 이름을 포함하는 CSV 파일의 이름입니다.

예

```
epmautomate deleteGroups group_file.CSV
```

deletePointOfView

Enterprise Profitability and Cost Management 애플리케이션의 POV에서 아티팩트 및 Oracle Essbase 큐브 데이터를 삭제합니다.

적용 대상

Enterprise Profitability and Cost Management

필수 역할

서비스 관리자

사용법

`epmautomate deletePointOfView POV_NAME [povDelimiter="DELIMITER"]` 설명:

- `POV_NAME`은 삭제할 POV의 이름을 나타냅니다.
- `povDelimiter`는 POV 값에서 사용되는 구분자입니다. 기본값은 `::`(이중 콜론)입니다. 이 값은 큰따옴표로 묶어야 합니다. 예: `povDelimiter="_"`. 기본값 외에는 다음 구분자만 지원됩니다. `_`(밑줄), `#`(해시), `~`(물결표), `%`(퍼센트), `;`(세미콜론), `:`(콜론), `-`(대시).

예

- 사용자정의 POV 구분자를 사용하는 POV 삭제
`epmAutomate deletePointOfView FY21_Jan_Actual_Working povDelimiter="_"`
- 기본 POV 구분자를 사용하는 POV 삭제
`epmAutomate deletePointOfView FY21::Jan::Actual::Working`

deletePOV

Profitability and Cost Management의 POV에서 모델 아티팩트 및 Oracle Essbase 큐브 데이터를 삭제합니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

① 주

대체 변수가 있으면 실제 매개변수 값 대신 대체 변수를 사용할 수 있습니다. 자세한 내용은 [일부 Enterprise Profitability and Cost Management 명령에서 대체 변수 사용](#)을 참조하십시오.

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

- `APPLICATION_NAME`은 삭제할 POV가 포함된 Profitability and Cost Management 애플리케이션의 이름입니다.
- `POV_NAME`은 삭제할 POV의 이름입니다. 이 값은 필수입니다.
- `stringDelimiter="DELIMITER"`는 POV 값에서 사용되는 구분자를 지정합니다. 구분자는 큰따옴표로 묶어야 합니다.

예

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

deployCube

Profitability and Cost Management 애플리케이션의 계산 큐브를 배포 또는 재배포합니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

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

- `APPLICATION_NAME`은 Profitability and Cost Management 애플리케이션의 이름입니다.
- `PARAMETER=VALUE`는 큐브를 배포할 런타임 매개변수와 해당 값을 나타냅니다. 프로세스에 필요한 개수만큼 매개변수와 값 쌍을 지정합니다. 적합한 매개변수 및 해당 값은 다음과 같습니다.

① 주

매개변수 값(true 또는 false)은 모두 소문자여야 합니다.

- isKeepData=true|false
기존 데이터를 보존할지 여부를 지정합니다(있는 경우).
- isReplaceCube=true|false는 기존 큐브를 대체할지 여부를 지정합니다.

① 주

isKeepData 및 isReplaceCube 값은 하나만 true로 설정될 수 있습니다.

- isRunNow=true|false는 즉시 프로세스를 실행할지 여부를 지정합니다.
- comment는 큰따옴표로 묶인 선택적 설명입니다.

예

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

deployEJTemplates

Financial Consolidation and Close에서 완성된 기업 분개 템플릿을 배포하여 기간을 개설합니다. 기업 분개 템플릿을 배포하면 선택한 기간 동안 템플릿과 연계된 반복 분개가 생성됩니다. 배포된 템플릿을 사용하여 임시 분개를 생성할 수도 있습니다.

Financial Consolidation and Close 화면 대신 이 명령을 사용하여 매월 초 새 기업 분개 템플릿을 배포할 수 있습니다.

적용 대상

재무 통합 및 마감

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

```
epmautomate deployEJTemplates YEAR PERIOD [Template=TEMPLATE_NAME]
[ResetJournals=true|false] 설명:
```

- Year는 분개 연도입니다.
- Period는 분개 기간입니다. 이 값은 연도가 지정된 경우에만 지정할 수 있습니다.
- Template=TEMPLATE_NAME은 배포할 분개의 이름입니다. 분개를 두 개 이상 배포하려면 각각 Template=TEMPLATE_NAME 형식의 고유한 템플릿 이름을 입력합니다(예: Template="Loan Details" Template="Housing Details" Template="Repayment Details"). 이 매개변수 값을 지정하지 않으면 명령은 지정한 연도 및 기간 조합의 모든 템플릿을 배포합니다.
- ResetJournals는 선택사항으로, 템플릿 재배포 후 모든 분개가 첫번째 단계로 재설정되어야 하는지 여부를 나타냅니다. 기본값은 false입니다. Financial Consolidation and Close는 템플릿에 대한 변경사항을 기반으로 내부적으로 이 값을 검증하고 필요한 경우 지정한 값을 대체할 수 있습니다.

예

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

deployFormTemplates

완료된 양식 템플릿을 새 데이터 수집 기간에 배포하여 보충 데이터 양식 템플릿을 생성함으로써 일관성 있고 반복 가능한 데이터 수집 프로세스를 보장합니다.

적용 대상

Financial Consolidation and Close, Tax Reporting.

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

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

- COLLECTION_INTERVAL은 템플릿이 배포되는 수집 간격의 이름입니다.
- DIMENSION은 데이터 수집 프로세스의 빈도 차원을 DIMENSION=MEMBER_NAME 형식으로 지정합니다(선택사항). 수집 간격에서 정의된 개수만큼 차원을 지정합니다(연도 및 기간을 포함하여 최대 4개)(예: "Year=2020" "Period=July" "Product=Oracle EPM" "Consolidation=entity Input"). 이 매개변수 값을 지정하지 않은 경우 기본값이 사용되지 않습니다.
- Template는 배포할 양식 템플릿의 고유 이름을 Template=TEMPLATE_NAME 형식으로 확인합니다(선택사항). 이 형식으로 고유 이름을 필요한 개수만큼 지정할 수 있습니다. 예: Template="Loan Details Template" Template="Housing Details Template" Template="Repayment Details Template" 이 등록정보 값을 지정하지 않으면 명령은 지정한 간격의 모든 템플릿을 배포합니다.
- resetWorkFlows는 재배포 후에 모든 양식을 첫번째 단계로 재설정할지 여부를 나타냅니다(선택사항). 기본값은 false입니다.

예

```
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

태스크 관리자 템플릿에서 태스크 스케줄로 태스크를 배포하여 반복 비즈니스 프로세스가 일관성 있게 실행되도록 합니다.

적용 대상

Enterprise Profitability and Cost Management, Financial Consolidation and Close, Tax Reporting

필수 역할

서비스 관리자

사용법

```
epmAutomate deployTaskManagerTemplate TEMPLATE_NAME SCHEDULE_NAME YEAR PERIOD  
DAY_ZERO_DATE [dateFormat=DATE_FORMAT] [orgUnit=ORGANIZATION UNIT] 설명:
```

- *TEMPLATE_NAME*: 배포할 태스크 관리자 템플리트의 이름입니다.
- *SCHEDULE_NAME*: 템플리트에서 생성할 스케줄 이름입니다.
- *YEAR*: 템플리트를 배포할 연도 차원 멤버입니다.
- *PERIOD*: 템플리트를 배포할 기간 차원 멤버입니다.
- *DAY_ZERO_DATE*: 스케줄을 생성하는 데 사용되는 적합한 형식의 0일 날짜입니다.
- dateFormat: 선택사항으로, 0일 날짜의 날짜 형식입니다. 기본 형식은 YYYY-MM-DD입니다.
- orgUnit: 선택사항으로, 조직 단위 이름입니다. 값을 지정하지 않으면 표준 날짜 매핑을 사용하여 스케줄이 생성됩니다. 휴일 규칙은 사용되지 않습니다.

예

- 0일 날짜에 기본 날짜 형식(YYYY-MM-DD)을 사용하여 *Ind* 조직 단위의 태스크 관리자 템플리트 배포:

```
epmautomate deployTaskManagerTemplate "Vision Monthly Close" "Qtr 2 Close"  
2021 July 2021-07-10 orgUnit=Ind
```

- 0일 날짜에 dd/mm/yyyy를 날짜 형식으로 사용하여 *Ind* 조직 단위의 태스크 관리자 템플리트 배포:

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

dismissIPMInsights

새 IPM(지능형 성과 관리) 인사이트 작업을 실행하기 전에 IPM 인사이트 데이터 제거를 자동화합니다. 데이터를 제거하면 작업할 계획이 없는 모든 열려 있는 인사이트가 닫힙니다. 이 명령은 IPM 인사이트 대시보드를 사용하여 데이터를 수동으로 제거하는 데 대한 대안이 됩니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

```
epmautomate dismissIPMInsights [comment="comment"] 설명: 설명(선택사항)은 열려 있는  
인사이트를 제거하는 근거입니다.
```

예

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

downloadFile

환경에서 로컬 컴퓨터로 파일을 다운로드합니다.

이 명령을 사용하여 로컬 저장영역의 데이터, 메타데이터, 백업 스냅샷을 다운로드할 수 있습니다. EPM Automate가 실행되는 폴더로 파일이 다운로드됩니다.

환경의 스냅샷이 생성되는 동안(예: 일별 유지관리 중) 이 명령을 실행하여 현재 스냅샷을 다운로드하는 경우 File not found 또는 Invalid File 오류가 표시됩니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

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



팁

[listFiles](#) 명령을 사용하여 다운로드할 파일이 있는 위치를 확인합니다.

예

- 유지관리 스냅샷 다운로드: `epmautomate downloadFile "Artifact Snapshot"`
- 사용자정의 스냅샷 다운로드: `epmautomate downloadFile "mySnapshot.zip"`
- Narrative Reporting 유지관리 스냅샷 다운로드: `epmautomate downloadFile "EPRCS_Backup.tar.gz"`
- 기본 다운로드 위치에서 파일 다운로드: `epmautomate downloadFile data.csv`
- Data Management 폴더에서 다운로드: `epmautomate downloadfile outbox/dm_data/data.csv`
- profitoutbox에서 다운로드: `epmautomate downloadFile profitOutbox/data.csv`
- Account Reconciliation 환경의 `apr/2025-06-01` 폴더에서 활동 보고서 `2025-06-01_20_38_43.html`을 다운로드합니다.
`epmautomate downloadfile "apr/2025-06-01_20_38_43/2025-06-01_20_38_43.html"`

enableApp

애플리케이션을 사용으로 설정합니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

`epmautomate enableapp APPLICATION_NAME` 설명: `APPLICATION_NAME`은 사용으로 설정할 Profitability and Cost Management 애플리케이션의 이름입니다.

예

```
epmautomate enableApp BksML12
```

enableQueryTracking

ASO 큐브에 대한 쿼리 추적을 사용으로 설정하여 사용자 데이터 검색 패턴(쿼리) 캡처를 시작합니다.

캡처된 데이터 검색 패턴을 사용하여 ASO 큐브 집계를 최적화하며 이 작업은 [executeAggregationProcess](#) 명령을 사용하여 시작됩니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

`epmautomate enableQueryTracking ASO_CUBE_NAME` 설명: `ASO_CUBE_NAME`은 쿼리 추적을 활성화할 ASO 큐브의 이름입니다.

예

```
epmautomate enableQueryTracking VISION_ASO
```

encrypt

고급 암호화 표준(AES/CBC/PKCS5Padding(256))을 사용하여 Oracle Fusion Cloud Enterprise Performance Management 비밀번호(또는 환경에 액세스하는 데 사용되는 OAuth2.0 새로고침 토큰 및 클라이언트 ID) 및 선택적으로 환경에 로그인하는 데 사용되는 인터넷 프록시 서버 비밀번호를 암호화하고 비밀번호 파일에 저장합니다.

비밀을 암호화하는 경우 서비스 관리자는 EPM Automate 스크립트를 작성하는 개발자가 스크립트를 실행할 수 있도록 암호화된 비밀번호 파일을 공유할 수 있습니다. 이렇게 하면 서비스 관리자 비밀번호를 공유하거나 스크립트 실행을 위해 일반적인 공유 Cloud EPM 계정을 생성할 필요가 없습니다.

비밀번호 암호화는 한 번만 처리하면 됩니다.

① 주

특수 문자가 포함된 비밀번호 암호화에 대한 자세한 내용은 [특수 문자 처리](#)를 참조하십시오.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사용법

```
epmautomate encrypt PASSWORD/REFRESH_TOKEN KEY PASSWORD_FILE [ClientID=CLIENT_ID]
[ProxyServerPassword=PROXY_PASSWORD] 설명:
```

- *PASSWORD/REFRESH_TOKEN PASSWORD*는 암호화할 비밀번호 또는 OAuth 새로고침 토큰입니다. 회사 인증서는 EPM Automate에서 사용할 수 없습니다.
- *KEY*는 비밀번호를 암호화하는 데 사용하는 개인 키입니다.
- *PASSWORD_FILE*은 암호화된 비밀번호 또는 새로고침 토큰을 저장하는 파일의 이름과 위치입니다. 비밀번호 파일은 .epw 확장자를 사용해야 합니다.
- ClientID(선택사항)는 OAuth 2.0 설정 중에 생성되는 클라이언트 ID입니다. OAuth 2.0 새로고침 토큰을 암호화하는 중에 이 값을 지정해야 합니다. 비밀번호를 암호화하는 중에는 이 값을 지정하지 마십시오.
- ProxyServerPassword는 HTTP 프록시 서버에 사용자를 인증할 비밀번호입니다. 네트워크에 대해 프록시 서버 인증을 사용으로 설정한 경우에만 필요합니다.

예

- **Cloud EPM 비밀번호만 암호화:** `epmautomate encrypt P@ssword1 myKey C:\mySecuredir\password.epw`
- **Cloud EPM 및 인터넷 프록시 서버 비밀번호 암호화:** `epmautomate encrypt E@example1 myKey C:\mySecuredir\password.epw ProxyServerPassword=Proxy_Pwd1`
- **새로고침 토큰 및 클라이언트 ID를 암호화합니다.** `epmautomate encrypt AAyyilyBAWD4...FVkxefd8kjoJr6HJPA= myEncryption42Key C:\mySecuredir\oauthfile1.epw ClientID=6fdf2e72fd343430ABR22394C`

essbaseBlockAnalysisReport

애플리케이션에서 블록 저장영역 옵션(BSO) 큐브(일반적으로 계산에 사용됨)를 조정할 수 있도록 Oracle Essbase 데이터를 분석하는 데 도움이 되는 Essbase 블록 분석 보고서를 생성합니다.

Essbase 블록 분석 보고서는 Essbase BSO 큐브의 반복되는 숫자 등 데이터 패턴으로 인해 발생하는 성능 이슈를 해결하는 데 유용합니다. 이 보고서는 다음 세 가지 영역에 대한 정보를 제공합니다.

1. **0만 포함된 블록의 퍼센트**는 익스포트 파일에 포함된 모든 블록 중 0만 포함된 블록의 퍼센트를 보여줍니다.
2. **숫자 셀 비율에 따른 상위 10개 반복 숫자 셀 값** 이 테이블은 익스포트 파일에 있는 모든 값의 퍼센트로 상위 10개 반복 값을 보여줍니다.
3. **반복 값이 있는 상위 100개 밀집 멤버 조합** 이 테이블은 큐브에서 반복 값이 있는 상위 100개 밀집 조합을 보여줍니다. **셀 값 열**은 계층에 나타나는 순서대로 각 멤버의 값을 다른 열로 표시합니다. 예를 들어, 기간이 열 전체에 걸쳐 있는 경우 1월, 2월 등에 대한 열이 다르게 표시됩니다. 다른 밀집 차원은 행에 나타납니다. 이는 반복 값의 위치를 확인하는 데 도움이 됩니다.

이 보고서를 실행하기 전에 [exportEssbaseData](#) 명령을 사용하여 블록 분석 보고서를 생성할 큐브의 데이터를 zip 파일로 익스포트합니다. 필요에 따라 level0 또는 모든 데이터를 익스포트할 수 있습니다. 이 명령을 실행하여 이 zip 파일에 대한 블록 분석 보고서를 생성할 수 있습니다. 보고서는 아웃박스에서 생성됩니다. [downloadFile](#) 명령을 사용하여 보고서를 로컬 컴퓨터에 다운로드하거나 [sendMail](#) 명령을 사용하여 전자메일로 보낼 수 있습니다.

적용 대상

Financial Consolidation and Close, Planning, Planning Modules, FreeForm, Strategic Workforce Planning, Sales Planning, Tax Reporting.

필수 역할

서비스 관리자

사용법

```
epmautomate essbaseBlockAnalysisReport EXPORT_DATA_FILE.zip REPORT_FILE
```

- EXPORT_DATA_FILE.zip은 [exportEssbaseData](#) 명령을 사용하여 이전에 BSO 큐브에서 익스포트한 Essbase 데이터가 포함된 zip 파일의 이름입니다.
- REPORT_FILE은 HTML 형식의 블록 분석 보고서 파일의 이름입니다.

예

```
epmautomate essbaseBlockAnalysisReport Plan1_cube_data.zip block_report.html
```

executeAggregationProcess

ASO 큐브의 성능을 높이도록 선택적으로 쿼리 추적 통계를 사용하여 집계 프로세스를 시작합니다. ASO 큐브 최적화에서 중요한 단계입니다.

이 명령을 실행하기 전에 다음을 수행합니다.

- [enableQueryTracking](#) 명령을 사용하여 ASO 집계를 최적화하는 데이터 검색 통계를 캡처합니다.
- 비즈니스 프로세스가 집계 뷰를 생성하는 데 사용할 수 있는 사용자 데이터 검색 패턴(쿼리)을 캡처하는 시간을 충분히 허용합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

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

- useQueryData는 쿼리 추적을 통해 수집되어 기록된 쿼리 데이터를 사용하여 가장 적합한 집계 뷰 세트를 선택합니다. 기본값은 false입니다.
- includeAlternateRollups는 뷰 선택 프로세스에 보조 계층(기본 레벨 사용 포함)을 포함합니다. 기본값은 disable입니다.
- growthSizeRatio는 선택사항으로, 서버에서 선택한 뷰를 집계하기 위한 최대 큐브 증가 비율입니다. 지정한 최대 증가 비율에 도달하면 큐브 증가가 중지됩니다. 기본 설정을 사용하면 증가 비율 한도 없이 큐브가 증가할 수 있습니다.

Note

기본 집계 뷰를 생성하려면 선택적 매개변수를 지정하지 않고 이 명령을 실행하십시오.

예

- [enableQueryTracking](#) 명령을 사용하여 캡처된 쿼리 데이터를 기반으로 집계 뷰 생성:

```
epmautomate executeAggregationProcess VISION_ASO useQueryData=true
includeAlternateRollups=enable
```
- 기본 집계 뷰 생성:

```
epmautomate executeAggregationProcess Vis1ASO
```

executeBurstDefinition

하나의 데이터 소스에서 단일 차원 멤버 둘 이상의 보고서 또는 장부를 실행하는 데 필요한 아티팩트와 POV, 기타 설정을 지정하는 버스팅 정의를 실행합니다.

적용 대상

Narrative Reporting

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

고급 사용자, 사용자, 조회자 역할의 사용자에게는 ACL을 통해 추가 보안이 지정되어야 합니다.

사용법

```
epmAutomate executeBurstDefinition ARTIFACT_NAME 설명: ARTIFACT_NAME은 버스팅 정의
경로 및 이름입니다.
```

예

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

executeReportBurstingDefinition

버스팅 정의를 사용하여 단일 차원의 둘 이상 멤버에 대해 단일 보고서 또는 장부의 버스팅을 실행하고 각 멤버의 PDF 또는 정적(Oracle Smart View for Office에서 새로 고칠 수 없음) Excel 출력을 게시합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

```
epmAutomate executeReportBurstingDefinition BURST_DEFINITION_NAME  
[jobName=JOB_NAME] 설명:
```

- *BURST_DEFINITION_NAME*은 버스팅 정의의 경로 및 이름입니다.
- *JOB_NAME*은 선택사항으로, 버스팅 정의를 실행하는 데 사용해야 하는 작업의 이름입니다. 기본값은 Execute Bursting Definition입니다.

예

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

exportAccessControl

환경에 사전 정의된 역할이 있는 사용자에게 대한 정보가 있으며 각 사용자 속성(예: 이름 및 전자메일) 및 액세스 정보(예: 그룹, 팀, 조직에 대한 지정)가 나열된 사용자 세부정보 보고서를 CSV 또는 XLS 파일로 익스포트합니다.

다음은 샘플 보고서입니다.

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

[downloadFile](#) 명령을 사용하여 이 보고서를 다운로드할 수 있습니다.

적용 대상

Account Reconciliation

필수 역할

서비스 관리자

사용법

```
epmAutomate exportAccessControl REPORT_NAME [reportFormat=XLS|CSV] 설명:
```

- *REPORT_NAME*은 보고서가 포함될 익스포트 파일의 이름입니다.
- *reportFormat*(선택사항)은 파일 형식입니다. 적합한 값은 XLS 및 CSV(기본값)입니다.

예

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

exportAppAudit

데이터 감사 레코드를 ZIP 파일로 익스포트하여 로컬 컴퓨터에서 다운로드하고 아카이브할 수 있습니다. 감사 정보는 최대 365일간 환경에서 사용할 수 있습니다. 특정 일수 또는 날짜 범위에 대한 보고서를 생성할 수 있습니다.

출력 CSV 파일의 첫번째 문자는 BOM(바이트 순서 표시) 문자 `\ufeff`이며, 큰따옴표로 묶인 암호화된 애플리케이션 ID가 뒤에 옵니다. CSV 파일 머리글은 애플리케이션 ID를 따릅니다.

적용 대상

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

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 감사 - 관리 애플리케이션 역할
- 임의 사전 정의된 역할 및 감사 - 보기 애플리케이션 역할

사용법

```
epmautomate exportAppAudit EXPORT_FILE_NAME [userNames=USER_NAMES]
[nDays=Number_of_Days] [startDate=START_DATE endDate=END_DATE]
[excludeApplicationId=true|false]:
```

- *EXPORT_FILE_NAME*은 익스포트한 감사 데이터를 저장할 ZIP 파일의 이름입니다. [downloadFile](#) 명령을 사용하여 환경에서 파일을 다운로드합니다.
- 선택사항인 *userNames*는 심표로 구분된 사용자 로그인 이름 목록입니다. 지정하면 해당 사용자가 생성한 감사 데이터만 익스포트됩니다. 모든 사용자의 감사 데이터를 익스포트하려면 이 값을 지정하지 마십시오.
- 선택사항인 *nDays*는 감사 레코드를 익스포트할 일수를 확인합니다. 기본값은 7일입니다. 가능한 값은 최근 365일 동안의 사용가능한 감사 데이터를 익스포트하는 all, 1, 2, 7, 30, 60, 180이 있습니다. 특정 날짜 범위에 대한 보고서를 생성하는 경우 이 값을 지정하지 마십시오.
- *startDate* 및 *enddate*는 선택사항으로, 감사 레코드를 익스포트할 시작 및 종료 날짜를 확인합니다. 이러한 값은 YYYY-MM-DD 형식으로 지정해야 합니다.
 - 시작 날짜는 종료 날짜 이전이어야 합니다. 보고서를 생성하려면 시작 날짜 및 종료 날짜를 모두 지정해야 합니다.

- 이러한 날짜는 미래 날짜일 수 없습니다.
- 선택사항인 `nDays` 값이 지정된 경우 이러한 날짜는 무시됩니다.
- `excludeApplicationId`: 선택사항으로, 애플리케이션 ID를 익스포트 파일에 기록할 것인지 여부를 나타냅니다. 기본값은 `false`입니다.

① 주

애플리케이션 ID가 포함되지 않은 익스포트된 파일의 데이터는 Oracle Fusion Cloud Enterprise Performance Management 환경으로 임포트할 수 없습니다.

예

- 애플리케이션 ID가 있는 상태로 지난 30일 동안 지정된 사용자의 감사 데이터 익스포트:
`epmautomate exportAppAudit auditData userNames=johnDoe,jane.doe@example.com ndays=30`
- 애플리케이션 ID 없이 지난 30일 동안 지정된 사용자의 감사 데이터 익스포트:
`epmautomate exportAppAudit auditData userNames=johnDoe,jane.doe@example.com ndays=30 excludeApplicationId=true`
- 애플리케이션 ID가 있는 상태로 모든 사용자의 2024년 8월 감사 데이터 익스포트:
`epmautomate exportAppAudit auditData startDate=2024-08-01 endDate=2024-08-31`

exportAppSecurity

로컬 저장영역으로 다운로드할 수 있는 CSV 파일로 아티팩트 레벨 액세스 지정(ACL)을 익스포트합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

`epmautomate exportAppSecurity EXPORT_FILE_NAME.CSV` 설명: `EXPORT_FILE_NAME`은 익스포트한 보안 데이터를 저장할 파일의 이름입니다. 이 파일은 아웃박스에 생성되며, 아웃박스에서 사용자 컴퓨터로 다운로드할 수 있습니다.

예

```
epmautomate exportAppSecurity app_security.CSV
```

exportARApplicationProperties

Account Reconciliation 애플리케이션 설정(Redwood 환경, 테마, 전자메일 통지, 비즈니스 프로세스 이름 관련 설정), 배경 이미지, 로고 이미지를 JSON 파일로 익스포트하여 동일한 환경 또는 다른 환경으로 임포트할 수 있습니다.

이 명령은 프로덕션 환경에서 테스트 환경으로 애플리케이션을 임포트하는 경우 유용합니다. 프로덕션 및 테스트 환경에서 애플리케이션 설정이 서로 다른 경우 테스트 환경에서 애플리케이션

설정을 익스포트한 후 프로덕션 환경에서 애플리케이션을 임포트하고 테스트 환경으로 설정을 임포트하여 원래 설정을 유지관리할 수 있습니다.

적용 대상

Account Reconciliation

필수 역할

서비스 관리자

사용법

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

- *FILE_NAME*은 익스포트된 등록정보 값을 저장할 JSON 파일의 이름입니다. [downloadFile](#) 명령을 사용하여 익스포트 파일을 다운로드할 수 있습니다. [uploadFile](#) 명령을 사용하여 타겟 환경에 업로드한 후 [importARApplicationProperties](#) 명령을 실행하여 타겟 환경에서 이러한 설정을 복원하십시오.
- Properties(선택사항)는 익스포트할 등록정보의 심표로 구분된 목록입니다. 다음 등록정보 중 일부 또는 전부를 익스포트할 수 있습니다. 이 등록정보를 생략하면 다음 등록정보를 모두 익스포트합니다.
 - Theme: 환경에 사용된 표시 테마를 익스포트합니다.
 - EmailNotification: 환경에 정의된 전자메일 통지 설정을 익스포트합니다.
 - DisplayBusinessProcessName: 환경의 페이지에 비즈니스 프로세스 이름을 표시할지 여부를 익스포트합니다.
 - RedwoodExperience: 환경의 Redwood 환경 설정을 익스포트합니다.
 - BackgroundImage: 환경에 사용된 배경 이미지를 익스포트합니다.
 - LogoImage: 환경에 사용된 로고 이미지를 익스포트합니다.

예

다음 예에서는 환경의 전자메일 통지, Redwood 환경 설정, 로고 이미지만 익스포트합니다.

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

exportBackgroundImage

Account Reconciliation 환경에 사용된 배경 이미지를 JPG 파일로 익스포트하여 다른 환경으로 임포트할 수 있도록 합니다.

적용 대상

Account Reconciliation

필수 역할

서비스 관리자

사용법

```
epmautomate exportBackgroundImage IMAGE_NAME.jpg, 설명: IMAGE_NAME은 배경 이미지
파일의 이름입니다.
```

[downloadFile](#) 명령을 사용하여 이미지 파일을 다운로드할 수 있습니다. [uploadFile](#) 명령을 사용하여 타겟 환경에 업로드한 후 [importBackgroundImage](#) 명령을 실행하여 임포트하십시오.

예

```
epmautomate exportBackgroundImage corpImage.jpg
```

exportCellLevelSecurity

비즈니스 프로세스의 셀 레벨 보안 설정을 ZIP 파일로 익스포트하면 [downloadFile](#) 명령을 사용하여 로컬 컴퓨터로 다운로드할 수 있습니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

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

설명:

- `FILE_NAME`은 셀 레벨 보안 정보가 포함된 Excel 파일을 보관하기 위해 생성할 ZIP 파일의 이름입니다.
- `names`는 선택사항으로, 애플리케이션에 있는 셀 레벨 보안 정의의 심표로 구분된 목록을 나타냅니다. 이 옵션을 제공하지 않으면 애플리케이션에 있는 모든 셀 레벨 보안 정의가 익스포트됩니다.

예

- **특정 셀 레벨 보안 정의를 익스포트합니다.**

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

- **모든 셀 레벨 보안 정의를 익스포트합니다.**

```
epmautomate exportCellLevelSecurity ExportCLSDRecordsFile.zip
```

exportConsolidationJournals

Financial Consolidation and Close에 정의된 작업을 사용하여 연결 분개를 익스포트합니다.

적용 대상

재무 통합 및 마감

필수 역할

서비스 관리자

사용법

```
epmautomate exportConsolidationJournals jobName [fileName=FILE_NAME] 설명
```

- `jobName`은 Financial Consolidation and Close에서 생성된 분개 익스포트 작업의 이름입니다.

- `fileName`은 선택사항으로, 분개 항목을 익스포트할 .JLF 파일의 이름입니다. [downloadFile](#) 명령을 사용하여 이 파일을 로컬 컴퓨터로 다운로드할 수 있습니다.

예

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

exportData

`export data` 유형의 작업에 지정된 데이터 익스포트 설정(파일 이름 포함)을 사용하여 애플리케이션 데이터를 ZIP 파일에 내보냅니다.

내보낸 데이터 파일은 컴퓨터에 다운로드할 수 있는 기본 다운로드 위치에 저장됩니다. 내보낸 파일의 세부정보를 보려면 인박스/아웃박스 탐색기를 사용합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

`epmautomate exportData JOB_NAME [FILE_NAME]` 설명: `JOB_NAME`은 애플리케이션에서 정의된 작업의 이름이고 `FILE_NAME`은 데이터를 익스포트할 ZIP 파일(선택사항)의 이름입니다.

예

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

exportDataManagement

환경의 Data Management 레코드를 ZIP 파일로 익스포트합니다.

이 명령은 참조 무결성 손실 없이 데이터를 임포트할 수 있도록 ID 열을 포함한 전체 설정 및 스테이지 테이블 데이터 세트를 ZIP 파일로 익스포트합니다.

익스포트된 파일(예: `dataFile.zip`)은 아웃박스에 저장됩니다. [downloadFile](#) 명령을 사용하여 익스포트된 파일(예: `epmAutomate downloadFile outbox/dataFile.zip`)을 다운로드할 수 있습니다. 이 ZIP 파일을 사용하면 [importDataManagement](#) 명령을 사용하여 데이터를 임포트할 수 있습니다.

적용 대상

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

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

`epmautomate exportDataManagement FILE_NAME.zip` 설명: `FILE_NAME`은 데이터를 익스포트할 ZIP 파일의 이름입니다.

예

```
epmautomate exportDataManagement dataFile.zip
```

exportDimension

Oracle Fusion Cloud Enterprise Data Management 애플리케이션의 차원을 준비 영역의 파일 또는 선택적으로 연결에 정의된 타겟 환경으로 익스포트합니다.

적용 대상

Cloud EDM

필수 역할

- 서비스 관리자
- 사용자(데이터 관리자 권한 필요)

사용법

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

설명:

- `APPLICATION`은 Cloud EDM 애플리케이션의 이름입니다.
- `DIMENSION`은 애플리케이션 차원의 이름입니다.
- `FILE_NAME` 은 익스포트한 매핑을 저장할 파일(파일로 익스포트할 CSV 또는 Oracle Financials Cloud로 익스포트할 ZIP)의 이름입니다. `connection` 매개변수 값을 설정하지 않으면 이 파일은 준비 영역에 생성됩니다. [downloadFile](#) 명령을 사용하여 로컬 컴퓨터로 다운로드하거나 [copyFileFromInstance](#) 명령을 사용하여 다른 Oracle Fusion Cloud Enterprise Performance Management 환경에 복사할 수 있습니다.
- `connection=NAME`은 선택사항으로, Cloud EDM에 정의된 연결 이름(인스턴스 위치)을 확인합니다. 지정되는 경우, 익스포트 파일이 타겟 환경(Cloud EPM의 경우 인박스, Oracle Financials Cloud의 경우 기본 업로드 위치)으로 업로드됩니다.

① 주

연결 정의에 지정된 인증서에는 타겟 환경에 쓸 수 있는 액세스 권한이 있어야 합니다.

예

- Cloud EDM 준비 영역으로 익스포트: `epmautomate exportDimension USOperations Entity EntityData.CSV`
- Oracle Financials Cloud로 익스포트 및 업로드: `epmautomate exportDimension USOperations Entity EntityData.zip Connection=ora_fusion_gl`
- 타겟 Cloud EPM 인박스로 익스포트 및 업로드: `epmautomate exportDimension USOperations Entity EntityData.CSV Connection=EPM_cloud_pln`

exportDimensionMapping

위치에 대한 특정 Oracle Fusion Cloud Enterprise Data Management 차원의 매핑 규칙을 익스포트하여 매핑 규칙 파일을 생성하고, 선택적으로 익스포트된 파일을 다른 Oracle Fusion Cloud Enterprise Performance Management 환경의 Data Management 인박스로 업로드합니다.

적용 대상

Cloud EDM

필수 역할

- 서비스 관리자
- 데이터 관리자 권한이 있는 사용자

사용법

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

- APPLICATION은 Cloud EDM 애플리케이션의 이름입니다.
- DIMENSION은 애플리케이션 차원의 이름입니다.
- LOCATION은 매핑 규칙을 익스포트해야 하는 특정 위치입니다.
- FILE_NAME은 익스포트한 매핑을 저장할 CSV 파일의 이름입니다. connection 매개변수를 설정하지 않으면 이 파일은 준비 영역에 생성됩니다. [downloadFile](#) 명령을 사용하여 로컬 컴퓨터로 다운로드하거나 [copyFileFromInstance](#) 명령을 사용하여 파일을 다른 Cloud EPM 환경에 복사할 수 있습니다.
- connection=NAME은 선택사항으로, Cloud EDM에 정의된 연결 이름(인스턴스 위치)을 확인합니다. 지정되는 경우, 이 명령은 익스포트된 파일을 타겟 환경의 기본 업로드 위치로 업로드합니다.

① 주

연결에 지정된 인증서에는 타겟 환경에 쓸 수 있는 액세스 권한이 있어야 합니다.

예

- 준비 영역으로 익스포트: `epmautomate exportDimensionMapping USOperations Entity Loc1 Loc1Mappings.CSV`
- 타겟 Cloud EPM 환경의 인박스로 익스포트 및 업로드: `epmautomate exportDimensionMapping USOperations Entity Loc1 Loc1Mappings.CSV Connection=EPM_cloud_pln`

exportEJournals

Financial Consolidation and Close에서 전기할 준비가 된 기업 분개를 ZIP 파일로 익스포트합니다. 그런 다음, 이 파일을 사용하여 분개 데이터를 ERP 시스템으로 임포트할 수 있습니다.

이 명령은 익스포트 파일로 분개를 익스포트한 후에 익스포트된 각 분개의 전기 상태를 Ready To Post에서 Post In Progress로 업데이트합니다.

적용 대상

재무 통합 및 마감

필수 역할

서비스 관리자

사용법

```
epmautomate exportEJournals FILE_NAME.zip [year=YEAR] [period=PERIOD]
[OPERATION=posting|validation] 설명:
```

- **FILE_NAME:** 분개 익스포트 CSV 파일을 아카이브할 ZIP 파일을 나타냅니다. 이 명령은 각 분개에 대해 하나의 CSV 파일(이름 형식은 YEAR_PERIOD_JOURNALID_YYYYDDMMHHMSS.csv)을 생성한 후 해당 파일을 압축하여 이 ZIP 파일을 생성합니다.
- **YEAR:** 선택사항으로, 분개 데이터를 익스포트할 데이터 수집 연도입니다. 지정하지 않으면 모든 연도의 데이터가 익스포트됩니다.
- **PERIOD:** 선택사항으로, 분개 데이터를 익스포트할 데이터 수집 기간입니다. 데이터 수집 연도가 지정된 경우에만 설정할 수 있습니다. 값을 지정하지 않으면 모든 기간의 데이터가 익스포트됩니다.

① Note

YEAR 및 PERIOD를 지정하지 않으면 이 명령은 모든 연도 및 기간에 걸친 Ready To Post 전기 상태의 분개를 모두 익스포트합니다.

- Operation은 선택사항으로, 수행할 작업입니다. 가능한 값은 posting 및 validation입니다. 기본값은 posting입니다.
operation=posting은 연도, 기간, 분개 ID, 전기 상태(전기됨/실패), 메시지, 오류 코드 및 오류 메시지 열이 사용된 분개 전기 결과가 포함되어 있는 CSV 파일을 제공합니다. 메시지, 오류 코드, 오류 메시지 열은 선택사항입니다. 파일에 제공된 상태를 기준으로 분개 전기 상태가 업데이트됩니다. 전기 상태가 *전기 진행* 중인 분개만 업데이트됩니다.
operation = validation은 연도, 기간, 분개 ID, 검증 상태(적합/실패), 메시지, 오류 코드 및 오류 메시지 열이 사용된 분개 검증 결과가 포함되어 있는 CSV 파일을 생성합니다. 메시지, 오류 코드, 오류 메시지 열은 선택사항입니다. 파일에 제공된 상태를 기준으로 분개 검증 상태가 업데이트됩니다. 검증 상태가 *검증 진행* 중인 분개만 업데이트됩니다.

예

- 모든 연도 및 기간의 분개 데이터 익스포트:
epmautomate exportEJournals Journal_Export.zip
- 특정 연도의 분개 데이터 익스포트:
epmautomate exportEJournals Journal_Export.zip year=2020
- 특정 연도 및 기간 조합의 분개 데이터 익스포트:
epmautomate exportEJournals Journal_Export.zip year=2024 period=March
- 특정 연도 및 기간 조합의 분개 데이터 익스포트, 검증 작업 수행:
epmautomate exportEJournals Journal_Export.zip year=2024 period=March
Operation=validation

exportEssbaseData

애플리케이션 큐브에서 아카이브로 데이터를 익스포트합니다. 레벨 0 데이터(ASO 및 BSO 큐브)만 익스포트하거나 큐브의 모든 데이터(BSO 큐브)를 익스포트할 수 있습니다.

익스포트한 아카이브를 사용하여 Oracle Essbase 데이터에서 패턴(예: 성능 향상에 도움이 되는 패턴)을 분석합니다.

① Note

- 아카이브에 포함된 데이터 파일의 확장자는 .txt입니다.
- 스마트 목록 및 셀 텍스트 객체는 UUID로 익스포트됩니다.

데이터 익스포트를 실행하면 큐브가 읽기 전용 모드로 전환되고 익스포트 작업이 진행되는 동안에는 모든 쓰기 활동이 금지됩니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

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

- `CUBE_NAME`은 데이터를 익스포트할 큐브를 확인합니다.
- `FILE_NAME`은 익스포트한 데이터를 포함할 ZIP 파일의 이름입니다. [downloadFile](#) 명령을 실행하여 이 아카이브를 다운로드할 수 있습니다.
- `level`은 선택사항으로, 익스포트할 데이터 레벨을 확인합니다. 기본값은 0입니다.
 - **ASO 큐브:** 레벨 0 데이터를 익스포트하려면 0을 지정합니다. All 옵션을 사용할 수 없습니다.
 - **BSO 큐브:** 레벨 0 데이터를 익스포트하려면 0을 지정하고 모든 데이터를 익스포트하려면 All을 지정합니다.

예

- BSO 큐브에서 모든 데이터 익스포트:
`epmautomate exportEssbaseData Report1 Report1_all_data.zip level=All`
- 큐브에서 레벨 0 데이터 익스포트:
`epmautomate exportEssbaseData Plan1 Plan1_lvl0_data.zip`

exportJobConsole

작업 콘솔 레코드를 CSV 파일로 익스포트하고 익스포트 ZIP 파일을 생성합니다.

출력 CSV 파일의 첫번째 문자는 BOM(바이트 순서 표시) 문자 `\uffeff`이며, 큰따옴표로 묶인 암호화된 애플리케이션 ID가 뒤에 옵니다. CSV 파일 머릿글은 애플리케이션 ID를 따릅니다.

적용 대상

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

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 작업 - 관리 애플리케이션 역할
- 임의 사전 정의된 역할 및 작업 - 보기 애플리케이션 역할

사용법

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

- *FILE_NAME*: 익스포트된 작업 콘솔 레코드를 저장할 ZIP 파일의 이름입니다. [downloadFile](#) 명령을 사용하여 환경에서 이 파일을 다운로드합니다.
- *nDays*: 선택사항으로, 작업 콘솔 레코드를 익스포트할 일수를 나타냅니다. 가능한 값은 모든 사용가능한 작업 콘솔 레코드를 익스포트하는 all(모두 소문자)과 1, 2, 7, 30, 60입니다. 기본값은 7입니다.
- *jobTypes*는 선택사항으로, 익스포트할 콘솔 레코드에 대한 작업 코드의 심표로 구분된 목록입니다. 기본값은 Rules입니다. 적합한 값은 다음과 같습니다.
 - all(모두 소문자)
 - 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**: 선택사항으로, 레코드를 익스포트할 작업 상태 코드의 심표로 구분된 목록입니다. 기본값은 2(성공적으로 완료됨)입니다. 가능한 값은 다음과 같습니다.
 - all(모두 소문자) - 모든 상태의 모든 작업인 경우 사용
 - 1 - 처리 중
 - 2 - 성공적으로 완료됨
 - 3 - 실패했으며 오류가 있음
 - 4 - 완료되었으나 상태를 알 수 없음
 - 5 - 완료되었으나 임계값 위반 상태임
 - 6 - 취소 대기 중
 - 7 - 취소됨
 - 8 - 완료되었으나 오류가 있음
 - 9 - 완료되었으나 경고가 표시됨
- **exportErrorDetails**: 선택사항으로, 이 값이 true로 설정되면 실패했거나 로그 파일에 오류가 보고된 작업에 대한 세부정보를 익스포트합니다. 이 오류 로그 파일은 출력 ZIP 파일에 포함됩니다. 기본값은 false입니다. 이 값이 true로 설정되면 다음 상태의 작업에 대한 상태 세부정보가 익스포트됩니다.
 - 실패했으며 오류가 있음
 - 완료되었으나 상태를 알 수 없음
 - 완료되었으나 임계값 위반 상태임
 - 완료되었으나 오류가 있음
 - 완료되었으나 경고가 표시됨
- **excludeApplicationId**: 선택사항으로, 애플리케이션 ID를 익스포트 파일에 기록할 것인지 여부를 나타냅니다. 기본값은 false입니다.

① Note

애플리케이션 ID가 포함되지 않은 익스포트된 파일의 데이터는 Oracle Fusion Cloud Enterprise Performance Management 환경으로 임포트할 수 없습니다.

예

- 사용가능한 모든 작업 콘솔 레코드 익스포트:

```
epmautomate exportJobConsole jobs.zip nDays=all jobTypes=all
jobStatusCodes=all
```

- 사용가능한 모든 규칙 작업 콘솔 레코드 익스포트:
epmautomate exportJobConsole jobs.zip nDays=all jobStatusCodes=all
- 애플리케이션 ID 없이 사용가능한 모든 규칙 작업 콘솔 레코드 익스포트:
epmautomate exportJobConsole jobs.zip nDays=all jobStatusCodes=all
excludeApplicationId=true
- 최근 14일 동안 성공적으로 완료된 규칙 작업 레코드만 익스포트:
epmautomate exportJobConsole jobs.zip nDays=14
- 최근 7일 동안 실행된 메타데이터 임포트 및 큐브 지우기 작업 중 실패했으며 오류가 있음 또는 완료되었으나 오류가 있음 상태인 작업의 콘솔 레코드 및 오류 익스포트:
epmautomate exportJobConsole jobs.zip jobtypes=IMPORT_METADATA,CLEAR_CUBE
jobStatusCodes=3,8 exportErrorDetails=true

exportLibraryArtifact

Narrative Reporting 라이브러리 아티팩트를 익스포트합니다. 선택적으로, 보고서 아티팩트의 경우에만 이 명령은 익스포트를 Financial Consolidation and Close, Planning, Planning 모듈 또는 Tax Reporting으로 임포트할 수 있는 LCM 파일로 변환할 수 있습니다.

익스포트가 완료되면 [downloadFile](#) 명령을 사용하여 익스포트 및 오류 파일을 로컬 컴퓨터로 다운로드합니다.

적용 대상

Narrative Reporting

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

고급 사용자, 사용자, 조회자 역할의 사용자에게는 ACL을 통해 추가 보안 권한이 부여되어야 합니다.

사용법

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

- *ARTIFACT_PATH*는 Narrative Reporting 라이브러리에 있는 아티팩트의 위치입니다.
- *EXPORT_FILE*은 아티팩트를 익스포트할 파일의 고유 이름입니다.
- *exportFormat*은 선택사항으로, 다음 중 하나입니다.
 - *Native*는 다른 Narrative Reporting 환경에서 함께 사용할 수 있는 zip 파일로 아티팩트를 익스포트합니다. 기본값입니다.
 - *File*은 파일을 Narrative Reporting 내에서 사용할 수 있는 원래 이진 형식(PDF, DOCX, Zip, JPEG 등)으로 익스포트합니다. 이 매개변수를 사용하여 이진 파일만 익스포트할 수 있습니다. 이 매개변수는 보고서 아티팩트와 함께 사용하지 않아야 합니다.
 - *LCM*은 보고서를 Migration에서 사용되는 형식으로 변환하고 Financial Consolidation and Close, Planning, Planning 모듈 또는 Tax Reporting 환경으로 임포트할 수 있는 ZIP 파일로 익스포트합니다.

- `applicationName`은 선택사항으로, 보고서를 임포트하려는 타겟 애플리케이션의 이름입니다. 이 값은 LCM을 `exportFormat` 매개변수 값으로 사용하는 경우에만 필요합니다.
- `errorFile`은 익스포트 관련 오류를 저장할 텍스트 파일의 고유 이름입니다(선택사항).

예

- 다른 Narrative Reporting 환경으로 임포트할 수 있도록 기본 형식으로 보고서 익스포트:

```
epmautomate exportLibraryArtifact "Library/Samples/Sample Report 1"
exp_SampleReport1.doc errorFile=export_errors.txt
```
- 원래 이진 형식으로 스프레드시트 익스포트:

```
epmautomate exportLibraryArtifact "Library/Spreadsheets/Sheet1.xlsx"
exp_Sheet1.xlsx exportFormat=File errorFile=export_errors.txt
```
- 보고서를 익스포트하고 Financial Consolidation and Close, Planning, Planning 모듈 또는 Tax Reporting으로 임포트할 수 있도록 형식 지정:

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

exportLibraryDocument

보고서 라이브러리에서 사용할 수 있는 문서를 파일로 익스포트합니다.

[downloadFile](#) 명령을 사용하여 익스포트된 파일을 로컬 컴퓨터로 다운로드할 수 있습니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

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

- `ARTIFACT_PATH`는 보고서 라이브러리에 있는 문서의 위치 또는 Universally Unique Identifier (UUID)입니다. 여러 문서를 익스포트하려면 UUID 또는 문서 위치의 심표로 구분된 목록을 지정합니다.
- `jobName`은 선택사항으로, 문서를 익스포트하는 데 사용할 라이브러리 문서 익스포트 작업의 이름입니다. 기본 작업 이름은 Copy Artifact From Library입니다.
- `EXPORT_FILE`은 선택사항으로, 익스포트 파일의 고유한 이름입니다.
`exportFormat=ZIP`을 사용하는 경우, 생성된 ZIP 파일은 이 이름을 사용합니다. 이 값에는 .zip 확장자를 사용해야 합니다.
`exportFormat=file`을 사용하는 경우:
 - 하나의 문서 익스포트: 지정한 파일 확장자가 문서의 파일 확장자와 일치하면 이 이름으로 익스포트 파일이 생성됩니다.
 - 여러 문서 익스포트: 명령에서 이 값을 무시하고 각 문서를 원래 이름으로 익스포트합니다.
- `exportFormat`은 선택사항으로, 다음 중 하나입니다.

- File은 라이브러리 내에서 사용가능한 원래 이진 형식(PDF, DOCX, Zip, JPEG 등)으로 문서를 익스포트합니다. 익스포트한 각 파일은 원래 이름을 사용하여 아웃박스에 생성됩니다. 기본값입니다.
- zip은 원래 이진 형식으로 익스포트한 모든 문서를 포함하는 ZIP 파일을 생성합니다. 선택사항인 exportFile 매개변수에 값을 지정하지 않으면 명령을 실행하는 사용자의 사용자 이름과 타임스탬프를 사용하여 zip 파일 이름이 자동으로 생성됩니다. 파일 이름 형식은 `username_exported_artifacts_yyyyMMddHHmm.zip`입니다(예: `JDoe_exported_artifacts_202410201559.zip`).
- `errorFile`은 선택사항으로, 익스포트 관련 오류를 저장할 파일의 고유 이름입니다. 이 값을 지정하지 않으면 오류 파일이 생성되지 않습니다.
- `overwrite`는 선택사항으로, 현재 기존 다운로드 위치에서 동일한 이름의 파일을 덮어써야 하는지 여부를 제어합니다. 기본값은 `false`로, 동일한 이름의 파일이 아웃박스에 존재하는 경우 명령이 실패함을 의미합니다.

예

- 하나의 문서 익스포트:

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

```
epmautomate exportLibraryDocument Library/folder1/WeeklySales.html,Library/
folder2/WeeklySalesReport.pdf jobName="Copy Weekly Sales"
exportFile=WeeklySales.zip errorFile=WeeklySalesError.log overWrite=true
exportFormat=zip
```

exportLogoImage

Account Reconciliation 비즈니스 프로세스에 사용된 기업 로고를 JPG 파일로 익스포트하여 다른 환경으로 임포트할 수 있도록 합니다.

적용 대상

Account Reconciliation

필수 역할

서비스 관리자

사용법

`epmautomate exportLogoImage IMAGE_NAME.jpg`, 설명: `IMAGE_NAME`은 로고 이미지 파일의 이름입니다.

[downloadFile](#) 명령을 사용하여 익스포트된 로고 파일을 다운로드할 수 있습니다. [uploadFile](#) 명령을 사용하여 타겟 환경에 업로드한 후 [importLogoImage](#) 명령을 실행하십시오.

예

```
epmautomate exportLogoImage corpLogo.jpg
```

exportMapping

특정 차원 또는 위치의 매핑 규칙을 내보내 매핑 규칙 파일을 생성합니다. 매핑을 익스포트하려면 파일 이름과 인박스 내의 위치(예: inbox/exportedAccountMap.txt 또는 inbox/france sales/exportedAccountMap.txt)를 지정해야 합니다.

[downloadFile](#) 명령을 사용하여 익스포트된 매핑 파일을 로컬 컴퓨터로 다운로드할 수 있습니다.

적용 대상

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

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

`epmautomate exportMapping DIMENSION_NAME/ALL FILE_NAME LOCATION` 설명:

- `DIMENSION_NAME/ALL`은 매핑을 익스포트할 소스 차원입니다. 매핑을 내보낼 차원의 이름을 지정하거나, 위치의 모든 차원에서 매핑을 내보내려면 `ALL`을 지정합니다.
- `FILE_NAME`은 매핑 파일의 고유 이름과 아웃박스 내의 위치입니다.
- `LOCATION`은 매핑 규칙을 익스포트해야 하는 Data Management 위치입니다.

예

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

exportMetadata

`export metadata` 유형의 작업에 지정된 설정을 사용하여 파일에 메타데이터를 내보냅니다. 내보낸 데이터 파일을 포함하는 파일은 로컬 컴퓨터에 다운로드할 수 있는 기본 다운로드 위치에 저장됩니다.

선택적으로, 내보낸 데이터의 파일 이름을 지정할 수 있습니다. 이 이름이 기본 파일 이름 (메타데이터를 내보내는 데 사용한 작업 이름)을 재정의합니다. 메타데이터는 ZIP 파일로만 내보냅니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

`epmautomate exportMetadata JOB_NAME [FILE_NAME]` 설명: `JOB_NAME`은 애플리케이션에서 정의된 작업 이름이고 `FILE_NAME`은 메타데이터를 익스포트할 ZIP 파일의 이름입니다. [downloadFile](#) 명령을 사용하여 이 파일을 로컬 서버로 다운로드할 수 있습니다.

예

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

exportOwnershipData

엔티티의 소유권 데이터를 심표로 구분된 CSV 파일로 익스포트합니다.

Financial Consolidation and Close를 통해 채워진 기본 소유권 데이터는 익스포트 파일에 포함되지 않습니다. 사용자가 기본 설정을 대체하도록 입력한 데이터만 익스포트 파일에 포함됩니다.

적용 대상

Financial Consolidation and Close 및 Tax Reporting.

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자

사용법

`epmautomate exportOwnershipData Entity Scenario Year Period FILE_NAME` 설명:

- `Entity`는 데이터를 익스포트할 엔티티의 이름입니다.
- `Scenario`는 데이터를 익스포트할 시나리오입니다.
- `Year`는 데이터를 익스포트할 연도입니다.
- `Period`는 데이터를 익스포트할 연도의 기간입니다.
- `FILE_NAME`은 데이터를 익스포트할 CSV 파일의 이름입니다. [downloadFile](#) 명령을 사용하여 이 파일을 로컬 서버로 다운로드할 수 있습니다.

예

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

exportQueryResults

애플리케이션에서 정의된 쿼리를 실행하고 결과를 텍스트 파일로 내보냅니다.

쿼리 결과 파일은 `profitoutbox`에 저장됩니다. [downloadFile](#) 명령을 사용하거나 Profitability and Cost Management 파일 탐색기를 사용하여 다운로드할 수 있습니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사용법

```
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] 설명:
```

- `APPLICATION_NAME`은 쿼리를 실행할 Profitability and Cost Management 애플리케이션의 이름입니다.
- `fileName`은 쿼리 결과를 저장할 파일의 이름입니다. `queryName` 매개변수 값을 지정하지 않으면 이 매개변수 값이 필요합니다. 선택적으로, `queryName` 매개변수 값을 지정하면 이 쿼리 이름이 쿼리 결과 파일의 이름으로 사용됩니다. 지정한 데이터 형식에 따라 출력 파일 형식이 결정됩니다. `dataFormat=NATIVE`(기본값)를 사용하는 경우 익스포트 프로세스에서 텍스트 파일이 생성됩니다. `dataFormat=COLUMNAR`를 사용하는 경우 익스포트 프로세스에서 순차적으로 번호가 지정된 텍스트 파일을 여러 개 생성하여 하나의 Zip 파일로 압축합니다.
- `fileOutputOptions`는 선택사항으로, 쿼리 결과 파일의 출력 형식을 나타냅니다. 기본값은 `ZIP_ONLY`이며, 이 값을 사용하면 `fileName` 매개변수 값이 지정되었는지 여부에 따라 `fileName.ZIP` 또는 `queryName.ZIP`이 생성됩니다. 다른 옵션으로는, 출력 파일을 텍스트 파일로 생성하는 `TEXT_ONLY` 및 텍스트 파일과 zip 파일을 둘 다 생성하는 `ZIP_AND_TEXT`가 있습니다.
- `queryName`은 애플리케이션에서 정의된 쿼리를 식별하는 선택적 매개변수입니다. 공백 문자가 포함된 쿼리 이름은 큰따옴표로 묶어야 합니다. 애플리케이션에 속하는 Oracle Essbase 데이터를 모두 익스포트하려는 경우 쿼리 이름을 지정하지 마십시오.

다음 조건에서 이 명령은 빈 데이터 파일을 생성할 수 있습니다.

- 데이터를 검색하지 않는 잘못된 형식의 쿼리
- 데이터를 너무 많이 생성하는 쿼리. 이 시나리오에서는 더 적은 데이터를 검색하거나 쿼리를 더 작은 쿼리로 나누도록 쿼리 범위를 좁히는 것이 좋습니다.
*Profitability and Cost Management 관리*에서 Oracle Profitability and Cost Management Cloud 쿼리 관리를 참조하십시오.
- `exportOnlyLevel0Flg`는 선택사항으로, 쿼리에서 level0 데이터만 검색할지 여부를 지정합니다. 이 매개변수 값을 모두 소문자로 지정합니다. 쿼리 이름을 생략하여 모든 애플리케이션 데이터를 내보내는 경우 이 매개변수는 무시됩니다.
- `roundingPrecision`은 선택사항으로, 쿼리 결과를 익스포트할 때 사용할 소수 자릿수(반올림 정밀도)를 지정합니다. `queryName`을 지정한 경우에만 적용할 수 있습니다. 기본값은 2입니다.
- `dataFormat`은 선택사항으로, 출력 형식을 나타냅니다. 적합한 값은 다음과 같습니다.
 - `NATIVE`는 쿼리 결과를 Essbase 기본 형식 데이터로 유지관리합니다. 기본값입니다.
 - `COLUMNAR`는 Essbase 기본 형식 데이터를 변환하며 열에 순서를 지정하여 쉽게 해석하고 다른 애플리케이션으로 임포트할 수 있도록 합니다.

이 옵션은 모든 Essbase 데이터를 익스포트하고 queryName 매개변수 값을 무시합니다. memberFilters 매개변수 값을 설정하여 데이터를 필터링할 수 있습니다.

① 주

이 명령에서는 dataFormat이 COLUMNAR로 지정된 경우에만 다음 선택적 매개변수를 고려합니다.

- memberFilters는 선택사항으로, JSON으로 형식 지정된 문자열에서 차원 및 level0 멤버별로 필터링하도록 허용합니다. 예를 들어 "{ \"Dim1\": [\"Mem1\"], \"Dim2\": [\"Mem21\", \"Mem22\"] }"가 있습니다.
- includeHeader는 선택사항으로, 차원 이름을 열 머리글로 추가합니다. 열 머리글을 제외하려면 이 값을 false로 설정하십시오. 기본값은 true입니다.
- delimiter는 선택사항으로, 쿼리 결과 파일에서 차원 멤버를 구분하는 데 사용하는 구분자를 나타냅니다. 구분자는 큰따옴표로 묶어야 합니다. 기본값은 공백(" ")입니다.
- keepDuplicateMemberFormat은 선택사항으로, 멤버 형식을 Essbase 중복 멤버 형식(예: [Account]@[Account])으로 인쇄할 것인지 지정합니다. 멤버 이름만 인쇄하려면 이 값을 false로 설정하십시오. 기본값은 true입니다.

예

- 모든 애플리케이션 데이터 익스포트:
epmautomate exportQueryResults BksML12 fileName="BksML12_MyQuery1.txt"
fileOutputOptions=TEXT_ONLY
- 특정 쿼리 결과 익스포트:
epmautomate exportQueryResults BksML12 queryName="My Product Query"
roundingPrecision=3
- NATIVE 데이터 형식으로 Level0 데이터 익스포트:
epmautomate exportQueryResults BksML30 fileName="BksML30_ExportLevel0-Data"
fileOutputOptions=ZIP_AND_TEXT exportOnlyLevel0Flg=true
- 단일 차원 및 단일 멤버 필터가 사용된 COLUMNAR 데이터 형식으로 Level0 데이터 익스포트:
epmautomate exportQueryResults BksML30 fileName="BksML30_Level0-Data"
dataFormat="COLUMNAR" memberFilters="{ \"Period\": [\"December\"] }"
includeHeader="true" delimiter="," roundingPrecision="3"
- 단일 차원 및 여러 멤버 필터가 사용된 COLUMNAR 데이터 형식으로 Level0 데이터 익스포트:
epmautomate exportQueryResults BksML30 fileName="BksML30_Level0-Data"
dataFormat="COLUMNAR" memberFilters="{ \"Period\": [\"November\", \"December\"] }"
includeHeader="true" delimiter="," roundingPrecision="3"
- 여러 차원 및 여러 멤버 필터가 사용된 COLUMNAR 데이터 형식으로 Level0 데이터 익스포트:
epmautomate exportQueryResults BksML30 fileName="BksML30_Level0-Data"
dataFormat="COLUMNAR" memberFilters="{ \"Year\": [\"2016\"], \"Period\": [\"November\", \"December\"] }" includeHeader="true" delimiter=","
roundingPrecision="3"

exportSnapshot

이전에 수행한 익스포트 작업을 반복하여 마이그레이션 콘텐츠의 스냅샷을 생성합니다.

마이그레이션을 사용하여 원하는 아티팩트를 선택하고 스냅샷(예: January16FullApp)으로 익스포트합니다. 이 명령에 스냅샷 이름을 사용하여 익스포트 작업을 순차적으로 반복합니다. 그러면

원래 익스포트 작업 중에 선택한 아티팩트만 익스포트됩니다. *마이그레이션 관리*의 아티팩트 및 애플리케이션 익스포트를 참조하십시오.

- 다음은 Planning, Planning 모듈, 무형식 애플리케이션 스냅샷에 포함되지 않습니다.
 - 감사 데이터
 - 작업 콘솔 데이터
 감사 및 작업 콘솔 데이터를 타겟 환경에 복사하려는 경우 [cloneEnvironment](#) 명령이나 환경 복제 기능을 사용합니다.
- 스냅샷에는 Data Management 스테이지 테이블 데이터가 포함되지 않습니다. 이 데이터를 임포트하려면 [exportDataManagement](#) 및 [importDataManagement](#) 명령이나 Data Management 시스템 유지관리 스크립트 인터페이스를 사용합니다. [cloneEnvironment](#) 명령이나 환경 복제 기능을 사용하여 Data Management 스테이지 테이블 데이터를 비롯한 환경의 동일한 복사본을 생성할 수 있습니다.

[downloadFile](#) 명령을 사용하여 익스포트된 스냅샷을 기본 위치에서 다운로드할 수 있습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

`epmautomate exportSnapshot SNAPSHOT_NAME` 설명: `SNAPSHOT_NAME`은 마이그레이션의 기존 스냅샷 이름입니다. 이 스냅샷은 새로운 스냅샷으로 대체됩니다.

예

```
epmautomate exportSnapshot January16FullApp
```

exportTemplate

애플리케이션을 .ZIP 파일에 템플릿으로 내보냅니다. 내보낸 파일은 profitoutbox에 저장됩니다.

[downloadFile](#) 명령을 사용하여 익스포트된 파일을 로컬 컴퓨터로 다운로드할 수 있습니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

`epmautomate exportTemplate APPLICATION_NAME File_Name` 설명:

- `APPLICATION_NAME`은 템플릿으로 익스포트할 Profitability and Cost Management 애플리케이션의 이름입니다.
- `File_Name`은 템플릿 파일의 이름입니다.

예

```
epmAutomate exportTemplate BksML12 template1
```

exportTaskManagerAccessControl

태스크 관리자, 보충 데이터 및 기업 분개 사용자 지정에 대한 사용자 세부정보 보고서를 익스포트합니다. Excel 또는 CSV 파일로 된 이 보고서에는 환경에 사전 정의된 역할이 있는 사용자에 대한 정보가 있으며 각 사용자 속성(예: 이름 및 전자메일) 및 해당 상태, 팀, 사전 정의된 역할, 워크플로우 역할, 조직, 그룹, 마지막 로그인 타임스탬프가 나열되어 있습니다.

다음은 샘플 태스크 관리자 액세스 제어 보고서입니다.

	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	Organization	Last Login	Form Preparer	Form Approver	Form Integrator	Groups	
1															
2	John Doe	john.doe@example.com	Available	Admins	john.doe@example.com	Administrator								Service Administrator	
3	Jane Doe	jane.doe@example.com	Available		jane.doe@example.com	Power User								Power User	
4	user Ats	ats_user	Available		example1@example.com	User								User	
5	user2 Ats	ats_user2	Available		example2@example.com	user								User	
6	View User	viewUser	Available		view.user@exaple.com	Viewer								Viewer	

적용 대상

Enterprise Profitability and Cost Management, Financial Consolidation and Close, Tax Reporting

필수 역할

서비스 관리자

사용법

`epmAutomate exportTaskManagerAccessControl REPORT_NAME`. 여기서 `REPORT_NAME`은 보고서가 포함될 익스포트 파일의 이름입니다(적합한 (CSV 또는 XLS) 확장자 포함).

이 보고서는 CSV 또는 XLS 형식으로 생성할 수 있습니다. [downloadFile](#) 명령을 사용하여 다운로드할 수 있습니다.

예

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

exportValidIntersections

비즈니스 프로세스의 적합한 교차점 그룹을 ZIP 파일로 익스포트하면 [downloadFile](#) 명령을 사용하여 로컬 컴퓨터로 다운로드할 수 있습니다. 적합한 교차점은 사용자가 정의하는 적합한 교차점 규칙이라는 규칙을 기반으로 필터링되는 셀 상호 작용입니다. 이러한 규칙에 따라 데이터를 입력하거나 런타임 프롬프트를 선택할 때 사용자에게 대한 특정 셀 상호 작용이 필터링됩니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

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

설명:

- *FILE_NAME*은 익스포트 ZIP 파일의 이름입니다. 명령에 표시된 모든 적합한 교차점은 Microsoft Excel 파일로 익스포트된 후 압축되어 이 파일이 생성됩니다.
- *names*는 선택사항으로, 익스포트할 적합한 교차점의 심표로 구분된 목록을 나타냅니다. 이 매개변수 값이 지정되지 않으면 명령이 애플리케이션의 모든 적합한 교차점을 익스포트합니다.

예

- **적합한 특정 교차점을 익스포트합니다.**

```
epmautomate exportValidIntersections VI_export_File.zip
names=VIAccountPeriod,VIEntityPeriod,VIProductPeriod
```
- **모든 적합한 교차점을 익스포트합니다.**

```
epmautomate exportValidIntersections VI_export_File.zip
```

extractDimension

Oracle Fusion Cloud Enterprise Data Management 차원을 파일 또는 글로벌 연결로 추출합니다.

적용 대상

Cloud EDM

필수 역할

- 서비스 관리자
- 데이터 관리자 권한이 있는 사용자

사용법

```
epmautomate extractDimension APPLICATION DIMENSION EXTRACT_NAME FILE_NAME
[connection=NAME]. 설명:
```

- *APPLICATION*은 Cloud EDM 애플리케이션의 이름입니다.
- *DIMENSION*은 추출할 차원의 이름입니다.
- *EXTRACT_NAME*은 애플리케이션에 정의된 추출 프로파일의 이름입니다. 이 프로파일은 차원을 추출하는 데 사용됩니다.
- *FILE_NAME*은 추출한 데이터를 저장할 파일(파일로 익스포트하는 경우 CSV, Oracle Financials Cloud로 익스포트하는 경우 ZIP)의 이름입니다. 연결 매개변수 값을 설정하지 않으면 이 파일은 준비 영역에 생성됩니다. [downloadFile](#) 명령을 사용하여 로컬 컴퓨터로 다운로드하거나 [copyFileFromInstance](#) 명령을 사용하여 다른 Cloud EDM 환경으로 복사할 수 있습니다.

- `connection=NAME`은 선택사항으로, Cloud EDM에 파일 위치로 정의된 글로벌 연결 이름 (인스턴스 위치)을 나타냅니다. 지정되는 경우, 추출 파일이 타겟 환경(Oracle Fusion Cloud Enterprise Performance Management의 경우 인박스, Oracle ERP의 경우 지정된 문서 계정)으로 업로드됩니다.

① Note

글로벌 연결에 지정된 인증서에는 타겟 환경에 쓸 수 있는 액세스 권한이 있어야 합니다.

예

- Cloud EDM 준비 영역으로 익스포트: `epmautomate extractDimension USOperations Entity EntityExtProfile EntityData.CSV`
- Oracle ERP로 추출 및 업로드: `epmautomate extractDimension USOperations Entity EntityExtProfile EntityData.zip Connection=ora_fusion_gl`
- 타겟 Cloud EPM 인박스로 추출 및 업로드: `epmautomate extractDimension USOperations Entity EntityExtProfile EntityData.CSV Connection=EPM_cloud_pln`

extractPackage

단일 작업을 사용하여 애플리케이션에 대한 여러 추출로 구성된 추출 패키지를 실행합니다. 패키지에 포함된 각 추출의 결과는 하나의 ZIP 파일에 추가되며, 이 파일은 준비 영역 또는 글로벌 연결에 기록될 수 있습니다.

적용 대상

Oracle Fusion Cloud Enterprise Data Management

필수 역할

- 서비스 관리자
- 데이터 관리자 권한이 있는 사용자

사용법

```
epmautomate extractPackage APPLICATION_NAME EXTRACT_PACKAGE_NAME FILE_NAME.zip
[connection=CONNECTION_NAME]:
```

- `APPLICATION_NAME`은 Cloud EDM 애플리케이션의 이름입니다.
- `EXTRACT_PACKAGE_NAME`은 Cloud EDM 애플리케이션에 정의된 패키지의 이름입니다.
- `FILE_NAME`은 추출 결과를 포함할 ZIP 파일의 이름입니다. 이 파일은 준비 영역에서 생성되거나 연결 이름이 지정된 경우 글로벌 연결에 기록됩니다. 패키지에는 각각 이름이 고유한 하나 이상의 추출 프로파일이 포함될 수 있습니다. 추출 패키지 ZIP 파일에는 패키지에 있는 추출의 출력 파일이 포함됩니다. ZIP 파일의 파일 수는 패키지의 설정에 따라 다릅니다.
- `CONNECTION_NAME`은 선택사항으로, Cloud EDM에 정의된 글로벌 연결 이름(인스턴스 위치)입니다. 이는 추출 결과가 포함된 ZIP 파일을 업로드할 위치를 확인합니다. 연결 이름을 지정하면 추출된 데이터가 포함된 ZIP 파일이 타겟 환경(Oracle Fusion Cloud Enterprise Performance Management 환경의 경우 인박스, Oracle ERP의 경우 지정된 문서 계정)에 업로드됩니다. 값을 지정하지 않으면 익스포트 파일이 Cloud EDM 환경의 기본 아웃박스(준비 영역)에 생성됩니다.

예

- COARedesignMaps 패키지를 Cloud EDM 준비 영역의 파일로 추출합니다.

```
epmautomate extractPackage FinancialsCloud COARedesignMaps COARedesign.zip
```

- COARedesignMaps 패키지를 파일로 추출하고 연결을 사용하여 Oracle Fusion Cloud EPM 인박스에 해당 파일을 업로드합니다.

```
epmautomate extractPackage CorporateAccount COARedesignMaps  
COARedesign.zip Connection=EPM_cloud_pln
```

feedback

환경의 서비스 관리자와 Oracle에 피드백을 전송하고 현재 디렉토리에서 최근 24시간 동안 생성된 모든 EPM Automate 로그 파일을 자동으로 업로드합니다.

오라클 고객센터에서 현재 이슈의 발생 이유를 진단하는 데 사용할 수 있도록 추가 파일(예: Fiddler 추적 파일)을 선택적으로 업로드할 수 있습니다.

서비스의 피드백 제공 기능을 모방하는 이 명령은 사용자 인터페이스가 응답하지 않거나 EPM Automate 실행 중 이슈가 발생하여 Oracle에 피드백을 제공하는 경우 특히 유용합니다.

피드백 제공 기능에 대한 자세한 내용은 *관리자용 시작 가이드*에서 피드백 제공 유틸리티를 사용하여 Oracle이 진단 정보를 수집할 수 있도록 지원을 참조하십시오.

이 명령은 피드백이 서비스 요청을 생성하지 않음을 알리는 다음과 유사한 메시지를 반환합니다. 문제를 해결하기 위해 Oracle의 도움이 필요한 경우 서비스 요청을 제출해야 합니다. 이 명령은 제출하는 서비스 요청에 포함해야 하는 UDR 참조 번호를 표시합니다.

```
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>
```

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사용법

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

- Comment는 이 피드백이 제출되는 이슈를 설명하는 텍스트입니다. 설명은 따옴표로 묶어야 합니다.
- Screenshot은 선택사항으로, 이 피드백이 제출되는 이슈를 보여주는 그래픽 파일의 이름을 확인합니다. 이 매개변수와 값을 필요에 따라 반복하여 여러 스크린샷을 제출할 수 있습니다.
- File은 선택사항으로, 오라클 고객지원센터에서 현재 이슈를 해결하는 데 사용할 수 있도록 제공할 파일의 이름을 확인합니다. 이 매개변수를 사용하여 Fiddler 추적 파일 또는 다른 파일을 Oracle에 제출합니다. 이 매개변수와 값을 필요에 따라 반복하여 여러 파일을 제출할 수 있습니다.

예

- **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

애플리케이션이 서비스 관리자만 액세스하여 관리 모드에 있는지 여부를 확인합니다.

애플리케이션이 관리 모드인 경우 true 관리 모드가 아닌 경우 false를 반환하는 이 명령은 자동화 스크립트를 실행하기 전에 애플리케이션의 상태를 확인하는 데 유용합니다. 예를 들어, [refreshCube](#) 명령은 애플리케이션이 관리 모드여야 합니다. 다음과 같이 자동화 스크립트에 이 명령을 사용하여 애플리케이션이 관리 모드에 있는지 확인할 수 있습니다.

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

적용 대상

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

필수 역할

서비스 관리자

사용법

```
epmautomate getApplicationAdminMode
```

예

```
epmautomate getApplicationAdminMode
```

getDailyMaintenanceStartTime

환경의 일별 유지관리가 시작되도록 스케줄링된 UTC(협정 세계시) 또는 시간대(선택사항)를 콘솔에 표시합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

`epmautomate getDailyMaintenanceStartTime [timezone=true|false]` 설명:
`timezone=true`는 선택사항으로, `America/Los_Angeles` 같이 시간대를 설정할 때 지정한 시간대로 일별 유지관리 시작 시간을 표시할지 여부를 나타냅니다. 기본값은 `false`입니다.

예

- 시간대를 설정할 때 지정한 시간대로 유지관리 시간을 표시합니다.
`epmautomate getDailyMaintenanceStartTime timezone=true`
- UTC로 유지관리 시간을 표시합니다.
`epmautomate getDailyMaintenanceStartTime`

getEssbaseQryGovExecTime

쿼리가 종료되기 전에 Oracle Essbase 쿼리가 정보를 검색 및 전달하기 위해 사용할 수 있는 현재 최대 시간(초)을 표시합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

`epmautomate getEssbaseQryGovExecTime`

샘플 명령 출력:

```
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
```

예

```
epmautomate getEssbaseQryGovExecTime
```

getIdleSessionTimeout

Oracle Fusion Cloud Enterprise Performance Management 환경의 세션 시간 초과(분)를 표시합니다. 이 기간 동안 세션이 유휴 상태인 경우 사용자를 로그인 페이지로 리디렉션합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자

사용법

```
epmautomate getIdleSessionTimeout
```

샘플 명령 출력:

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

getIPAllowlist(사용되지 않음)

현재 허용 목록에 포함된 IP 주소와 CIDR(Classless Inter-Domain Routings)을 표시합니다.

Note

이 명령은 더 이상 사용되지 않으며 이후 업데이트에서 제거될 예정입니다. 환경별 IP 허용 목록을 사용하는 대신 Oracle Identity Cloud Service에서 네트워크 경계를 설정하여 전체 도메인에 대한 특정 IP 주소 또는 범위로 로그인을 제한합니다. 다음 정보 소스를 참조하십시오.

- *관리자용 시작 가이드*의 네트워크 경계 설정
- Oracle Identity Cloud Service 관리의 [Oracle Identity Cloud Service 네트워크 경계 관리](#)

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자

사용법

```
epmAutomate getIPAllowlist
```

예

현재 허용 목록에 포함된 IP 주소 및 CIDR 표시:

```
epmAutomate getIPAllowlist
```

getRestrictedDataAccess

서비스 관리자가 피드백 제공 유틸리티를 사용하는 동안 Oracle에 애플리케이션 스냅샷을 제출하는 데 동의하지 못하도록 환경이 구성되어 있는지 여부를 표시합니다.

이 명령은 서비스 관리자가 애플리케이션 스냅샷을 제출하는 데 동의하지 못하도록 환경이 구성된 경우 true를 표시하고 그렇지 않은 경우 false를 표시합니다.

적용 대상

Account Reconciliation, Oracle Fusion Cloud Enterprise Data Management, Enterprise Profitability and Cost Management, Financial Consolidation and Close, FreeForm, Planning, Planning Modules, Profitability and Cost Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning, Tax Reporting

필수 역할

서비스 관리자

사용법

현재 데이터 액세스 제한 상태를 변경하려면 [setRestrictedDataAccess](#)를 사용합니다.

예

```
epmautomate getRestrictedDataAccess
```

getSubstVar

대체 변수 값을 검색하여 화면에 표시합니다.

표시 형식은 `CUBE_NAME.SUBSTVAR=value`입니다(예: `Plan2.CurYear=2016`). 애플리케이션 레벨 대체 변수 값은 `ALL.SUBSTVAR=value` 형식(예: `ALL.CurYear=2016`)으로 표시됩니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

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

- `CUBE_NAME`은 대체 변수를 검색할 소스 큐브(예: `Plan1`, `Plan2`)입니다. 애플리케이션 레벨에서 대체 변수를 검색하려면 `ALL`을 사용합니다.
- `name=VARIABLE_NAME`은 선택적으로 값을 검색할 대체 변수를 확인합니다. 변수 이름을 지정하지 않으면 명령은 모든 대체 변수 값을 검색합니다.

예

- 애플리케이션 및 큐브 레벨에서 모든 대체 변수의 값 가져오기: `epmautomate getSubstVar ALL`
- 애플리케이션 레벨에서 특정 대체 변수 하나의 값 가져오기: `epmautomate getSubstVar ALL name=CurYear`
- 큐브 레벨에서 모든 대체 변수의 값 가져오기: `epmautomate getSubstVar Plan2`
- 큐브 레벨에서 특정 대체 변수 하나의 가져오기: `epmautomate getSubstVar Plan2 name=CurYear`

getVirusScanOnFileUploads

업로드되는 모든 파일을 검사하여 바이러스가 없는지 확인하기 위해 환경이 사용으로 설정되었는지 확인합니다.

이 명령은 파일을 환경에 업로드하기 전에 바이러스 검사를 실행할지 여부를 확인합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자

사용법

```
epmautomate getVirusScanOnFileUploads
```

이 명령은 업로드된 파일에 대해 바이러스를 검사하도록 환경이 설정되어 있으면 true를 출력하고, 그렇지 않으면 false를 출력합니다.

groupAssignmentAuditReport

지정된 날짜 범위 동안 액세스 제어 그룹에 지정되었거나 지정 취소(제거)된 사용자와 그룹을 나열하는 보고서를 생성합니다. 이 보고서는 최대 120일에 대해 생성될 수 있습니다.

CSV 파일로 생성되는 이 보고서를 사용하여 보안 감사 작업을 지원할 수 있습니다. 생성된 CSV 파일의 각 행은 추가되었거나 제거된 사용자 또는 그룹, 사용자 또는 그룹이 추가되었거나 제거된 그룹, 그룹 유형, 작업을 수행한 서비스 관리자, 작업이 완료된 날짜 및 시간을 제공합니다.

	A	B	C	D	E	F	G
1	User/Group Name	User/Group Type	Action	Group	Group Type	Performed By	Date and Time
2	Reviewer	EPM Group	Assigned	Strategic Planner	EPM	john.smith@example.com	April 07, 2025 09:40:03 UTC
3	Super User	EPM Group	Assigned	exampleGroup	EPM	john.smith@example.com	April 07, 2025 09:41:03 UTC
4	IDCS_Group1	IDCS Group	Assigned	Group1	EPM	john.smith@example.com	April 07, 2025 09:42:02 UTC
5	Power User	Predefined Group	Unassigned	Group1	EPM	john.smith@example.com	April 07, 2025 09:55:05 UTC
6	jDoe	User	Unassigned	Group1	EPM	john.smith@example.com	April 07, 2025 09:50:05 UTC

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할
- 사전 정의된 역할 및 액세스 제어 - 애플리케이션 역할 보기

사용법

```
epmAutomate groupAssignmentAuditReport FROM_DATE TO_DATE REPORT_NAME 설명
```

- *FROM_DATE*는 보고서를 생성할 기간의 시작 날짜(YYYY-MM-DD 형식)입니다.
- *TO_DATE*는 보고서를 생성할 기간의 종료 날짜(YYYY-MM-DD 형식)입니다.
- *REPORT_NAME*은 보고서 CSV 파일의 이름입니다. [downloadFile](#) 명령을 사용하여 생성된 보고서를 다운로드할 수 있습니다.

Note

Oracle Fusion Cloud Enterprise Performance Management에서는 보고서 생성 중 적합한 날짜 범위만 사용되도록 합니다. 이러한 검증은 *시작* 및 *종료* 날짜에 대해 수행됩니다.

- *FROM_DATE*는 현재 날짜부터 허용된 최대 보존 기간(120일)보다 이전일 수 없습니다.
- *TO_DATE*는 *FROM_DATE*부터 최대 보존 기간보다 이후일 수 없습니다.
- *TO_DATE*는 *FROM_DATE*보다 이전일 수 없습니다.

예

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

help

모든 EPM Automate 명령 또는 특정 명령에 대한 도움말을 표시합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사용법

epmAutomate [*COMMAND_NAME*] help 설명: *COMMAND_NAME*은 선택사항으로, 콘솔에 도움말을 표시하려는 명령의 이름입니다.

예

- 모든 명령에 대한 도움말에 액세스할 수 있는 웹 페이지 표시:
epmAutomate help
- 콘솔에 encrypt 명령에 대한 명령 도움말 표시:
epmAutomate encrypt help

importAppAudit

환경에서 감사 데이터를 익스포트하여 생성한 ZIP 파일에서 데이터 감사 레코드를 임포트합니다.

임포트 파일은 [exportAppAudit](#) 명령(epmAutomate exportAppAudit auditData ndays=All)을 사용하여 생성합니다. 이 명령을 사용하여 마이그레이션 또는 재해 복구를 위한 복제 중 한 환경에서 다른 환경으로 감사 레코드를 복제합니다.

적용 대상

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

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 감사 - 관리 애플리케이션 역할

사용법

`epmautomate importAppAudit FILE_NAME [logFilename=LOG_FILE_NAME]` 설명:

- `FILE_NAME`은 애플리케이션으로 임포트할 데이터 감사 레코드가 포함된 ZIP 파일의 이름입니다. 이 명령을 실행하기 전에 [uploadFile](#) 명령을 사용하여 이 파일을 환경에 업로드합니다.
- `logFileName`은, 선택사항으로, 임포트 중 발생한 오류가 기록될 오류 로그 파일을 나타냅니다. 이 값을 지정하지 않으면 명령은 `username_date_timestamp` 규칙을 사용하여 이름이 지정된 오류 파일을 생성합니다. [downloadFile](#) 명령을 사용하여 이 파일을 다운로드할 수 있습니다.

예

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

importAppSecurity

인박스에 제공된 CSV 파일에서 애플리케이션 사용자 또는 그룹에 대한 액세스 권한을 로드합니다.

액세스 권한을 임포트하면 임포트한 멤버, 데이터 양식, 데이터 양식 폴더, 태스크 목록, Calculation Manager 비즈니스 규칙, 비즈니스 규칙 폴더에 대한 기존 지정만 덮어씁니다. 다른 기존 액세스 권한은 모두 그대로 유지됩니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

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

- `ACL_FILE_NAME`은 애플리케이션으로 임포트할 액세스 권한이 포함된 CSV 파일의 이름입니다. 이 명령을 실행하기 전에 [uploadFile](#) 명령을 사용하여 이 파일을 인박스로 업로드합니다. 샘플 입력 파일의 콘텐츠는 다음 이미지처럼 표시될 수 있습니다.

	A	B	C	D	E	F	G	H
1	Object Name	Name	Parent	Is User	Object Type	Access Type	Access Mode	Remove
2	AllAB	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

열 머리글 및 가능한 값에 대한 설명은 *Oracle Fusion Cloud EPM용 REST API 가이드*의 보안 임포트를 참조하십시오.

- `ERROR_FILE`은 이 작업 중 감지되는 오류를 기록하도록 EPM Automate가 아웃박스에 생성하는 CSV 파일의 이름입니다. 이 파일을 로컬 컴퓨터로 다운로드하여 보고된 오류를 분석하고 정정할 수 있습니다. 샘플 오류 파일의 콘텐츠는 다음 이미지처럼 표시될 수 있습니다. 이 파일의 열은 입력 파일의 머리글 열에 해당합니다.

	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`(선택사항)은 파일에서 새 권한을 로드하기 전에 기존 액세스 권한을 삭제할 것인지 여부를 지정합니다. 기본값은 `false`입니다.

예

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

importARApplicationProperties

익스포트 JSON 파일에서 사용할 수 있는 애플리케이션 설정(Redwood 환경, 테마, 전자메일 통지, 비즈니스 프로세스 이름), 로고, 백그라운드 이미지를 Account Reconciliation 환경으로 임포트합니다.

적용 대상

Account Reconciliation

필수 역할

서비스 관리자

사용법

```
epmautomate importARApplicationProperties FILE_NAME 설명: FILE_NAME은 환경에서 익스포트된 JSON 파일의 이름입니다.
```

[exportARApplicationProperties](#) 명령을 사용하여 다른 환경에서 익스포트된 이 파일은 애플리케이션 설정을 복원 중인 환경에서 사용할 수 있어야 합니다.

예

```
epmautomate importARApplicationProperties myProp.JSON
```

importBackgroundImage

익스포트 파일에서 Account Reconciliation 환경으로 배경 이미지를 임포트합니다.

적용 대상

Account Reconciliation

필수 역할

서비스 관리자

사용법

`epmautomate importBackgroundImage FILE_NAME.jpg`, 설명: `FILE_NAME`은 다른 환경에서 익스포트된 배경 이미지 파일의 이름입니다.

예

```
epmautomate importBackgroundImage image_file.jpg
```

importBalances

Data Management를 사용하여 데이터 로드 정의에서 잔액 데이터를 임포트합니다.

적용 대상

Account Reconciliation.

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

고급 사용자, 사용자, 조회자 역할의 사용자에게는 ACL을 통해 추가 보안 권한이 부여되어야 합니다.

사용법

`epmautomate importBalances DL_DEFINITION PERIOD`, 설명:

- `DL_DEFINITION`은 Account Reconciliation의 기존 데이터 로드 정의입니다.
- `PERIOD`는 기간 이름입니다.

예

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

importCellLevelSecurity

셀 레벨 보안 레코드가 포함된 하나의 Excel 파일이 있는 ZIP 파일에서 비즈니스 프로세스로 셀 레벨 보안 설정을 임포트합니다. 이 명령을 실행하기 전에 [uploadFile](#) 명령을 사용하여 임포트 파일을 환경에 업로드합니다.

임포트 ZIP 파일에는 셀 레벨 보안을 성공적으로 임포트하는 데 필요한 두 개의 워크시트(규칙 및 하위 규칙)가 포함된 하나의 Excel 파일이 있어야 합니다. 규칙 시트에는 셀 레벨 보안 정의, 포함된 차원 및 등록정보(예: Unspecified Valid, Additional Dims Required)가 있어야 합니다. 하위 규칙 시트에는 멤버 선택항목 및 제외항목이 있어야 합니다. 임포트 파일 형식 템플리트를 가져오는 가장 좋은 방법은 애플리케이션에서 셀 레벨 보안을 익스포트하는 것입니다. 다음 그림에 샘플 형식이 있습니다.

	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	Dim1 Required	Dim2	Dim2 Required
1											
2	Sample-CLS	1		true	All	Product	true	Account	false		
3	Sample-CLS-Dup	2		true	All	Product	true	Account	false		
4											
5											

	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												

적용 대상

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

필수 역할

서비스 관리자

사용법

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

- FILE_NAME은 셀 레벨 보안 정보가 포함된 Excel 파일이 있는 ZIP 파일의 이름입니다.
- ErrorFile은 선택사항으로, 오류 레코드가 기록될 텍스트 파일의 이름을 나타냅니다. 이 매개변수 값이 지정되지 않으면 명령이 자동으로 오류 파일을 생성합니다. 이 파일의 이름은 작업 콘솔에서 볼 수 있습니다. [downloadFile](#) 명령을 사용하여 오류 파일을 로컬 컴퓨터로 다운로드합니다.

예

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

importConsolidationJournals

.JLF 파일에서 Financial Consolidation and Close로 연결 분개를 임포트합니다.

- [exportConsolidationJournals](#) 명령을 사용하여 이 명령의 입력으로 사용되는 .JLF 파일을 생성합니다.
- 이 명령을 실행하기 전에 [uploadFile](#) 명령을 사용하여 입력 파일을 환경으로 로드합니다.

적용 대상

재무 통합 및 마감

필수 역할

서비스 관리자

사용법

```
epmautomate importConsolidationJournals jobName [fileName=FILE_NAME]
[errorFileName=ERROR_FILE_NAME] 설명
```

- `jobName`은 Financial Consolidation and Close에서 생성된 분개 импорт 작업의 이름입니다.
- `fileName`은 선택사항으로, 분개 항목을 импорт할 .JLF 파일의 이름입니다.
- `errorFileName`은 선택사항으로, импорт 프로세스 중 생성된 메시지를 기록할 로그 파일의 이름입니다.

예

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

importData

`import data` 유형의 작업에 지정된 데이터 импорт 설정을 사용하여 파일의 데이터를 애플리케이션으로 가져옵니다.

[uploadFile](#) 명령을 사용하여 애플리케이션 데이터가 포함된 파일을 기본 업로드 위치로 업로드할 수 있습니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

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

- `JOB_NAME`은 애플리케이션에서 정의된 작업 이름입니다.
- `FILE_NAME`은 импорт할 데이터가 있는 ZIP, CSV 또는 TXT(Essbase 형식 데이터 파일) 파일의 이름을 확인합니다(선택사항). 파일 이름을 지정한 경우 작업의 импорт 파일 이름은 무시됩니다. Essbase 형식으로 데이터를 가져오도록 작업이 정의된 경우 ZIP 파일에 Essbase 형식 TXT 파일이 있어야 합니다. 기타 импорт 작업의 경우 ZIP 파일에는 파일 이름(예: data1-3.csv, data2-3.csv, and data3-3.csv)으로 импорт 순서를 식별하는 하나 이상의 CSV 파일이 포함될 수 있습니다.
- 선택사항인 `errorFile`은 импорт 작업 중에 거부된 레코드(있는 경우)가 기록되는 ZIP 파일의 이름을 확인합니다. 아웃박스에서 동일한 이름의 ZIP 파일(있는 경우)을 덮어씁니다. [downloadFile](#) 명령을 사용하여 이 파일을 다운로드할 수 있습니다.

예

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

importDataManagement

Data Management 레코드를 ZIP 파일에서 환경으로 임포트합니다.

이 명령은 [exportDataManagement](#) 명령을 사용하여 생성된 ZIP 파일에서 설정 및 스테이지 테이블로 데이터를 임포트합니다. 서비스 관리자는 [uploadFile](#) 명령(예: `epmAutomate uploadFile "C:/datafile/datafile.zip" inbox`)을 사용하여 임포트 ZIP 파일을 Data Management 인박스 또는 인박스 내 폴더로 업로드합니다.

Note

이 명령은 동일한 월별 업데이트로 실행되고 있는 다른 환경에서 익스포트된 Data Management 레코드만 임포트할 수 있습니다. 예를 들어 21.11 Oracle Fusion Cloud Enterprise Performance Management 환경에서 익스포트된 레코드만 다른 21.11 환경으로 임포트할 수 있습니다.

적용 대상

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

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

`epmautomate importDataManagement FILE_NAME.zip` 설명: `FILE_NAME`은 임포트할 Data Management 데이터가 포함되어 있는 ZIP 파일의 이름입니다.

예

- Data Management 인박스에서 임포트:
`epmautomate importDataManagement inbox/dataFile.zip`
- 인박스 내 폴더에서 임포트:
`epmautomate importDataManagement inbox/dm_data/dataFile.zip`

importDimension

파일의 차원을 Oracle Fusion Cloud Enterprise Data Management 애플리케이션으로 임포트합니다.

이 명령은 Cloud EDM 또는 준비 영역에 정의된 연결에서 입력 파일을 임포트할 수 있습니다.

Cloud EDM 준비 영역에서 파일을 임포트할 경우 서비스 관리자는 [uploadFile](#) 명령을 사용하여 타겟 Cloud EDM 환경에 파일을 업로드합니다. 서비스 관리자는 [copyFileFromInstance](#) 명령을 사용하여 다른 Oracle Fusion Cloud Enterprise Performance Management 환경에서 파일을 복사할 수도 있습니다.

적용 대상

Cloud EDM.

필수 역할

- 서비스 관리자
- 데이터 관리자 권한이 있는 사용자

사용법

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

- APPLICATION은 Cloud EDM 애플리케이션의 이름입니다.
- DIMENSION은 임포트하는 애플리케이션 차원의 이름입니다.
- IMPORT_TYPE은 임포트를 수행하는 방법을 나타냅니다. 적합한 임포트 유형은 다음과 같습니다.
 - ResetDimension - 기존 차원 데이터를 모두 삭제하고 새 데이터를 임포트합니다.
 - ReplaceNodes - 노드를 추가 또는 업데이트하고 임포트 중에 기존 계층을 바꿉니다.
 - Merge - 임포트 요청을 사용하여 노드 및 계층에 대한 증분 변경사항을 처리합니다.
- FILE_NAME은 임포트할 차원 데이터가 포함된 파일(CSV 또는 ZIP)의 이름입니다. 파일 이름은 앞에 _(밑줄 문자)가 추가된 차원 이름으로 끝나야 합니다(예: import_Entity.csv). 여러 개의 임포트 파일이 포함된 ZIP 파일에서 임포트하는 경우 이 명령은 ZIP 파일 내의 파일 이름을 사용하여 올바른 임포트 파일을 확인합니다.
 connection 값을 지정한 경우 ZIP 파일(예: importdata_Entity.zip)에서 차원을 임포트해야 합니다.
- connection=NAME은 선택사항으로, Cloud EDM에 임포트 파일 위치로 정의된 연결 이름 (인스턴스 위치)을 나타냅니다. 지정하지 않으면 임포트 프로세스는 로컬 준비 영역에서 임포트 파일을 찾습니다.

예

- 준비 영역으로 업로드된 파일에서 임포트: `epmautomate importDimension USOperations Entity ReplaceNodes data_Entity.CSV`
- 다른 Cloud EPM 환경의 아웃박스에서 임포트: `epmautomate importDimension USOperations Entity ReplaceNodes data_Entity.ZIP Connection=Cloud-EPM_pln`

importEJDimensionMembers

Enterprise Journals 차원 멤버를 Financial Consolidation and Close 애플리케이션으로 임포트합니다.

이 명령은 지정하는 명령행 매개변수를 사용하여 임포트 작업을 생성합니다. 이 작업에 대한 정보는 작업 콘솔에서 사용할 수 있습니다.

적용 대상

재무 통합 및 마감

필수 역할

- 서비스 관리자

- 고급 사용자

사용법

`epmautomate importEJDimensionMembers Dimension_Name File_Name`
`[ImportMode=Replace|Update] [delimiter="DELIMITER"] [dateFormat=DATE_FORMAT]`. 설명:

- `DIMENSION_NAME`은 멤버가 임포트되는 Enterprise Journals 차원의 이름입니다.
- `FILE_NAME`은 `DIMENSION_NAME` 매개변수로 지정된 차원의 멤버 데이터를 포함하는 소스 CSV 파일의 이름입니다. [uploadFile](#) 명령을 사용하여 기본 업로드 위치에 이 파일을 업로드합니다.
- `ImportMode`는 선택사항으로, 임포트 작업에 사용할 모드입니다. 다음을 지정합니다.
 - `Update` - 소스 CSV 파일에 포함된 멤버만 변경합니다. 다른 멤버는 영향을 받지 않습니다.
 - `Replace` - 양식의 차원 멤버를 지우고 소스 CSV 파일의 멤버로 바꿉니다. 이 모드에서는 임포트 후 소스 파일에서 사용할 수 없는 값을 비워 둡니다. 이 모드가 기본 모드입니다.
- `delimiter`는 선택사항으로, 소스 CSV 파일에서 차원 멤버를 구분하는 데 사용하는 구분자를 나타냅니다. 기본값은 쉼표입니다(.). 탭 문자를 구분자로 사용하려면 `\t`를 지정합니다.
- `dateFormat`은 선택사항으로, 소스 CSV 파일에서 날짜 또는 날짜/시간 값(있는 경우)을 변환하는 데 사용할 날짜 형식을 지정합니다. 지원되는 날짜 형식은 `MM/dd/yyyy`, `dd/MM/yyyy`, `d-M-yyyy`, `dd-MMM-yy`, `MM-dd-yyyy` 및 `MMM d, yyyy`(기본 날짜 형식)입니다.

예

```
epmautomate importEJDimensionMembers Regions RegionMembers.csv ImportMode=Update
delimiter=\t dateFormat="dd/MM/yyyy"
```

importJobConsole

환경에서 익스포트된 작업 콘솔 레코드가 포함되어 있는 ZIP 파일을 사용하여 작업 콘솔 레코드를 복제합니다.

이 명령을 사용한 작업 콘솔 레코드 임포트는 [recreate](#) 명령을 실행한 후 수행해야 하는 일회성 태스크입니다. 이 명령을 사용하여 작업 콘솔 레코드를 이미 임포트한 경우 환경을 재생성할 때까지 후속 명령 호출이 실패합니다.

[exportJobConsole](#) 명령(`epmAutomate exportJobConsole FILE_NAME.zip nDays=All jobTypes=All jobStatusCode=All`)을 사용하여 이 명령의 입력으로 사용되는 ZIP 파일을 생성합니다.

적용 대상

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

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 작업 - 관리 애플리케이션 역할

사용법

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

- `FILE_NAME`은 임포트할 작업 콘솔 레코드가 포함된 ZIP 파일의 이름입니다. [uploadFile](#) 명령을 사용하여 환경에 이 파일을 업로드합니다.

- logFileName은 선택사항으로, jobConsoleLog를 импорт 중 발생한 오류가 기록될 로그 파일로 확인합니다. 이 값을 지정하지 않으면 명령은 usernameimportLog_date_timestamp.zip 규칙에 따라 이름이 지정된 오류 파일을 생성합니다. [downloadFile](#) 명령을 사용하여 이 파일을 다운로드할 수 있습니다.

예

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

importLibraryArtifact

아카이브 또는 파일의 라이브러리 아티팩트를 Narrative Reporting 라이브러리로 импорт합니다.

이 명령을 실행하기 전에 서비스 관리자는 [uploadFile](#) 명령을 사용하여 소스 아카이브 또는 파일을 환경에 로드합니다.

적용 대상

Narrative Reporting

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

고급 사용자, 사용자, 조회자 역할의 사용자에게는 ACL을 통해 추가 보안 권한이 부여되어야 합니다.

사용법

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

- SOURCE_FILE은 라이브러리로 импорт할 아티팩트가 포함된 아카이브의 이름입니다. 인박스에서 이 파일을 사용할 수 있어야 합니다.
- errorFile은 импорт 관련 오류를 저장할 텍스트 파일의 고유 이름입니다(선택사항).
- importFormat은 선택사항으로, 다음 중 하나입니다.
 - Native는 exportFormat=Native 옵션이 포함된 [exportLibraryArtifact](#) 명령을 사용하여 생성된 zip 파일의 아티팩트를 импорт합니다. 기본값입니다.
 - File은 이진 파일을 импорт합니다.

① 주

[importSnapshot](#) 명령을 사용하여 라이브러리 아티팩트 zip 파일(exportFormat=LCM 옵션이 포함된 [exportLibraryArtifact](#) 명령을 사용하여 생성)을 Financial Consolidation and Close, Planning, Planning 모듈 또는 Tax Reporting 환경으로 импорт할 수 있습니다.

- importFolder는 선택사항으로, импорт된 아티팩트를 저장할 라이브러리 위치입니다. 이 위치가 Library(기본 импорт 위치)와 다른 경우 이 경로를 지정합니다.

- `importPermission`은 아티팩트에 대해 설정된 액세스 권한을 임포트할지 여부를 나타냅니다. 기본값은 `False`입니다.
- `overwrite`는 지정한 라이브러리 위치에서 동일한 이름의 아티팩트(있는 경우)를 덮어쓰지 여부를 확인합니다. 기본값은 `False`이며, 동일한 이름의 아티팩트가 임포트 위치에 있으면 프로세스에서 아티팩트를 임포트하지 않습니다.

임포트가 완료되면 [downloadFile](#) 명령을 사용하여 오류 파일을 로컬 컴퓨터로 다운로드합니다.

예

- 이진 형식으로 파일 임포트:

```
epmautomate importLibraryArtifact newReports.doc
errorFile=report_imp_errors.txt importFormat=File importFolder="Library/My
Reports" importPermission=true overwrite=true
```
- 익스포트된 zip 파일에서 아티팩트 임포트:

```
epmautomate importLibraryArtifact newReports.zip
errorFile=report_imp_errors.txt importFormat=Native importFolder="Library/My
Reports" importPermission=true overwrite=true
```
- 익스포트된 zip 파일에서 Financial Consolidation and Close, Planning, Planning 모듈 또는 Tax Reporting 환경으로 보고서 임포트:

```
epmautomate importSnapshot newReports.zip
```

importLibraryDocument

ZIP 파일에 포함된 하나 이상 문서를 Planning, Financial Consolidation and Close, Enterprise Profitability and Cost Management 또는 Tax Reporting 라이브러리의 특정 위치로 임포트합니다.

이 명령을 사용하여 임포트할 수 있는 문서에는 Microsoft Word 문서(.DOC, .DOCX), Excel 문서(.XLS, .XLSX), Power Point 프리젠테이션(.PPT, .PPTX), 심표로 구분된 값 파일(.CSV), PDF 파일(크기 최대 100MB), 텍스트 파일(.TXT) 및 파일 아카이브(.ZIP, 크기 최대 100MB)가 포함됩니다. 이 명령을 실행하기 전에 임포트 마이그레이션 화면 또는 EPM Automate 명령(예: [uploadFile](#), [copyFileFromInstance](#) 및 [copyFromObjectStorage](#))을 사용하여 소스 아카이브 또는 파일을 환경에 로드합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

```
epmautomate importLibraryDocument SOURCE_FILE DESTINATION [jobName=JOB_NAME]
[description=DESCRIPTION][overwrite=true|false][deleteAfterImport=true|false]
[errorFile=FILE_NAME.log]. 설명:
```

- `SOURCE_FILE`은 라이브러리로 임포트할 문서가 하나 이상 포함된 문서 또는 아카이브의 이름입니다.
- `DESTINATION`은 아카이브의 파일 또는 콘텐츠를 임포트할 대상 폴더의 정규화된 경로 또는 UUID(Universally Unique Identifier)입니다.
- `jobName`은 선택사항으로, 문서를 임포트하는 데 사용할 문서 임포트 라이브러리 작업의 이름입니다. 값을 지정하지 않으면 기본적으로 문서를 라이브러리로 임포트 작업이 사용됩니다.

- `description`은 선택사항으로, 임포트되는 문서에 대한 설명입니다.
- `overwrite`는 선택사항으로, 대상 폴더의 기존 문서를 소스 문서와 동일한 이름으로 덮어쓸지 여부를 확인합니다. 기본값은 `false`입니다.
- `deleteAfterImport`는 선택사항으로, 임포트 프로세스가 완료된 후 소스 파일을 삭제할지 여부를 지정합니다. 기본값은 `false`입니다.
- `errorFile`은 선택사항으로, 명령 실행 프로세스 중 보고된 정보를 캡처할 파일의 이름입니다. 이 파일은 아웃박스에 생성되며 `.log` 확장자를 사용합니다. [downloadFile](#) 명령을 사용하여 이 파일을 로컬 컴퓨터로 다운로드합니다.

예

```
epmautomate importLibraryDocument "FinancialReport2024.xlsx" "Library/Insights"
jobName=ImportSingleDocument description="Annual Financial Report 2024"
overwrite=true deleteAfterImport=true errorFile=FinancialReport2024ImportLog.log
```

importLogoImage

Account Reconciliation 환경에서 사용된 기업 로고를 익스포트 파일에서 다른 환경으로 임포트합니다.

적용 대상

Account Reconciliation

필수 역할

서비스 관리자

사용법

`epmautomate importLogoImage IMAGE_NAME.jpg`, 설명: `IMAGE_NAME`은 로고 이미지 파일의 이름입니다. [downloadFile](#) 명령을 사용하여 익스포트된 이미지를 다운로드할 수 있습니다. [uploadFile](#) 명령을 사용하여 타겟 환경에 업로드한 후 [importLogoImage](#) 명령을 실행하여 임포트합니다.

예

```
epmautomate importLogoImage corpLogo.jpg
```

importMapping

이전에 환경으로 업로드된 매핑 임포트 파일에서 매핑을 임포트합니다.

서비스 관리자는 [uploadFile](#) 명령을 사용하여 Data Management 인박스 또는 인박스 내 폴더에 파일을 업로드합니다.

적용 대상

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

필수 역할

- 서비스 관리자

- 고급 사용자

사용법

epmautomate importMapping *DIMENSION_NAME/ALL FILE_NAME IMPORT_MODE VALIDATION_MODE LOCATION* 설명:

- *DIMENSION_NAME/ALL*은 매핑 수신자를 나타냅니다. 매핑을 가져올 차원의 이름을 지정하거나, 파일에 포함된 모든 매핑 파일을 적절한 차원으로 가져오려면 ALL을 지정합니다.
- *FILE_NAME*은 Data Management 인박스 또는 하위 디렉토리에서 사용할 수 있는 매핑 임포트 파일의 이름 및 위치입니다. 파일 이름(표준 Data Management 형식의 TXT 파일) 및 해당 경로(예: inbox/AccountMap.txt 또는 inbox/pbcs_maps/AccountMap.txt)를 지정합니다.
- *IMPORT_MODE*는 매핑을 임포트하기 전에 기존 매핑 규칙을 지우는 REPLACE 또는 기존 규칙에 새 매핑 규칙을 추가하는 MERGE입니다.
- *VALIDATION_MODE*는 애플리케이션에 대해 타겟 멤버를 검증하는 TRUE 또는 검증을 실행하지 않고 매핑 파일을 로드하는 FALSE입니다.
- *LOCATION*은 매핑 규칙을 로드해야 하는 Data Management 위치입니다.

예

- epmautomate importMapping Account inbox/AccountMap.txt MERGE FALSE "France Sales"
- epmautomate importMapping ALL "inbox/France Sales/AllMaps.txt" MERGE FALSE "France Sales"(매핑 임포트 파일의 매핑을 France Sales 위치의 매핑된 모든 차원에 로드)

importMetadata

import metadata 유형의 작업에 지정된 임포트 설정을 사용하여 메타데이터를 애플리케이션으로 임포트합니다. 선택적으로, 가져올 메타데이터가 있는 ZIP 파일의 이름을 지정할 수 있습니다.

[uploadFile](#) 명령을 사용하여 메타데이터가 포함된 파일을 기본 업로드 위치로 업로드할 수 있습니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

epmautomate importMetadata *JOB_NAME [FILE_NAME] errorFile=ERROR_FILE.zip* 설명:

- *JOB_NAME*은 애플리케이션에서 정의된 작업 이름입니다.
- *FILE_NAME*은 임포트할 메타데이터가 있는 ZIP 파일의 이름을 확인합니다(선택사항). 지정하면 이 ZIP 파일의 콘텐츠가 작업에 정의된 파일 이름보다 우선합니다. ZIP 파일에는 하나 이상의 CSV 파일이 포함될 수 있습니다. 차원의 메타데이터를 포함하는 파일 이름은 작업에 정의된 임포트 파일 이름과 일치하거나 *_DIMENSIONNAME.csv*로 끝나야 합니다(예: metadata_Entity.csv, metadata_HSP_Smart Lists.csv, metadata_Exchange Rates.csv).
- 선택사항인 *errorFile*은 임포트 작업 중에 거부된 레코드(있는 경우)가 기록되는 ZIP 파일의 이름을 확인합니다. 아웃박스에서 동일한 이름의 ZIP 파일(있는 경우)을 덮어씁니다. [downloadFile](#) 명령을 사용하여 이 파일을 다운로드할 수 있습니다.

① 주

- old_name 또는 unique_name 등록정보가 수정된 로드 파일로 메타데이터 임포트 작업을 실행하여 멤버 이름을 바꿀 수 없습니다. 멤버 이름 바꾸기는 무시됩니다.
- 이 명령을 사용하여 메타데이터를 임포트하는 동안에는 속성 차원을 삭제할 수 없습니다.
- 메타데이터 임포트가 작업에 설정된 차원의 메타데이터만 가져옵니다. 다른 차원이 ZIP 파일에 포함된 경우 해당 메타데이터는 무시됩니다.

ZIP 파일이 다음 조건에 둘 다 해당하는 경우 모호한 임포트 상황이 생성됩니다.

- ZIP에 포함된 메타데이터 파일의 이름이 작업에 정의된 파일 이름과 일치합니다.
- ZIP에 포함된 메타데이터 파일의 이름이 `_DIMENSIONNAME.CSV` 또는 `_DIMENSIONNAME.TXT`로 끝납니다. 여기서 `DIMENSIONNAME`은 메타데이터를 임포트하는 차원의 이름입니다.

ZIP 파일에 작업에 참조된 이름과 동일한 이름의 메타데이터 파일이나 이름이 `_DIMENSIONNAME.CSV`(또는 `_DIMENSIONNAME.TXT`)로 끝나는 파일 중 하나만 포함하고 둘 다 포함하지 않는 것이 좋습니다. 예를 들어 `Employees_A-Z.CSV` 메타데이터 파일을 참조하는 작업을 `Employees` 차원에 로드하는 경우 ZIP 파일에 `Employees_A-Z.CSV` 또는 `New_Employees.CSV` 중 하나만 포함할 수 있고 둘 다 포함하면 안 됩니다. ZIP에 `Employees_A-Z.CSV` 및 `New_Employees.CSV`가 포함되어 있는 경우 EPM Automate가 ZIP의 파일 순서에 따라 임포트할 파일을 선택할 수 있습니다. `Employees_A-Z.CSV` 파일은 파일 이름이 작업에 참조된 파일 이름과 일치하기 때문에 임포트할 일치 항목일 수 있습니다. `New_Employees.CSV`도 파일 이름이 `_DIMENSIONNAME.CSV` 패턴과 일치하기 때문에 일치 항목일 수 있습니다.

예

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

importOwnershipData

해당 환경의 사용가능한 CSV 파일에서 기간으로 소유권 데이터를 임포트합니다.

이 명령을 실행하기 전에 서비스 관리자는 [uploadFile](#) 명령을 사용하여 임포트 소스 CSV 파일을 환경에 로드합니다. 이 CSV 파일의 머리글은 다음과 같습니다.

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

POwn, Control 및 Method 값은 선택사항입니다.

임포트한 소유권 데이터는 기존 데이터와 병합되어 부적합한 소유권 항목을 생성할 수도 있습니다. 엔티티가 두 개 이상의 계층 분기에 있는 경우 임포트한 소유권 데이터로 인해 엔티티의 조합된 소유권 %가 100%를 초과할 수 있습니다. 소유권 %를 수동으로 정정하여 100%를 초과하지 않도록 해야 합니다.

적용 대상

Financial Consolidation and Close 및 Tax Reporting.

필수 역할

- 서비스 관리자

- 고급 사용자
- 엔티티에 대한 쓰기 액세스 권한이 있는 사용자

사용법

`epmautomate importOwnershipData Scenario Year Period FILE_NAME` 설명:

- `Scenario`는 소유권 데이터를 임포트할 시나리오입니다.
- `Year`는 데이터를 임포트할 연도입니다.
- `Period`는 소유권 데이터를 임포트할 연도의 기간입니다.
- `FILE_NAME`은 임포트할 데이터가 있는 CSV 파일의 이름입니다.

예

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

importPreMappedBalances

Account Reconciliation 저장소의 파일에서 사전 매핑된 잔액 데이터를 임포트합니다.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

`epmautomate importPreMappedBalances PERIOD FILE_NAME BALANCE_TYPE CURRENCY_BUCKET`
설명:

- `PERIOD`는 기간 이름입니다.
- `FILE_NAME`은 임포트할 데이터가 포함된 CSV 파일의 이름입니다.
- `BALANCE_TYPE`은 SRC 또는 SUB입니다.
- `CURRENCY_BUCKET`은 Entered, Functional 또는 Reporting입니다.

예

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

importPreMappedTransactions

Account Reconciliation 저장소의 CSV 파일에서 사전 매핑된 트랜잭션을 임포트합니다.

적용 대상

Account Reconciliation

필수 역할

서비스 관리자, 고급 사용자, 사용자, 조회자
사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

`epmautomate importPreMappedTransactions PERIOD TRANSACTION_TYPE FILE_NAME DATE_FORMAT` 설명:

- `PERIOD`는 기간 이름입니다.
- `TRANSACTION_TYPE`는 다음 중 하나입니다.
 - BEX - 잔액 설명을 로드하는 데 사용됩니다.
 - SRC - 소스 시스템 조정을 로드하는 데 사용됩니다.
 - SUB - 하위 시스템 조정을 로드하는 데 사용됩니다.
 - VEX - 차이 분석 설명을 로드하는 데 사용됩니다.
- `FILE_NAME`는 임포트할 데이터가 있는 CSV 파일의 이름입니다
- `DATE_FORMAT`은 날짜 형식 텍스트 문자열입니다(예: MMM d, yyyy).

예

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

importProfiles

Account Reconciliation 저장소의 CSV 파일에서 새 프로파일 정의를 임포트합니다.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

epmautomate importProfiles *FILE_NAME PROFILE_TYPE METHOD DATE_FORMAT* 설명:

- *FILE_NAME*은 임포트할 데이터가 있는 CSV 파일의 이름입니다
- *PROFILE_TYPE*은 profiles 또는 children입니다.
- *METHOD*는 Replace 또는 Update입니다.
- *DATE_FORMAT*은 날짜 형식 텍스트 문자열입니다(예: MMM d, yyyy).

예

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

importRates

Account Reconciliation 저장소의 CSV 파일에서 환율을 임포트합니다.

적용 대상

Account Reconciliation

필수 역할

서비스 관리자, 고급 사용자, 사용자, 조회자
사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

epmautomate importRates *PERIOD RATE_TYPE REPLACE_MODE FILE_NAME* 설명:

- *PERIOD*는 기간 이름입니다.
- *RATE_TYPE*은 사전 정의된 환율 유형입니다.
- *REPLACEMENT_MODE*는 Replace 또는 ReplaceAll입니다.
- *FILE_NAME*은 임포트할 환율이 있는 CSV 파일의 이름입니다

예

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

importRCAttributeValues

Account Reconciliation 조정 준수 또는 그룹 속성으로 속성 값을 임포트합니다.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자(ACL을 통해 제공되는 추가 보안 필요)

사용법

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

- *ATTRIBUTE_NAME*은 값을 임포트할 타겟 목록 또는 그룹 속성의 이름입니다.
- *FILE_NAME*은 값을 임포트할 소스 CSV 임포트 파일입니다. 해당 명령을 실행하기 전에 [uploadFile](#) 명령을 사용하여 이 파일을 환경에 업로드합니다.
- *METHOD*는 선택사항으로, 값을 임포트하는 방법입니다. 적합한 값은 다음과 같습니다.
 - Replace - 임포트 파일의 모든 값을 조정 준수의 속성 값으로 추가합니다. 기존 속성 값은 임포트 파일의 값으로 대체됩니다. 속성에 없지만 임포트 파일에 있는 값이 추가됩니다. 임포트 파일에는 없고 속성에 있는 값은 변경되지 않습니다. 특정 키 값에 대한 모든 속성 데이터는 파일의 콘텐츠로 대체되거나 지워집니다. 새 값은 파일에 표시되는 순서대로 맨아래에 추가됩니다.
이 임포트 유형은 소스 시스템에서 최근 변경사항만 이동하는 경우(예: 임포트 파일의 값으로 지정된 속성 값(있는 경우)만 바꾸기 위해 취득 과정에서 새 스토어 데이터를 추가하는 경우) 가장 유용합니다. 이것이 기본값입니다.
 - Replace All - 기존 속성 값을 임포트의 값으로 바꿉니다. 속성에 있지만 임포트 파일에는 없는 값은 삭제됩니다.
이 임포트 유형은 전체 업데이트를 통해 소스 시스템의 값을 미러링하는 경우(예: 매주 업데이트를 완료하여 소스 시스템의 스토어 데이터와 동기화하는 경우) 가장 유용합니다.
 - Update - 임포트 파일의 모든 값을 대체하거나 속성에 추가합니다. 기존 속성 값은 임포트 파일의 값으로 대체됩니다. 임포트 파일에 있지만 속성에는 없는 값이 추가됩니다. 속성에 있지만 임포트 파일에는 없는 값은 변경되지 않습니다. 특정 키 값에 대한 속성 데이터만 파일의 콘텐츠로 대체됩니다. 파일에서 사용할 수 없는 속성에 대한 데이터는 변경되지 않습니다. 임포트 파일에 있는 키가 속성에는 없으면 오류가 발생합니다.
예를 들어 나머지 저장소 데이터에 영향을 주지 않고 재구성 후 저장소 관리자를 업데이트하는 동안 이 유형의 임포트는 모든 속성 값 중에서 몇몇 속성을 업데이트하는 데 가장 유용합니다.
- *Dateformat*은 선택사항으로, 구문분석할 적합한 날짜 형식을 지정합니다(예: DD/MM/YYYY, DD-MMM-YYYY(기본값), MMM d,yyyy, All). 세미콜론으로 구분하여 여러 날짜 형식 값을 지정할 수 있습니다.

예

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

importReconciliationAttributes

서비스 관리자가 [uploadFile](#) 명령을 사용하여 Account Reconciliation 환경으로 업로드한 파일에서 기존 조정으로 조정 속성을 임포트합니다.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자

- 조회자

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

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

- `FILE`은 조정으로 импорт할 조정 속성이 포함된 CSV 파일의 이름입니다.
- `Period`는 조정이 속한 기간을 나타냅니다.
- `Rules`는 선택사항으로, 속성을 импорт한 후 영향을 받는 조정에서 실행할 규칙을 나타냅니다. 여러 규칙 이름을 구분하는 데는 쉼표를 사용합니다. 적합한 값은 다음과 같습니다.
 - `None`: 영향을 받는 조정에서 규칙을 실행하지 않습니다. 기본값이며, 이 값은 다른 값과 결합해서는 안 됩니다.
 - `ALL`: 지정된 기간 동안 조정에 대해 정의된 모든 규칙을 실행합니다. 이 값은 자체적으로 사용되어야 하며, 다른 규칙 이름과 결합해서는 안 됩니다.
 - `SET_ATTR_VAL`: 사전 정의된 규칙을 실행하여 속성 값을 설정합니다.
 - `CRT_ALT`: 사전 정의된 규칙을 실행하여 알림을 생성합니다.
 - `AUTO_APP`: 사전 정의된 규칙을 실행하여 자동으로 조정을 승인합니다.
 - `AUTO_SUB`: 사전 정의된 규칙을 실행하여 자동으로 조정을 제출합니다.
 - `EMAIL_ON_SAVE`: 조정이 업데이트되면 사전 정의된 규칙을 실행하여 자동으로 전자메일을 전송합니다.
- `Reopen`은 선택사항으로, импорт 작업 완료 시 변경된 조정을 다시 열지 여부를 지정합니다. 기본값은 `false`입니다.
- `Dateformat`은 선택사항으로, 적합한 날짜 형식(예: `MM-dd-yyyy`, `dd-MMMM-yy`, `MMM d` 및 `yyyy`)을 지정하여 구문분석합니다. 세미콜론으로 구분하여 여러 날짜 형식 값을 지정할 수 있습니다.

예

- **기간의 속성 값을 импорт하고 여러 날짜 형식으로 여러 규칙을 실행합니다.**

```
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"
```
- **규칙을 실행하지 않고 기간의 속성 값을 импорт합니다.**

```
epmAutomate importReconciliationAttributes Reconciliations.csv "July 2020"
```
- **기간의 속성 값을 импорт하고, 적용 가능한 모든 규칙을 실행하고, 영향을 받는 조정을 다시 엽니다.**

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

importSDMDimensionMembers

Supplemental Data Manager 차원 멤버를 Financial Consolidation and Close 애플리케이션으로 импорт합니다.

이 명령은 지정하는 명령행 매개변수를 사용하여 импорт 작업을 생성합니다. 이 작업에 대한 정보는 작업 콘솔에서 사용할 수 있습니다.

적용 대상

Financial Consolidation and Close, Tax Reporting

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

```
epmautomate importSDMDimensionMembers DIMENSION_NAME FILE_NAME
```

[ImportMode=Replace|Update] [delimiter=DELIMITER] [dateFormat=DATE_FORMAT]. 설명:

- *DIMENSION_NAME*은 멤버가 임포트되는 Supplemental Data Manager 차원의 이름입니다.
- *FILE_NAME*은 *DIMENSION_NAME* 매개변수로 지정된 차원의 멤버 데이터를 포함하는 소스 CSV 파일의 이름입니다. [uploadFile](#) 명령을 사용하여 기본 업로드 위치에 이 파일을 업로드합니다.
- ImportMode는 선택사항으로, 작업에 사용할 모드입니다. 다음을 지정합니다.
 - Update - 소스 CSV 파일에 포함된 멤버만 변경합니다. 다른 멤버는 영향을 받지 않습니다.
 - Replace - 양식의 차원 멤버를 지우고 소스 CSV 파일의 멤버로 바꿉니다. 이 모드에서는 임포트 후 소스 파일에서 사용할 수 없는 값을 비워 둡니다. 이 모드가 기본 모드입니다.
- delimiter는 선택사항으로, 소스 CSV 파일에서 차원 멤버를 구분하는 데 사용하는 구분자를 나타냅니다. 기본값은 쉼표입니다(.). 탭 문자를 구분자로 사용하려면 `\t`를 지정합니다.
- dateFormat은 선택사항으로, 소스 CSV 파일에서 날짜 또는 날짜/시간 값(있는 경우)을 변환하는 데 사용할 날짜 형식을 지정합니다. 지원되는 날짜 형식은 MM/dd/yyyy, dd/MM/yyyy, d-M-yyyy, dd-MMM-yy, MM-dd-yyyy 및 MMM d, yyyy(기본 날짜 형식)입니다.

예

```
epmautomate importSDMDimensionMembers Regions RegionMembers.csv delimiter=\t
dateFormat=dd/MM/yyyy
```

importSnapshot

스냅샷의 콘텐츠를 환경으로 임포트합니다. 가져오는 스냅샷은 기본 업로드 위치에 있어야 합니다.

서비스 관리자는 [uploadFile](#) 명령을 사용하여 스냅샷을 업로드하거나 [copySnapshotFromInstance](#) 명령을 사용하여 다른 인스턴스에서 스냅샷을 복사합니다.

- 다음은 Planning, Planning 모듈, 무형식 애플리케이션 스냅샷에 포함되지 않습니다.
 - 감사 데이터
 - 작업 콘솔 데이터

감사 및 작업 콘솔 데이터를 타겟 환경에 복사하려는 경우 [cloneEnvironment](#) 명령이나 환경 복제 기능을 사용합니다.

Planning 비즈니스 프로세스에 사용자정의 기간 멤버로 대체된 이름이 변경된 시드된 기간 멤버가 포함된 경우 스냅샷 임포트가 실패할 수 있습니다. 예를 들어, 시드된 *YearTotal* 기간 멤버를 *unused_YearTotal*로 이름을 변경한 다음 대체 유형 기간 멤버를 원래 시드된 멤버 이름(이 예에서 *YearTotal*)로 추가했습니다. 이 시나리오에서는 스냅샷 임포트가 실패할 수 있습니다.

- 스냅샷에는 Data Management 스테이지 테이블 데이터가 포함되지 않습니다. 이 데이터를 임포트하려면 [exportDataManagement](#) 및 [importDataManagement](#) 명령이나 Data

Management 시스템 유지관리 스크립트 인터페이스를 사용합니다. [cloneEnvironment](#) 명령이나 환경 복제 기능을 사용하여 Data Management 스테이지 테이블 데이터를 비롯한 환경의 동일한 복사본을 생성할 수 있습니다.

이 명령을 사용하여 완료할 수 있는 작업은 역할에 따라 달라집니다.

- 서비스 관리자는 애플리케이션 아티팩트만 서비스 환경으로 임포트할 수 있습니다.
- 애플리케이션 콘텐츠를 서비스 환경으로 임포트하고 ID 도메인 아티팩트(사용자 및 사전 정의된 역할 지정)를 환경의 ID 도메인으로 임포트하려면 서비스 관리자 및 ID 도메인 관리자 역할이 둘 다 필요합니다.
임포트되는 스냅샷에서 ID 도메인에 없는 사용자가 참조되는 경우 이 명령은 ID 도메인에 사용자를 생성하고 명령에 지정된 기본 비밀번호를 지정합니다. 명령에 비밀번호가 지정되지 않은 경우 각 사용자에게 고유한 비밀번호를 지정합니다. 기본적으로 사용자는 처음 사인인할 때 비밀번호를 재설정해야 합니다.

① 주

- Account Reconciliation, Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management 이외의 비즈니스 프로세스: 아웃라인에서 공유 멤버가 기준 멤버 앞에 표시되므로 이전 시도에서 거부된 레코드가 발생한 경우 Oracle Fusion Cloud Enterprise Performance Management에서 메타데이터를 로드하는 동안 로드 통과를 여러 번 수행할 수 있습니다. 이러한 시도로 인해 명령 처리 시간이 증가할 수 있습니다.
- 액세스 제어에서 그룹의 멤버인 사용자는 사전 정의된 역할에 지정되어야 합니다. 사전 정의된 역할에 지정되지 않은 사용자는 그룹에 지정할 수 없습니다.

적용 대상

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

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

`importUsers=true` 옵션을 사용하여 사용자 및 사전 정의된 역할 지정을 임포트하려면 ID 도메인 관리자 역할이 필요합니다.

사용법

```
epmautomate importSnapshot SNAPSHOT_NAME [importUsers=true|false]
[userPassword=DEFAULT_PASSWORD] [resetPassword=true|false] 설명:
```

- `SNAPSHOT_NAME`은 기본 업로드 위치에 있는 스냅샷의 이름입니다.
- `importUsers`는 선택사항으로, 스냅샷에서 사용자 및 사전 정의된 역할 지정을 임포트할 것인지 지정합니다. 기본값은 `false`입니다. 소스 스냅샷에 새 사용자에게 대한 데이터가 포함된 경우 또는 현재 사용자에게 새 역할이 지정된 경우 `importUsers=true`를 사용하여 사용자 및 사전 정의된 역할 지정을 ID 도메인으로 임포트합니다.
사용자 로그인 값은 대소문자를 구분하지 않습니다. 예를 들어 사용자 로그인 값 `jane.doe@example.com`은 `Jane.Doe@Example.com` 또는 대소문자의 모든 변형과 동일한 것으로 처리됩니다. 변형이 ID 도메인에 있는 사용자 로그인과 일치하는 경우 이 명령은 스냅샷에서 사용자를 임포트하지 않습니다.

① 주

- ID 도메인 관리자가 아닌 사용자가 임포트 작업을 수행하면 사용자 및 사전 정의된 역할 임포트가 실패합니다. 마이그레이션 상태 보고서에 다음 오류가 기록됨: 외부 디렉토리 아티팩트 `ARTIFACT_NAME` 임포트 실패. `USER_NAME` 사용자는 이 작업을 수행할 수 있는 권한이 없습니다. 이 작업을 수행하려면 사용자에게 ID 도메인 관리자 역할이 있어야 합니다.
- 사용자를 임포트하지 않고 소스 스냅샷의 사용자가 타겟 환경의 사전 정의된 역할에 지정되지 않은 경우 오류(EPMIE-00070: 지정된 역할 임포트 중 사용자를 찾지 못했습니다.)가 표시됩니다.

- 사용자의 사전 정의된 역할 변경사항은 소스 스냅샷에서 지정된 역할에 따라 업데이트됩니다. 그러나 타겟의 역할 지정이 소스 스냅샷의 역할 지정과 일치하도록 제거되지는 않습니다. 예를 들어 `jdoe`가 타겟 환경에서는 고급 사용자 사전 정의된 역할에 지정되었지만 소스 스냅샷에서는 사용자 역할만 있다고 가정합니다. 이런 상황에서 이 명령은 사용자 역할에 `jdoe`를 지정하고 타겟 환경의 고급 사용자 역할 지정을 제거하지 않습니다.
- 이 명령은 소스 스냅샷에 없는 경우에도 타겟 환경의 기존 사용자를 삭제하지 않습니다. 예를 들어 `jdoe`는 타겟 환경에 계정이 있지만 소스 스냅샷에는 이 계정이 없습니다. 이런 상황에서 타겟 환경의 `jdoe` 계정은 삭제되지 않습니다.
- 이 명령은 타겟 환경에 없는 사용자를 추가합니다. 소스 스냅샷과 다른 경우에도 타겟 환경의 현재 사용자 등록정보를 업데이트하지 않습니다. 예를 들어 소스 스냅샷에 있는 `jdoe`의 성찰자가 타겟 환경과 다른 경우에도 타겟 환경에서 변경되지 않습니다.
- 이 명령은 소스 스냅샷과 다른 경우에도 타겟 환경의 기존 사용자 비밀번호를 변경하지 않습니다.
- `userPassword`는 선택사항으로, ID 도메인에 생성된 새 사용자에게 지정하는 기본 비밀번호를 나타냅니다. 지정한 비밀번호가 최소 비밀번호 요구사항을 충족해야 합니다. 이 매개변수 값을 지정하지 않으면 각 사용자에게 고유한 임시 비밀번호가 지정됩니다.
- `resetPassword`는 선택사항으로, 새 사용자가 처음 로그인 시 비밀번호를 변경해야 하는지 여부를 나타냅니다. 기본값은 `true`이며, 새 사용자가 처음 로그인 시 비밀번호를 변경해야 합니다. 이 값을 `true`로 설정하면 새 사용자는 비밀번호를 변경하도록 요청하는 계정 활성화 전자메일을 받게 됩니다.

예

- 애플리케이션 아티팩트 임포트만 해당: `epmautomate importSnapshot April16FullApp`
- 애플리케이션 및 ID 도메인 아티팩트 임포트(Service Administrator 및 Identity Domain Administrator 역할 필요):
 - 각각의 새 사용자에게 고유한 임시 비밀번호를 지정하고, 사용자가 처음 로그인한 후 비밀번호를 재설정하도록 강제 적용합니다.
`epmautomate importSnapshot April16FullApp importUsers=true`
 - 특정 비밀번호를 지정하고 사용자가 선택하는 경우 이 비밀번호를 변경하지 않을 수 있도록 합니다. 프로덕션 환경으로 임포트하는 데는 사용하지 않는 것이 좋습니다.
`epmautomate importSnapshot April16FullApp importUsers=true
userPassword=P@ssw0rd1 resetPassword=false`

importSupplementalCollectionData

파일의 보충 컬렉션 데이터를 애플리케이션으로 임포트합니다.

`uploadFile` 명령을 사용하여 데이터가 포함된 파일을 기본 업로드 위치로 업로드할 수 있습니다. 임포트 파일 형식은 다음과 같습니다.

```
#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
```

예를 들어 다음과 같습니다.

```
#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
```

적용 대상

Financial Consolidation and Close, 및 Tax Reporting.

필수 역할

서비스 관리자

사용법

① 주

모든 명령 매개변수는 큰따옴표로 묶어야 합니다.

```
epmautomate importSupplementalCollectionData "FILE_NAME" "COLLECTION_NAME" "YEAR"
"PERIOD" "[FREQUENCY_DIMENSION=MEMBER]" 설명:
```

- `FILE_NAME`은 올바른 형식의 보충 데이터를 포함하며, 기본 업로드 위치에서 사용할 수 있는 CSV 파일의 이름입니다.
- `COLLECTION_NAME`은 파일에 있는 보충 데이터를 임포트해야 하는 컬렉션의 이름입니다.
- `YEAR`는 컬렉션에 사용할 연도 차원 멤버입니다.
- `PERIOD`는 컬렉션에 사용할 기간 차원의 이름입니다.
- `FREQUENCY_DIMENSION`은 수집에 사용할 빈도 차원의 이름입니다(선택사항). 빈도 차원을 필요한 개수만큼 "`FREQUENCY_DIMENSION1=MEMBER`" "`FREQUENCY_DIMENSION2=MEMBER`" 형식으로 지정할 수 있습니다.

예

```
epmautomate importSupplementalCollectionData "datafile.csv" "Journal Data
Collection" "FY20" "Jan" "Account=PAYROLL" "JournalID=LNR 113"
```

importSupplementalData

파일의 보충 데이터를 애플리케이션으로 임포트합니다.

[uploadFile](#) 명령을 사용하여 데이터가 포함된 파일을 기본 업로드 위치로 업로드할 수 있습니다.

적용 대상

Financial Consolidation and Close 및 Tax Reporting.

필수 역할

서비스 관리자

사용법

① 주

모든 명령 매개변수는 큰따옴표로 묶어야 합니다.

```
epmautomate importSupplementalData "FILE_NAME" "DATA_SET_NAME" "YEAR"
"PERIOD_NAME" "SCENARIO_NAME" 설명:
```

- FILE_NAME은 올바른 형식의 보충 데이터를 포함하며, 기본 업로드 위치에서 사용할 수 있는 CSV 파일의 이름입니다.
- DATA_SET_NAME은 파일에 있는 보충 데이터를 임포트해야 하는 데이터 세트의 이름입니다.
- YEAR는 데이터 세트가 배포된 연도입니다.
- PERIOD_NAME은 데이터 세트가 배포된 기간의 이름입니다.
- SCENARIO_NAME은 데이터 세트가 배포된 시나리오의 이름입니다.

예

```
epmautomate importSupplementalData "DatasetImport.csv" "EmployeeDataSet" "FY17"
"Jan" "Actual"
```

importTemplate

profitinbox에 있는 템플릿 파일에서 가져와 애플리케이션 구조를 생성합니다.

서비스 관리자는 [uploadFile](#) 명령을 사용하여 템플릿 파일을 profitinbox로 업로드합니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

```
epmautomate importTemplate APPLICATION_NAME File_Name
isApplicationOverwrite=true|false 설명:
```

- *APPLICATION_NAME*은 템플릿을 임포트하여 생성할 Profitability and Cost Management 애플리케이션의 이름입니다.
- *File_Name*은 애플리케이션 템플릿이 포함된 .ZIP 파일의 이름입니다. 이 파일은 profitinbox에 있어야 합니다.
- *isApplicationOverwrite*는 기존 애플리케이션(있는 경우)을 덮어쓸지 여부를 지정합니다. 이 매개변수 값을 모두 소문자로 지정합니다.

예

```
epmautomate importTemplate BksML12 template1.zip isApplicationOverwrite=true
```

importTMAttributeValues

Account Reconciliation 트랜잭션 일치 그룹 속성으로 값으로 임포트합니다.

서비스 관리자는 이 명령을 실행하기 전에 [uploadFile](#) 명령을 사용하여 트랜잭션 일치 속성 값이 포함된 임포트 파일을 환경에 업로드합니다.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자(ACL을 통해 제공되는 추가 보안 필요)

사용법

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

- *ATTRIBUTE_NAME*은 값을 임포트할 타겟 그룹 속성의 이름입니다.
- *FILE_NAME*은 값을 트랜잭션 일치로 임포트할 소스 CSV 임포트 파일입니다.
- *METHOD*는 선택사항으로, 값을 임포트하는 방법입니다. 적합한 값은 다음과 같습니다.
 - Replace - 임포트 파일의 모든 값을 트랜잭션 일치 그룹 속성에 추가합니다. 기존 속성 값은 임포트 파일의 값으로 대체됩니다. 속성에 없지만 임포트 파일에 있는 값이 추가됩니다. 임포트 파일에는 없고 속성에 있는 값은 변경되지 않습니다. 특정 키 값에 대한 모든 속성 데이터는 파일의 콘텐츠로 대체되거나 지워집니다. 새 값은 파일에 표시되는 순서대로 맨아래에 추가됩니다.
이 임포트 유형은 소스 시스템에서 최근 변경사항만 이동하는 경우(예: 임포트 파일의 값으로 지정된 속성 값(있는 경우)만 바꾸기 위해 취득 과정에서 새 스토어 데이터를 추가하는 경우) 가장 유용합니다. 이것이 기본값입니다.
 - Replace All - 기존 속성 값을 임포트의 값으로 바꿉니다. 속성에 있지만 임포트 파일에는 없는 값은 삭제됩니다.
이 임포트 유형은 전체 업데이트를 통해 소스 시스템의 값을 미러링하는 경우(예: 매주 업데이트를 완료하여 소스 시스템의 스토어 데이터와 동기화하는 경우) 가장 유용합니다.

- Update - импорт 파일의 모든 값을 대체하거나 속성에 추가합니다. 기존 속성 값은 импорт 파일의 값으로 대체됩니다. импорт 파일에 있지만 속성에는 없는 값이 추가됩니다. 속성에 있지만 импорт 파일에는 없는 값은 변경되지 않습니다. 특정 키 값에 대한 속성 데이터만 파일의 콘텐츠로 대체됩니다. 파일에서 사용할 수 없는 속성에 대한 데이터는 변경되지 않습니다. импорт 파일에 있는 키가 속성에는 없으면 오류가 발생합니다. 예를 들어 나머지 저장소 데이터에 영향을 주지 않고 재구성 후 저장소 관리자를 업데이트하는 동안 이 유형의 임포트는 모든 속성 값 중에서 몇몇 속성을 업데이트하는 데 가장 유용합니다.
- Dateformat은 선택사항으로, 구문분석할 적합한 날짜 형식을 지정합니다(예: DD/MM/YYYY, DD-
MMM-YYYY(기본값), MMM d,yyyy, All). 세미콜론으로 구분하여 여러 날짜 형식 값을 지정할 수 있습니다.

예

```
epmautomate importTMAAttributeValues TMGA TMGA.csv METHOD=Replace DATEFORMAT="All"
```

importTmPremappedTransactions

특정 데이터 소스의 경우 Account Reconciliation 저장소의 파일에서 사전 매핑된 트랜잭션 데이터를 트랜잭션 일치로 임포트합니다.

서비스 관리자는 [uploadFile](#) 명령을 사용하여 트랜잭션 파일을 서비스에 업로드합니다.

이 명령은 임포트 상태 및 임포트 로그 파일 이름을 콘솔에 표시합니다. 서비스 관리자는 [downloadFile](#) 명령을 사용하여 로그 파일을 로컬 컴퓨터에 다운로드합니다.

임포트 파일 형식 요구사항 및 데이터 임포트에 대한 자세한 내용은 *Account Reconciliation*을 통한 계정 조정에서 데이터 임포트를 참조하십시오.

① 주

- 한 번에 하나의 일치 유형에 대해서만 트랜잭션을 임포트할 수 있습니다. 하지만 병렬 임포트는 일치 유형이 서로 다른 경우에도 실행될 수 있습니다.
- [작업] 화면에서와 달리, 한 번에 하나의 파일에서만 사전 매핑된 트랜잭션 데이터를 임포트할 수 있습니다.
- 모든 데이터 소스에 대해 사전 매핑된 트랜잭션을 임포트한 후 runautomatch 명령을 실행합니다.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

epmautomate importTmPremappedTransactions *MATCH_TYPE DATA_SOURCE FILE_NAME*
[*DATE_FORMAT*] 설명:

- *MATCH_TYPE*은 Account Reconciliation에서 정의된 일치 유형입니다.
- *DATA_SOURCE*는 지정한 조정 유형과 연계된 데이터 소스의 식별자입니다.
- *FILE_NAME*은 임포트할 트랜잭션이 포함된 CSV 파일의 이름입니다. 서비스에서 이 파일을 사용할 수 있어야 합니다.
- *DATE_FORMAT*은 트랜잭션 임포트 파일에 포함된 일자 필드의 형식을 나타내는 선택적 매개변수입니다. 기본값은 dd-*MMM*-*YYYY*입니다. 지원되는 다른 날짜 형식은 MM/dd/*yyyy*, dd/MM/*yyyy*, MM-dd-*yyyy*, d-M-*yyyy* 및 *MMM* d.*yyyy*입니다.

예

```
epmautomate importTmPremappedTransactions "INTERCOMPANY" "AP" dailydata.csv d-M-YYYY
```

importValidIntersections

ZIP 파일에서 비즈니스 프로세스로 적합한 교차 그룹을 임포트합니다. 이 명령을 실행하기 전에 [uploadFile](#) 명령을 사용하여 ZIP 파일을 환경에 업로드합니다.

ZIP 파일에는 두 개의 워크시트(규칙 및 하위 규칙)가 있는 하나의 Excel 파일(적합한 교차 정의 포함)이 포함되어야 합니다. 규칙 워크시트에서는 교차 그룹, 포함된 차원, 등록정보(예: Unspecified Valid, Additional Dims Required)를 정의해야 합니다. 하위 규칙 워크시트에서는 멤버 선택항목 및 제외항목을 제공해야 합니다. 자세한 내용은 *Planning 관리*에서 다음 항목을 참조하십시오.

- 앵커 및 비앵커 차원
- 적합한 교차점 예

임포트 파일 형식 템플릿을 가져오는 가장 좋은 방법은 애플리케이션에서 적합한 교차점을 익스포트하는 것입니다. 다음 그림에 샘플 형식이 있습니다.

	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
1										
2	Region - Product	1		true	Entity	true	Product	false		
3	Region-Product-VI-Copy	2		true	Entity	true	Product	false		
4										
5										
6										

	A	B	C	D	E	F	G
1	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							

적용 대상

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

필수 역할

서비스 관리자

사용법

```
epmautomate importValidIntersections FILE_NAME.zip
[ErrorFile=ERROR_FILE_NAME.txt] 설명:
```

- FILE_NAME은 적합한 교차점 정의의 Excel 파일이 있는 ZIP 파일의 이름입니다.
- ErrorFile은 선택사항으로, 오류 레코드가 기록될 텍스트 파일의 이름을 나타냅니다. 이 매개변수 값이 지정되지 않으면 명령이 자동으로 오류 파일을 생성합니다. 이 파일의 이름은 작업 콘솔에서 볼 수 있습니다.

예

```
epmautomate importValidIntersections VI_Import_File.zip
ErrorFile=VI_Import_Log.txt
```

invalidLoginReport

환경에 대해 지정된 감사 보존 기간에 해당하는 지정된 기간 동안 환경에 로그인하는 데 실패한 시도가 나열된 부적합한 로그인 보고서를 생성합니다. 기본 보존 기간은 30일입니다. Oracle Cloud Identity Console에서 **감사 보존 기간(일)** 설정을 변경하여 최대 90일로 연장할 수 있습니다. 감사 데이터를 90일이 넘는 기간 동안 보존하려면 이 보고서 및 [역할 지정 감사 보고서](#)를 정기적으로 다운로드 및 아카이브합니다.

부적합한 로그인 보고서에는 다음 정보가 포함됩니다.

- 로그인을 시도한 사용자의 사용자 이름
- 사용자가 로그인하려고 시도한 원격 IP 주소
- 로그인 시도 타임스탬프

이 보고서에는 해당 Identity Cloud Service에 대한 실패한 로그인 시도가 모두 표시됩니다. 일부는 하나의 Oracle Fusion Cloud Enterprise Performance Management 인스턴스와 관련이 없을 수도 있습니다.

	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

[downloadFile](#) 명령을 사용하여 보고서를 로컬 컴퓨터로 다운로드할 수 있습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

ID 도메인 관리자 및 사전 정의된 역할

사용법

`epmAutomate invalidLoginReport FROM_DATE TO_DATE FILE_NAME.CSV` 설명:

- `FROM_DATE`는 보고서를 생성할 기간의 시작 날짜(YYYY-MM-DD 형식)를 나타냅니다.
- `TO_DATE`는 보고서를 생성할 기간의 종료 날짜(YYYY-MM-DD 형식)를 나타냅니다.
- `FILE_NAME`은 보고서 CSV 파일의 이름입니다.

Note

Cloud EPM에서는 보고서 생성 중에 적합한 날짜 범위만 사용되도록 합니다. 이러한 검증은 시작 및 종료 날짜에 대해 수행됩니다.

- `FROM_DATE`는 현재 날짜부터 허용된 최대 보존 기간보다 이전일 수 없습니다. 이 날짜는 Oracle Cloud Identity Console의 감사 데이터 보존 기간 설정 값 내에 있어야 합니다.
- `TO_DATE`는 `FROM_DATE`부터 최대 보존 기간보다 이후일 수 없습니다.
- `TO_DATE`는 `FROM_DATE`보다 이전일 수 없습니다.

예

```
epmAutomate invalidLoginReport 2021-06-01 2021-06-30 invalidLoginReport.CSV
```

listBackups

환경의 사용가능한 백업 스냅샷을 나열하여 특정 백업을 사용할 수 있는지 판별하므로, 이를 통해 직접 백업을 아카이브하거나 백업을 사용하여 현재 환경을 복원할 수 있습니다.

특정 백업을 복원하기 전에 이 명령을 사용하여 Oracle Object Storage에서 필요한 백업을 사용할 수 있는지 확인하십시오. 백업을 사용할 수 있는 경우 [restoreBackup](#) 명령을 실행하여 복원(사용자 환경에 복사)할 수 있습니다. 백업을 복사한 후 [importSnapshot](#) 명령을 사용하여 임포트할 수 있습니다. 환경 셀프 서비스 복원을 사용하면 처리 시간이 절약됩니다.

Narrative Reporting 이외의 서비스에서는 이 명령이 이름 지정 규칙 YYYY-MM-DDTHH:MM:SS/Artifact_Snapshot.zip(예: 2022-02-16T21:00:02/Artifact_Snapshot.zip)을 사용하여 사용가능한 백업 스냅샷(일별 유지관리 프로세스에서 생성됨)을 나열합니다. Narrative Reporting에서는 사용가능한 스냅샷이 이름 지정 규칙 YYYY-MM-DDTHH:MM:SS/EPRCS_Backup.tar.gz(예: 2022-02-16T21:00:02/EPRCS_Backup.tar.gz)를 사용합니다. 두 경우 모두 타임스탬프는 스냅샷이 생성된 UTC 시간을 반영합니다. 다음 그림은 샘플 명령 출력을 보여 줍니다.

```
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
```

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

```
epmAutomate listBackups
```

예

```
epmAutomate listBackups
```

listFiles

기본 위치, Data Management 폴더 및 profitinbox/profitoutbox(Profitability and Cost Management)에 있는 파일의 이름을 나열합니다.

또한, 이 명령은 증분 및 백업 익스포트 파일과 마이그레이션 스냅샷, 액세스 로그, 활동 보고서를 나열합니다. 이 그림은 명령 출력의 잘린 버전을 보여 줍니다.

```

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

```

환경의 스냅샷이 생성되는 동안(예: 일별 유지관리 중) 이 명령을 실행하면 현재 스냅샷이 나열되지 않습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

```
epmautomate listFiles
```

예

```
epmautomate listFiles
```

loadData

profitinbox에 제공된 파일을 사용하여 데이터를 계산 큐브로 로드합니다.

서비스 관리자는 [uploadFile](#) 명령을 사용하여 profitinbox에 파일을 로드합니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자

- 고급 사용자

사용법

epmautomate loadData APPLICATION_NAME dataFileName=File_Name PARAMETER=VALUE 설명:

- APPLICATION_NAME은 데이터를 로드할 Profitability and Cost Management 애플리케이션의 이름입니다.
- dataFileName=File_Name은 profitinbox에서 사용할 수 있는 데이터 로드 파일을 지정합니다. 데이터 파일 이름은 큰따옴표로 묶어야 합니다.
- PARAMETER=VALUE는 데이터를 로드할 런타임 매개변수와 해당 값을 나타냅니다. 프로세스에 필요한 개수만큼 매개변수와 값 쌍을 지정합니다. 적합한 매개변수 및 해당 값은 다음과 같습니다.
 - clearAllDataFlag=true|false는 애플리케이션 큐브에서 기존 데이터를 지울지 여부를 지정합니다.
 - dataLoadValue=OVERWRITE_EXISTING_VALUES|ADD_TO_EXISTING은 기존 데이터를 처리하는 방법을 지정합니다.

예

```
epmautomate loadData BksML12 dataFileName="data1.txt"clearAllDataFlag=true
dataLoadValue="OVERWRITE_EXISTING_VALUES"
```

loadDimensionViewpoint

정의된 로드를 사용하여 로드 파일의 데이터를 바인딩되지 않거나, 바인딩되거나, 부분적으로 바인딩된 뷰포인트(노드의 서브세트)로 로드합니다.

로드 파일(CSV, Excel(XLSX) 파일 또는 1개의 CSV 또는 XLSX 파일이 포함된 ZIP 파일)은 뷰포인트를 로드하는 환경에서 사용할 수 있어야 합니다. [uploadFile](#) 또는 [copyFileFromInstance](#) 명령을 사용하여 로드 파일을 환경으로 업로드할 수 있습니다.

기본 로드 정의를 사용하여 뷰포인트를 로드하려면 [loadViewpoint](#) 명령을 사용합니다.

적용 대상

Oracle Fusion Cloud Enterprise Data Management

필수 역할

서비스 관리자

사용법

epmautomate loadDimensionViewpoint APPLICATION_NAME DIMENSION_NAME FILE_NAME LOAD_NAME [loadOption=ReplaceNodes|Merge] [purpose="PURPOSE"] 설명:

- APPLICATION_NAME은 Cloud EDM 애플리케이션의 이름입니다.
- DIMENSION_NAME은 로드할 차원의 이름입니다.
- FILE_NAME은 뷰포인트를 로드할 파일의 이름(CSV, XLSX 또는 ZIP 확장자 포함)입니다.
- LOAD_NAME은 뷰포인트 로드에서 사용할 정의된 로드의 이름입니다.
- loadOption은 선택사항으로, 뷰포인트를 로드하는 방법을 나타냅니다. 적합한 로드 옵션은 다음과 같습니다.

- ReplaceNodes - 로드 파일의 관계를 제외한 모든 관계(단독 관계 및 동일한 계층 세트를 사용하는 다른 뷰포인트에서 사용하는 관계 포함)를 계층에서 지웁니다. 기본 로드 유형입니다.
- Merge - 증분 변경사항만 처리하여 기존 관계를 유지합니다.
- purpose는 선택사항으로, 큰따옴표로 묶인 텍스트 문자열이며, 뷰포인트를 로드하는 이유를 나타냅니다.

예

- 기존 관계를 바꿔 뷰포인트 로드: `epmautomate loadDimensionViewPoint "Data Warehouse" Citizen Data_Warehouse_Citizen_20241108.csv DW-Citizen`
- 증분 변경사항을 지정된 로드와 병합하여 뷰포인트 로드: `epmautomate loadDimensionViewPoint "Data Warehouse" Citizen Data_Warehouse_Citizen_20241108.csv DW-Citizen loadOption=Merge purpose="Load Test"`

loadDimData

profitinbox에 있는 파일 하나 이상의 차원 메타데이터를 애플리케이션에 로드합니다.

서비스 관리자는 [uploadFile](#) 명령을 사용하여 profitinbox에 메타데이터 파일을 로드합니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

```
epmautomate loadDimData APPLICATION_NAME dataFileName=File_Name
[stringDelimiter="DELIMITER"] [acceptableDecreasePercentage=PERCENTAGE]:
```

- `APPLICATION_NAME`은 차원 메타데이터를 로드할 Profitability and Cost Management 애플리케이션의 이름입니다.
- `dataFileName`은 profitinbox에서 사용할 수 있는 차원 메타데이터 로드 파일을 지정합니다. 여러 파일에서 메타데이터를 로드하려면 파일 이름을 구분자로 구분하여 나열합니다.
- `stringDelimiter`는 선택사항으로, 메타데이터 파일 이름을 구분하는 데 사용되는 구분자를 지정합니다. 구분자는 큰따옴표로 묶어야 합니다.
- `acceptableDecreasePercentage`는 선택사항으로, 작업에 허용되는 멤버 수 차이 퍼센트(% 기호 없음)를 지정합니다. 들어오는 파일의 새 멤버 수가 기존 멤버 수보다 적은 경우, 이 값은 허용되는 감소 퍼센트를 나타냅니다. 멤버 수의 편차가 이 퍼센트를 초과하면 차원 데이터 로드에 실패합니다.

예

```
epmautomate loadDimData BksML12 dataFileName="dimdata1.txt#dimdata1.txt"
stringDelimiter="#" acceptableDecreasePercentage=5
```

loadViewpoint

로드 파일에서 뷰포인트(노드의 서브세트)를 Oracle Fusion Cloud Enterprise Data Management 애플리케이션으로 로드합니다.

뷰포인트 로드를 사용하면 바인딩되지 않거나, 바인딩되거나, 부분적으로 바인딩된 뷰포인트에 데이터를 로드할 수 있습니다. 뷰포인트 로드 파일(CSV, Excel(XLSX) 파일 또는 1개의 CSV 또는 XLSX 파일이 포함된 ZIP 파일)은 뷰포인트를 로드하는 환경에서 사용할 수 있어야 합니다. [uploadFile](#) 또는 [copyFileFromInstance](#) 명령을 사용하여 로드 파일을 환경으로 업로드할 수 있습니다.

적용 대상

Cloud EDM

필수 역할

서비스 관리자

사용법

```
epmautomate loadViewpoint VIEW VIEWPOINT PURPOSE FILE_NAME
[loadType=ReplaceNodes|Merge]
, 여기서:
```

- VIEW는 Cloud EDM 뷰의 이름입니다.
- VIEWPOINT는 로드할 뷰포인트의 이름입니다.
- PURPOSE는 큰따옴표로 묶인 텍스트 문자열로, 뷰포인트가 로드되는 이유를 나타냅니다.
- FILE_NAME은 뷰포인트를 로드할 파일의 이름(확장자 포함)입니다.
- loadType은 선택적으로 뷰포인트를 로드하는 방법을 나타냅니다. 유효한 값은 Merge 및 ReplaceNodes입니다.
 - 증분 변경을 처리하여 기존 관계를 유지하려면 Merge를 사용합니다.
 - 로드 파일의 관계를 제외한 모든 관계(고아 관계 및 동일한 계층 집합을 사용하는 다른 뷰포인트에서 사용하는 관계 포함)를 계층에서 지우려면 ReplaceNodes를 사용합니다. 기본 로드 유형입니다.

예

- **증분 변경 병합:** `epmautomate loadViewpoint USOperations Entity "Daily Upstream Load" data_Entity.CSV loadType=Merge`
- **기존 계층 바꾸기:** `epmautomate loadViewpoint USOperations Entity "Replace US Operations data" data_Entity.CSV`

login

환경에 대한 보안 연결을 설정합니다. 이 명령은 일반 텍스트 비밀번호를 사용하거나 비밀번호 또는 OAuth 2.0 새로고침 토큰이 포함된 암호화된 비밀번호 파일을 사용하여 환경에 로그인하는 것을 지원합니다.

로그인하여 세션을 시작합니다. 이 세션은 로그아웃할 때까지 활성 상태로 남아 있습니다.

① 주

- 이 명령은 다중 요소 인증(MFA)을 사용하여 기본 인증을 설정한 사용자에게는 지원되지 않습니다.
- EPM Automate는 조직의 SSO 인증서를 사용하는 로그인을 지원하지 않습니다.
- EPM Automate는 SOCKS 프록시에서는 작동하지 않습니다. HTTP/HTTPS 프록시에서만 작동합니다.
- बै치 파일에서 이 명령을 사용하여 작업을 자동화하는 경우 बै치 파일에 일반 텍스트 비밀번호가 기록되지 않도록 암호화된 비밀번호 또는 OAuth 2.0 새로고침 토큰을 사용하는 것이 좋습니다.
- Windows 컴퓨터에서는 이 명령이 연결이 설정되지 않도록 할 수 있는 누락된 프록시 서버 중간 보안 인증서를 자동으로 확인하고 C:\Oracle\EPM Automate 아래에 설치된 JRE에 추가합니다. 이렇게 하면 프록시 서버를 사용하여 인터넷에 액세스하는 경우 발생하는 보안 인증서 관련 로그인 오류가 방지됩니다.
Linux 컴퓨터에서는 login 명령이 프록시 서버에서 누락된 보안 인증서를 확인하고, 다운로드하고, 오류를 표시합니다. 그러면 root 액세스 권한이 있는 사용자가 환경 변수에 확인된 JAVA_HOME에서 사용할 수 있는 JRE에 다운로드된 인증서를 설치할 수 있습니다. 다음 정보 소스를 참조하십시오.
 - [Java Runtime Environment 및 EPM Automate](#)
 - Keytool Java 설명서

이전 버전을 사용하는 경우 로그인하면 EPM Automate를 업그레이드하라는 메시지가 표시됩니다. [upgrade](#) 명령을 사용하여 자동으로 설치를 업그레이드할 수 있습니다.

[addUsers](#), [removeUsers](#), [assignRole](#) 또는 [unassignRole](#) 명령을 실행하려면 OAuth 새로고침 토큰을 사용하여 로그인하지 마십시오. 이러한 명령에는 기본 인증을 사용해야 합니다. 모든 다른 명령은 OAuth 2.0에서 작동합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사용법

- **암호화되지 않은 비밀번호 사용:** `epmautomate login USERNAME PASSWORD URL [IDENTITYDOMAIN] [ProxyServerUserName=PROXY_USERNAME ProxyServerPassword=PROXY_PASSWORD ProxyServerDomain=PROXY_DOMAIN] [KeystorePassword=PASSWORD]`

- **암호화된 파일 사용:** `epmautomate login USERNAME PASSWORD_FILE URL [IDENTITYDOMAIN] [ProxyServerUserName=PROXY_USERNAME] [ProxyServerPassword=PROXY_PASSWORD] [ProxyServerDomain=PROXY_DOMAIN] [KeystorePassword=KEYSTORE_PASSWORD]`

명령 설명:

- `USERNAME`은 사용자의 사용자 이름입니다.

① 주

이 매개변수는 필수 매개변수이지만 해당 값은 기본 인증에만 사용됩니다. OAuth2 기반 인증의 경우 이 값은 OAuth2 토큰에서 파생됩니다. 이 명령을 실행하는 동안 제공된 값은 무시됩니다.

- `PASSWORD`는 사용자의 비밀번호입니다.
- `PASSWORD_FILE`은 사용자의 암호화된 비밀번호 또는 OAuth 2.0 새로고침 토큰을 저장하는 파일의 이름과 위치입니다. [encrypt](#) 명령을 참조하십시오.
- `URL`은 연결할 환경의 기본 URL입니다. Oracle Fusion Cloud Enterprise Performance Management URL 대신 사용자정의 URL 또는 Vanity URL을 사용할 수 있습니다. [관리자용 시작 가이드](#)의 Vanity URL 사용을 참조하십시오.

① 주

API 게이트웨이 또는 역방향 프록시를 사용하는 경우 Cloud EPM URL 대신 해당 URL 및 환경에 대해 정의된 컨텍스트를 사용하십시오.

- `IDENTITYDOMAIN`은 선택사항으로, 환경의 ID 도메인입니다. 이 값은 Cloud EPM URL에서 자동으로 파생됩니다. 지정한 값은 모두 무시됩니다.
- `ProxyServerUserName`은 인터넷에 대한 액세스를 제어하는 HTTP 프록시 서버와의 보안 세션을 인증할 사용자 이름입니다. 도메인 이름 접두어 없이 사용자 이름을 지정하십시오. 네트워크에 대해 프록시 서버 인증을 사용으로 설정한 경우에만 필요합니다.
- `ProxyServerPassword`는 프록시 서버에 사용자를 인증할 비밀번호입니다. 네트워크에 대해 프록시 서버 인증을 사용으로 설정한 경우에만 필요합니다. 이 비밀번호는 암호화할 수 있습니다. [encrypt](#) 명령을 참조하십시오. 이 비밀번호가 암호화된 경우 `PASSWORD_FILE`에서 읽습니다.
- `ProxyServerDomain`은 HTTP 프록시 서버에 대해 정의된 도메인의 이름입니다(서버 이름 또는 프록시 서버 호스트 이름이 아님). 네트워크에 대해 프록시 서버 인증을 사용으로 설정했으며 프록시 서버 도메인이 구성된 경우에만 필요합니다.
- `KeystorePassword`(선택사항)는 프록시 서버 보안 인증서를 임포트하는 데 필요한 키 저장소 비밀번호입니다. 이 매개변수는 Windows에서만 사용하며 프록시 서버를 사용하여 인터넷 액세스 채널을 지정하는 환경에서 다음 오류가 발생하는 경우에만 사용하십시오.

EPMAT-7: 키 저장소에 일부 SSL 인증서가 누락되어 있으므로 연결할 수 없음

EPMAT-7: 키 저장소에 위에 언급된 SSL 인증서가 누락되어 있으므로 연결할 수 없음

① 주

EPM Automate는 컴퓨터의 HTTP/HTTPS 프록시 설정을 감지하고 사용합니다.
EPM Automate는 프록시 서버 연결에 다음 인증 메커니즘을 지원합니다.

- 기본 인증
- 다이제스트 인증
- Kerberos 인증
- Negotiate 프록시 인증
- NTLM 인증

사용가능한 인증 방법 및 해당 구성은 사용 중인 프록시 서버에 따라 다릅니다.

Linux 컴퓨터에서는 프록시 설정에 의해 프록시 서버에 인증해야 하는 경우 이 명령에 대한 매개변수로 프록시 서버 도메인, 사용자 이름, 비밀번호를 입력해야 합니다. 프록시 서버 도메인 이름 및 인증서에 대한 도움이 필요하면 네트워크 관리자에게 문의하십시오.

예

- 암호화되지 않은 Cloud EPM 비밀번호 사용, 프록시 인증 없음:
`epmautomate login serviceAdmin P@ssword1 https://test-cloud-pln.pbcs.us1.oraclecloud.com`
- 암호화된 파일 사용, 프록시 인증 없음:
`epmautomate login serviceAdmin C:\mySecuredir\password.epw https://test-cloud-pln.pbcs.us1.oraclecloud.com`
- 암호화된 파일 사용, 서버 도메인으로 프록시 서버 인증을 사용으로 설정한 경우:
`epmautomate login serviceAdmin C:\mySecuredir\password.epw https://test-cloud-pln.pbcs.us1.oraclecloud.com ProxyServerUserName=john.doe@example.com ProxyServerPassword=example ProxyServerDomain=example`
- 암호화된 파일 사용, 서버 도메인 없이 프록시 서버 인증을 사용으로 설정한 경우:
`epmautomate login serviceAdmin C:\mySecuredir\password.epw https://test-cloud-pln.pbcs.us1.oraclecloud.com ProxyServerUserName=john.doe@example.com ProxyServerPassword=example`
- 서버 도메인을 사용하여 프록시 서버에서의 인증이 사용으로 설정된 경우 암호화된 Cloud EPM 및 프록시 서버 비밀번호 사용:
`epmautomate login serviceAdmin C:\mySecuredir\password.epw https://test-cloud-pln.pbcs.us1.oraclecloud.com ProxyServerUserName=john.doe@example.com ProxyServerDomain=example`
- 서버 도메인 없이 프록시 서버에서의 인증이 사용으로 설정된 경우 암호화된 Cloud EPM 및 프록시 서버 비밀번호 사용:
`epmautomate login serviceAdmin C:\mySecuredir\password.epw https://test-cloud-pln.pbcs.us1.oraclecloud.com ProxyServerUserName=john.doe@example.com`
- OAuth2 새로고침 토큰과 함께 암호화된 파일 사용:
`epmautomate login ignoredUsername C:\mySecuredir\OAuth2.epw https://test-cloud-pln.epm.us-ashburn-1.ocs.oraclecloud.com`

① 주

이 명령에서 EPM Automate는 지정된 사용자 이름 *ignoredUsername*을 무시합니다. 대신 환경에 로그인하는 데 사용되는 사용자 이름은 OAuth2 토큰에서 파생됩니다.

- APIGEE API 게이트웨이를 통해 암호화된 파일 사용:

```
epmautomate login serviceAdmin C:\mySecuredir\password.epw https://
exampleapigee.apigee.com/epm example_ID_DOM
```
- Vanity URL 사용:

```
epmautomate login serviceAdmin C:\mySecuredir\password.epw https://rebrand.ly/
Automate
```

logout

환경에 대한 현재 연결을 종료합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사용법

```
epmautomate logout
```

예

```
epmautomate logout
```

maskData

애플리케이션 데이터를 마스킹하여 데이터 개인 정보를 보호합니다. 애플리케이션 개발자로부터 중요한 데이터를 숨기려면 테스트 환경에서만 이 명령을 사용합니다.

경고: 이 명령은 현재 애플리케이션 데이터를 무작위로 추출하여 의미 없게 만들기 때문에 프로덕션 환경에서는 사용하지 마십시오. 이 명령의 결과는 실행취소할 수 없습니다. 서비스 환경의 데이터를 실수로 마스킹한 경우 백업 또는 유지관리 스냅샷에서 데이터를 복원해야 합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

epmautomate maskData [-f], 설명: -f는 사용자 확인 없이 마스킹 프로세스를 강제로 시작하는 옵션입니다. -f 옵션을 사용하지 않으면 명령에서 작업을 확인하라는 메시지를 표시합니다.

예

```
epmautomate maskData [-f]
```

mergeDataSlices

집계 저장영역 큐브의 모든 증분 데이터 슬라이스를 기본 데이터베이스 슬라이스로 병합하고, 선택적으로 값이 0인 셀을 제거합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

epmautomate mergeDataSlices CUBE_NAME [keepZeroCells=true|false] 설명:

- CUBE_NAME은 모든 데이터 슬라이스를 병합할 집계 저장영역 큐브를 확인합니다.
- keepZeroCells는 값이 0인 셀(영역에서 데이터를 논리적으로 지우면 값이 0인 셀이 생성됨)을 제거할지 여부를 지정합니다(선택사항). 기본값은 true입니다.

예

```
epmautomate mergeDataSlices repl keepZeroCells=false
```

mergeSlices

증분 데이터 슬라이스를 기본 데이터베이스 큐브에 병합하고, 선택적으로 0이 값으로 포함된 Oracle Essbase 셀을 제거하여 큐브가 압축되도록 합니다.

0이 포함된 셀을 제거하면 큐브 성능이 최적화됩니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

`epmautomate mergeSlices applicationName [removeZeroCells=true|false]` 설명:

- `applicationName`은 Profitability and Cost Management 애플리케이션의 이름입니다.
- `removeZeroCells`는 선택적으로 0이 포함된 셀을 제거할 것인지 지정합니다. 이 매개변수의 기본값은 `false`입니다.

예

- 0이 포함된 셀을 제거하지 않고 슬라이스 병합:
 - `epmautomate mergeSlices BksML30`
 - `epmautomate mergeSlices BksML30 removeZeroCells=false`
- 슬라이스 병합 및 0이 포함된 셀 제거: `epmautomate mergeSlices BksML30 removeZeroCells=true`

optimizeASOCube

ASO 큐브의 데이터 추출에 필요한 집계 뷰를 선택하는 쿼리의 성능을 최적화합니다.

이 명령을 사용하면 데이터 크기가 커서 기본 집계가 데이터 추출 또는 보고 요구를 충족하는 데 부족하다고 간주되는 경우 ASO 큐브에 대한 쿼리 최적화 작업을 수행할 수 있습니다. 일반적인 최적화 프로세스는 다음과 같습니다.

- 기본 집계 및 쿼리 기반 집계를 삭제합니다.
- 쿼리 추적을 시작합니다.
- Profitability and Cost Management Query Manager, Oracle Smart View for Office 또는 Data Management의 샘플 쿼리와 Oracle Essbase를 교육하는 데 최적화가 필요한 쿼리 유형을 나타내는 다른 MDX 쿼리를 실행합니다.
- 최적화된 쿼리 또는 기본 쿼리를 기반으로 집계를 생성합니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

`epmautomate optimizeASOCube APPLICATION_NAME OPTIMIZATION_TYPE` 설명:

- `APPLICATION_NAME`은 ASO 큐브가 속한 Profitability and Cost Management 애플리케이션의 이름입니다.
- `OPTIMIZATION_TYPE`은 큐브 최적화 작업입니다. 허용되는 값은 다음과 같습니다.
 - `clearAggregations` - 기본 뷰 및 쿼리 기반 뷰를 제거합니다.
 - `createAggregations` - 기본 Essbase 집계 뷰를 생성합니다. 이 옵션은 쿼리 기반 집계가 아닌 기본 집계를 수행하는 데 사용됩니다.
 - `startQueryTracking` - 쿼리 추적을 시작합니다.

- stopQueryTracking - 쿼리 추적을 중지합니다. 이 옵션은 Essbase의 최적화 정보 수집을 중지하는 데 사용됩니다. Essbase는 사용자가 쿼리 추적을 중지하거나 Essbase를 중지할 때까지 계속 최적화 정보를 수집합니다. Essbase는 쿼리 추적이 중지될 때까지 수집된 데이터를 기반으로 뷰를 집계할 수 있습니다.
- createQBOAggregations - 쿼리 추적을 사용으로 설정한 후 실행한 최적화된 쿼리를 기반으로 Essbase 집계 뷰를 생성합니다.

예

- 기본 집계 뷰 및 쿼리 기반 집계 뷰 삭제:
epmautomate optimizeASOCube BksML12 clearAggregations
- 쿼리 추적 시작
epmautomate optimizeASOCube BksML12 startQueryTracking
- 쿼리 추적을 시작한 후 실행한 최적화된 쿼리를 기반으로 Essbase 집계 뷰 생성:
epmautomate optimizeASOCube BksML12 createQBOAggregations

programDocumentationReport

Profitability and Cost Management 애플리케이션 논리가 포함된 프로그램 설명서 보고서를 생성합니다.

[downloadFile](#) 명령을 사용하여 보고서를 로컬 컴퓨터로 다운로드할 수 있습니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사용법

```
epmautomate programDocumentationReport APPLICATION_NAME POV_NAME
[fileName=FILE_NAME] [fileType=PDF|WORD|EXCEL|HTML] [useAlias=true|false]
[skipFilters=true|false] stringDelimiter="DELIMITER" 설명:
```

- APPLICATION_NAME은 프로그램 설명서 보고서를 생성할 Profitability and Cost Management 애플리케이션의 이름입니다.
- POV_NAME은 보고서를 생성할 애플리케이션의 모델 POV 이름입니다.
- fileName은 보고서 파일의 고유 이름(확장자 포함)입니다(선택사항). 기본 보고서 파일 이름은 HPCMMMLProgramDocumentationReport_APPLICATION_NAME_POV_NAME.pdf입니다.
- fileType은 출력 파일 형식입니다(선택사항). 기본값은 PDF입니다.
- useAlias는 멤버 이름 대신 별칭을 인쇄할지 여부를 지정합니다(선택사항). 기본값은 false입니다.
- skipFilters는 선택사항으로, 필터를 사용하는 규칙이 많은 큰 모델의 보고서 생성 속도가 높아지도록 필터를 무시할지 지정합니다. 기본값은 false입니다.
이 매개변수를 true로 설정하면 각 규칙 필터를 해결하는 프로세스를 건너뛰어 보고서의 예상 소스 수, 예상 대상 수, 예상 타겟 수 값이 결정됩니다. 대신, 프로세스에서는 해당 레벨 0 멤버

수를 사용합니다. 이 매개변수 값이 `false`로 설정되거나 지정되지 않으면 명령에서 모든 필터를 해결하여 더 정확한 수가 생성됩니다.

- `stringDelimiter`는 POV 값에서 사용되는 구분자입니다. 구분자는 큰따옴표로 묶어야 합니다.

예

- 규칙 필터 해결 후 보고서 생성:
`epmautomate programDocumentationReport BksML30 2024_Feb_Actual fileName=Feb-Actual.xls fileType=Excel useAlias=true stringDelimiter="_"`
- 필터를 무시하고 보고서 생성:
`epmautomate programDocumentationReport BksML30 2024_Feb_Actual fileName=Feb-Actual.xls fileType=Excel useAlias=true skipFilters=true stringDelimiter="_"`

provisionReport

역할 지정 보고서(.CSV 파일)를 생성하여 기본 다운로드 위치에 저장합니다.

보고서에는 사용자에게 지정된 사전 정의된 역할(예: 서비스-이름 고급 사용자)과 애플리케이션 역할(예: Planning 애플리케이션 역할인 대량 할당)이 나열됩니다. [downloadFile](#) 명령을 사용하여 보고서를 다운로드할 수 있습니다.

간소화된 버전 또는 클래식 버전 두 가지로 보고서를 생성할 수 있습니다. 액세스 제어 화면에서 사용할 수 있는 역할 지정 보고서와 동일한 간소화된 보고서에는 사전 정의된 역할에 포함되는 애플리케이션 역할 또는 사용자에게 지정된 애플리케이션 역할의 구성요소 역할은 나열되지 않습니다. 클래식 버전의 보고서에는 사용자에게 지정된 사전 정의된 역할에 포함되는 구성요소 역할이 나열됩니다. 직접 또는 그룹을 통해 사용자에게 지정된 애플리케이션 역할도 나열됩니다.

이 보고서를 생성하면 액세스 제어에서 사용할 수 있는 사용자 및 역할 정보가 새로 고쳐집니다.

Oracle Fusion Cloud Enterprise Performance Management는 사용자가 비활성화되어 있는 경우 사전 정의된 역할이 있더라도 비활성화된 사용자를 사전 정의된 역할이 지정되지 않은 사용자와 동일하게 간주합니다. 비활성화된 사용자에 대한 정보는 이 보고서에 포함되지 않습니다.

① 주

이 명령은 향후 릴리스에서는 더 이상 사용되지 않습니다. 이 명령 대신 [roleAssignmentReport](#) 명령을 사용하여 같은 보고서를 생성합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할
- 사전 정의된 역할 및 액세스 제어 - 애플리케이션 역할 보기

사용법

`epmautomate provisionReport REPORT_NAME [format=classic|simplified]`
`[userType=serviceUsers|IDAdmins]` 설명:

- `REPORT_NAME`은 보고서의 이름입니다.
- `format`은 선택적으로 보고서의 서식을 지정하는 방법을 확인합니다. 허용되는 값:
 - `simplified`는 기본 옵션으로, 액세스 제어 화면에서 생성된 역할 지정 보고서와 동일한 보고서를 생성합니다.
 - `classic`은 사용자에게 지정된 사전 정의된 역할에 포함되는 구성요소 역할이 나열된 보고서를 생성합니다. 직접 또는 그룹을 통해 사용자에게 지정된 애플리케이션 역할도 나열합니다.
- `userType`은 선택적으로 보고서에 포함될 사용자를 확인합니다. 이 매개변수의 값을 지정하지 않으면 기본값인 `serviceUsers`가 사용됩니다. 허용되는 값:
 - `serviceUsers`는 모든 기능 사용자에 대한 정보를 포함하는 보고서를 생성합니다 (애플리케이션에 대한 액세스 권한을 부여하는 사전 정의된 역할에 지정되지 않은 경우 ID 도메인 관리자는 포함되지 않음).
 - `IDAdmins`는 ID 도메인 관리자 역할에 지정된 사용자만 나열하는 보고서를 생성합니다. 이 보고서는 클래식 및 간소화 형식에서 동일합니다.

예

- 클래식 보고서 생성: `epmautomate provisionReport myProvReport.CSV format=classic`
- 간소화 보고서 생성:
 - `epmautomate provisionReport myProvReport.CSV format=simplified`
 - `epmautomate provisionReport myProvReport.CSV userType=serviceUsers`
- ID 도메인 관리자만 나열하는 보고서 생성:
 - `epmautomate provisionReport myProvReport.CSV userType=IDAdmins`
 - `epmautomate provisionReport myProvReport.CSV userType=IDAdmins format=classic`

purgeArchivedTmTransactions

Account Reconciliation 애플리케이션에서 아카이브된 일치 트랜잭션을 제거합니다.

정기적으로 [archiveTmTransactions](#) 명령을 사용하여 오래된 일치 트랜잭션을 아카이브한 후 이 명령을 실행하여 Account Reconciliation에서 해당 트랜잭션을 제거하면 애플리케이션 크기가 최적의 상태로 유지됩니다.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

`epmautomate purgeArchivedTMTransactions JobID=JOB_ID`, 여기서 JobID는 일치 트랜잭션을 아카이브하기 위해 실행된 TM 트랜잭션 아카이브 작업의 ID입니다. [archiveTmTransactions](#) 명령을 실행하면 EPM Automate 콘솔에 이 작업 ID가 표시됩니다. 작업 콘솔에서 찾을 수도 있습니다.

예

```
epmautomate purgeArchivedTMTransactions JobID=100000002655003
```

purgeTmTransactions

Account Reconciliation에서 일치 트랜잭션을 제거합니다.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

```
epmautomate purgeTmTransactions matchType age [filterOperator=VALUE]
[filterValue=VALUE] [logFilename=FILE_NAME] 설명:
```

- `matchType`은 일치 트랜잭션을 삭제할 일치 유형의 ID(TextID)입니다.
- `age`는 트랜잭션이 일치된 이후의 일수를 나타냅니다. 이 값의 일수보다 오래되거나 일수가 같은 일치 트랜잭션은 삭제됩니다.
- `filterOperator`는 선택사항으로, 삭제할 일치 트랜잭션이 포함된 계정을 확인하는 다음 필터 조건 중 하나입니다. 이 값과 `filterValue` 값을 조합하여 일치 트랜잭션을 제거할 계정을 확인합니다.
 - 같음
 - `not_equals`
 - `starts_with`
 - `ends_with`
 - `contains`
 - `not_contains`
- `filterValue`는 선택사항으로, 제거할 트랜잭션을 확인하는 필터 값입니다. `filterOperator`가 `equals` 또는 `not_equals`인 경우 공백으로 구분된 목록을 사용하여 여러 값을 지정할 수 있습니다(예: `filterValue=101-120 filterValue=102-202`). 여러 값이 지정되면 필터 연산자 및 필터 값 조합과 일치하는 계정의 트랜잭션이 제거용으로 선택됩니다.

- logFilename은 선택사항으로, 명령 활동에 대한 정보를 기록할 로그 파일의 이름입니다. 파일 이름을 지정하지 않으면 이름이 PurgeTransactions_JOB_ID인 로그 파일이 자동으로 생성됩니다.

① Note

filterOperator 및 filterValue가 지정되지 않으면 age의 일수보다 오래되거나 일수가 같은 모든 일치 트랜잭션이 matchType이 지정된 모든 계정에서 제거됩니다.

예

- 일수가 180일 이상이 되었고 일치 유형이 cashrecon인 일치 트랜잭션 제거:
epmautomate purgeTMTransactions cashrecon 180 logFileName=tmlogs.log
- 계정 101-120 또는 102-202에 대해 일수가 180일 이상이 되었고 일치 유형이 cashrecon인 일치 트랜잭션 제거:
epmautomate purgeTMTransactions cashrecon 180 filterOperator>equals
filterValue=101-120 FilterValue=102-202
- 문자열 11이 포함된 계정에 대해 일수가 180일 이상이 되었고 일치 유형이 cashrecon인 일치 트랜잭션 제거:
epmautomate purgeTMTransactions cashrecon 180 filterOperator=contains
filterValue=11

recomputeOwnershipData

소유권 데이터를 재계산합니다

다음 상황에서는 Financial Consolidation and Close에서 소유권 데이터를 재계산해야 합니다.

- 소유권 관리 계정에 대한 대체 규칙을 추가하거나 삭제한 후
- 연결 방법 범위 설정을 변경한 후
- 엔티티 구조 변경 여부에 관계없이 데이터베이스 새로고침 후

엔티티 구조가 변경되지 않은 경우에도 각 데이터베이스 새로고침 후에 Tax Reporting에서 소유권 데이터를 재계산해야 합니다.

적용 대상

Financial Consolidation and Close 및 Tax Reporting.

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자

사용법

epmautomate recomputeOwnershipData Scenario Year Period 설명:

- Scenario는 재계산할 시나리오의 이름입니다.
- Year는 재계산할 연도입니다.
- Period는 재계산할 연도의 첫번째 기간입니다.

선택한 기간 및 모든 후속 기간이 재계산됩니다.

① 주

재계산이 필요한 POV는 소유권 데이터가 재계산된 후에만 연결할 수 있습니다.

예

```
epmautomate recomputeOwnershipData FCCS_total_Actual FY19 Jan
```

recreate

재생성하여 환경을 정리된 상태로 복원합니다.

환경을 재생성하여 다음 태스크를 완료합니다.

- 전체 스냅샷을 임포트하기 전에 환경을 정리합니다.
- 환경에 배포할 수 있는 비즈니스 프로세스를 변경합니다.

⚠ 주의

- 이 명령은 환경에서 기존 애플리케이션 및 선택적으로 모든 사용자 정의 아티팩트를 삭제합니다. 또한 데이터베이스를 재생성하고 기존 데이터를 모두 제거합니다. 서비스를 재생성한 후 새 비즈니스 프로세스를 생성하거나 마이그레이션 또는 EPM Automate를 사용하여 비즈니스 프로세스를 임포트할 수 있습니다.
- 이 명령은 마이그레이션 내역을 삭제합니다. 따라서 마이그레이션에서 사용가능한 마이그레이션 상태 보고서에는 내역 정보가 포함되지 않습니다.
- 이 명령을 사용하기 전에 환경의 전체 백업을 수행합니다. [runDailyMaintenance](#) 명령을 실행하여 백업 스냅샷을 생성할 수 있습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자

사용법

```
epmautomate recreate [-f] [removeAll=true|false] [TempServiceType=Service_type]
```

설명:

- -f는 재생성 프로세스가 사용자 확인 없이 강제로 시작하도록 합니다. -f 옵션을 사용하지 않으면 명령에서 작업을 확인하라는 메시지를 표시합니다.
- removeAll은 모든 스냅샷과 인박스(업로드된 파일) 및 아웃박스(환경에서 익스포트된 파일) 콘텐츠를 제거합니다(선택사항). 기본값은 false로, 스냅샷과 인박스 및 아웃박스 콘텐츠가 유지됩니다.

- TempServiceType은 선택사항으로, 환경을 다른 서비스 환경으로 변환합니다. 환경에 배포할 수 있는 비즈니스 프로세스는 사용자가 보유한 구독 유형에 따라 달라집니다. 예를 들어 EPM Standard Cloud Service 구독이 있는 경우 환경을 Account Reconciliation에서 Planning으로 변환한 후에는 무형식 애플리케이션을 생성할 수 없습니다. EPM Enterprise Cloud Service 구독이 있는 경우 서비스 유형을 적절하게 변경한 후 환경에 비즈니스 프로세스를 생성할 수 있습니다. *관리자용 시작 가이드*의 새 Cloud EPM 서비스 정보를 참조하십시오.

이 매개변수의 동작은 구독에 따라 다릅니다.

- **EPM Standard Cloud Service 및 EPM Enterprise Cloud Service 이외의 구독:**
TempServiceType 옵션을 사용하여 Planning, Enterprise Planning, Tax Reporting 또는 Financial Consolidation and Close 환경을 Account Reconciliation, Cloud EDM 또는 Profitability and Cost Management 환경으로 일시적으로 변환할 수 있습니다. 예를 들어 Planning 환경을 구매한 경우 다음 명령을 실행하여 Account Reconciliation 환경으로 변환할 수 있습니다.

```
epmautomate recreate -f removeAll=true TempServiceType=ARCS
```

환경을 Account Reconciliation으로 변환한 후 적절한 TempServiceType 값을 사용하여 Cloud EDM 또는 Profitability and Cost Management 환경으로 변환할 수 있습니다. 예를 들어 Profitability and Cost Management 환경으로 변환하려면 다음 명령을 실행합니다.

```
epmautomate recreate -f removeAll=true TempServiceType=PCMCS
```

환경을 다시 원래 서비스 유형으로 변환하려면 다음 명령을 실행합니다.

```
epmautomate recreate -f
```

Profitability and Cost Management: 다음 명령을 실행하여 Profitability and Cost Management 환경을 Planning, Enterprise Planning 또는 Enterprise Profitability and Cost Management 환경으로 변환할 수 있습니다.

```
epmautomate recreate -f removeAll=true TempServiceType=PBCS
```

환경을 다시 원래 Profitability and Cost Management 환경으로 변환하려면 다음 명령을 사용합니다.

```
epmautomate recreate -f TempServiceType=PCMCS
```

① 주

Profitability and Cost Management 환경은 Account Reconciliation, Cloud EDM 또는 Narrative Reporting 환경으로 변환될 수 없습니다.

- **EPM Standard Cloud Service 및 EPM Enterprise Cloud Service 구독:**
TempServiceType 옵션을 사용하여 Oracle Fusion Cloud Enterprise Performance Management 환경을 지원되는 다른 환경으로 변환할 수 있습니다.

EPM Enterprise Cloud Service 구독은 일반적인 Cloud EPM 플랫폼을 사용합니다. 처음에는 지원되는 Cloud EPM 비즈니스 프로세스를 배포할 수 있습니다.

배포된 비즈니스 프로세스에서 다른 비즈니스 프로세스로 전환하려면 환경에 대한 새 서비스 유형을 지정하여 환경을 재생성합니다. 예를 들어, Account Reconciliation 비즈니스 프로세스를 생성했으나 이제 Cloud EDM 환경을 생성하려는 경우 다음과 같이 재생성 명령을 실행합니다.

```
epmautomate recreate -f removeAll=true TempServiceType=EDMCS
```

한 비즈니스 프로세스(예: Account Reconciliation)를 Planning, Tax Reporting 또는 Financial Consolidation and Close로 변환하려면 TempServiceType 값을 지정하지 마십시오. 예를 들어 Account Reconciliation 비즈니스 프로세스를 생성했으나 이제 Planning 모듈 환경을 생성하려는 경우 다음과 같이 재생성 명령을 실행합니다.

```
epmautomate recreate -f removeAll=true
```

허용되는 TempServiceType 값:

- ARCS - 환경을 Account Reconciliation 환경으로 변환합니다.
- EDMCS - 환경을 Cloud EDM 환경으로 변환합니다.
- EPRCS - 환경을 Narrative Reporting 환경으로 변환합니다.
- PCMCS - 환경을 Profitability and Cost Management 환경으로 변환합니다.

예

- 사용자 생성 스냅샷과 인박스 및 아웃박스 콘텐츠를 제거하지 않고 현재 환경을 재생성하여, 원래 서비스 유형으로 복원(전에 TempServiceType 매개변수를 사용하여 재생성이 실행된 경우):

```
epmautomate recreate -f
```

- 현재 환경을 재생성하고, 원래 서비스 유형으로 복원하고(전에 TempServiceType 매개변수를 사용하여 재생성이 실행된 경우), 스냅샷과 인박스 및 아웃박스 콘텐츠 제거:

```
epmautomate recreate -f removeAll=true
```

- 현재 환경을 Enterprise Data Management 환경으로 재생성하고 인박스 및 아웃박스 콘텐츠와 기존 스냅샷 제거:

```
epmautomate recreate -f removeAll=true TempServiceType=EDMCS
```

- 현재 EPM Enterprise Cloud Service Account Reconciliation 환경을 Financial Consolidation and Close 환경으로 재생성하고 인박스 및 아웃박스 콘텐츠와 기존 스냅샷을 제거합니다.

```
epmautomate recreate -f removeAll=true
```

refreshCube

애플리케이션 큐브를 새로 고칩니다. 일반적으로 애플리케이션에 메타데이터를 가져온 다음 큐브를 새로 고칩니다.

큐브 새로고침 작업을 완료하는 데 필요한 시간은 애플리케이션 구조의 변경사항과 큐브에 미치는 영향에 따라 다릅니다. 예를 들어 최소 블록 저장영역 큐브 멤버를 업데이트한 후 새로 고치는 경우 시간이 오래 걸리지 않을 수 있지만, 밀집 블록 저장영역 큐브 멤버 또는 집계 저장영역 큐브 멤버를 업데이트한 후 큐브를 새로 고치는 경우 시간이 오래 걸릴 수 있습니다. 다음 유지관리 기간에 애플리케이션이 백업되기 전에 큐브 새로고침 작업이 완료되었는지 확인해야 합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

`epmautomate refreshCube [JOB_NAME]`, 설명: `JOB_NAME`은 선택사항으로, 애플리케이션에서 정의된 데이터베이스 새로고침 작업의 이름입니다.
명령이 실행되는 콘솔에 작업 상태가 에코됩니다. 애플리케이션의 **작업** 화면에 있는 **최근 활동** 페이지에서도 상태를 볼 수 있습니다.

예

```
epmautomate refreshCube DaliyCubeRefresh
```

refreshReportPackageDatasource

Narrative Reporting이 데이터 소스와 연계된 모든 Doclet을 체계적으로 업데이트하도록 보고서 패키지의 데이터 소스를 새로고침합니다.

이 명령을 완료하는 데 걸린 시간은 영향을 받는 Doclet 수와 직접 관련이 있습니다.

Narrative Reporting 데이터 소스에는 Planning, FreeForm, Financial Consolidation and Close, Enterprise Profitability and Cost Management, Tax Reporting, Profitability and Cost Management, Oracle ERP Cloud 및 Oracle Essbase가 포함됩니다.

적용 대상

Narrative Reporting

필수 역할

서비스 관리자

사용법

```
epmautomate refreshReportPackageDatasource REPORT_PACKAGE_NAME  
[refreshableSources=DATA_SOURCE_NAME]. 설명:
```

- `REPORT_PACKAGE_NAME`은 새로고침할 데이터 소스를 사용하는 보고서 패키지의 위치(경로) 및 이름입니다.
- `refreshableSources`는 선택사항으로, 보고서 패키지에 사용되는 심표로 구분된 데이터 소스 목록입니다. 이 값을 지정하지 않으면 이 명령은 보고서 패키지에서 사용하는 모든 데이터 소스를 새로고침합니다.

예

```
epmautomate refreshReportPackageDatasource "Samples/Report Packages/Sample Report  
Package - MS Word" refreshableSources="Sample Model, Sample Word Model"
```

removeUserFromGroups

ANSI 또는 UTF-8 인코딩된 CSV 파일에서 확인된 액세스 제어 그룹에서 사용자의 멤버십을 제거합니다.

파일 형식은 다음과 같습니다.

```
Group Name
Group1
Group2
```

① 주

이 그룹이 액세스 제어에 있어야 합니다. 그룹 이름 값은 대소문자를 구분하지 않습니다.

서비스 관리자는 [uploadFile](#) 명령을 사용하여 파일을 환경에 업로드합니다.

완료되면 명령이 실패한 각 항목에 대한 정보를 콘솔에 인쇄합니다. 이 정보를 검토하여 CSV 파일의 일부 항목에서 명령 실행이 실패한 이유를 알아봅니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할

사용법

`epmautomate removeUserFromGroups FILE_NAME User_Login` 설명:

- `FILE_NAME`은 사용자의 멤버십을 제거할 액세스 제어 그룹 이름이 포함된 CSV 파일의 이름입니다.
- `User_Login`은 액세스 제어 그룹에서 멤버십을 제거할 Oracle Fusion Cloud Enterprise Performance Management 사용자의 로그인 ID입니다. 이 사용자 로그인 ID는 환경을 서비스하는 ID 도메인에 있어야 하며 사전 정의된 역할에 지정되어야 합니다. 이 값은 대소문자를 구분하지 않습니다.

예

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

removeUsers

ID 도메인에서 환경에 업로드된 ANSI 또는 UTF-8 인코딩의 CSV 파일에서 확인된 계정을 삭제합니다.

파일 형식은 다음과 같습니다.

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

서비스 관리자는 [uploadFile](#) 명령을 사용하여 파일을 환경에 업로드합니다. 사용자 로그인 값은 대소문자를 구분하지 않습니다. 예를 들어 jane.doe@example.com은 Jane.Doe@Example.com 또는 대소문자의 모든 변형과 동일한 것으로 처리됩니다.

① 주

- CSV 파일에는 이 명령을 실행하는 사용자의 계정이 포함되면 안됩니다.
- 사용자 계정은 ID 도메인 관리자가 지원하는 모든 서비스 환경에 공통적으로 적용되므로 한 환경에 대해 계정을 삭제하면 ID 도메인 관리자를 공유하는 모든 환경에 대해 삭제됩니다.

완료되면 명령이 실패한 각 항목에 대한 정보를 콘솔에 인쇄합니다. 이 정보를 검토하여 CSV 파일의 일부 항목에서 명령 실행이 실패한 이유를 알아봅니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

ID 도메인 관리자 및 사전 정의된 역할

사용법

`epmautomate removeUsers FILE_NAME` 설명: *FILE_NAME*은 ID 도메인에서 제거할 사용자의 로그인 ID가 포함된 CSV 파일의 이름입니다.

예

```
epmautomate removeUsers remove_users.CSV
```

removeUsersFromGroup

ANSI 또는 UTF-8 인코딩 CSV 파일에 나열된 사용자를 액세스 제어에서 유지관리되는 그룹에서 제거합니다.

파일 형식은 다음과 같습니다.

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

사용자 로그인 값은 대소문자를 구분하지 않습니다. 예를 들어 jane.doe@example.com은 Jane.Doe@Example.com 또는 대소문자의 모든 변형과 동일한 것으로 처리됩니다. 서비스 관리자는 [uploadFile](#) 명령을 사용하여 사용자 로그인이 포함된 파일을 환경에 업로드합니다.

완료되면 명령이 실패한 각 항목에 대한 정보를 콘솔에 인쇄합니다. 이 정보를 검토하여 CSV 파일의 일부 항목에서 명령 실행이 실패한 이유를 알아봅니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할

사용법

`epmautomate removeUsersFromGroup FILE_NAME GROUP_NAME` 설명:

- `FILE_NAME`은 액세스 제어에서 유지관리되는 그룹에서 제거할 사용자의 로그인 이름을 포함하는 CSV 파일의 이름입니다.
- `GROUP_NAME`은 사용자를 제거할 액세스 제어 그룹의 이름입니다. 이 값은 대소문자를 구분하지 않습니다.

① 주

다음 두 조건이 모두 충족되는 경우에만 사용자가 그룹에서 제거됩니다.

- 파일에 포함된 사용자 로그인이 환경을 서비스하는 ID 도메인에 있습니다.
- 사용자가 ID 도메인에서 사전 정의된 역할에 지정되었습니다.

예

```
epmautomate removeUsersFromGroup user_file.CSV example_group
```

removeUsersFromTeam

CSV 파일에 나열된 Oracle Fusion Cloud Enterprise Performance Management 사용자를 팀에서 제거합니다.

CSV 파일에 포함된 사용자가 팀 멤버가 아닌 경우 이 명령은 해당 사용자를 무시합니다. 이 파일의 값은 대소문자를 구분하지 않습니다. CSV 파일 형식은 다음과 같습니다.

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

서비스 관리자는 [uploadFile](#)을 사용하여 .CSV 파일을 환경에 업로드합니다.

적용 대상

Financial Consolidation and Close, Tax Reporting, Account Reconciliation

필수 역할

- 서비스 관리자

- 사전 정의된 역할 및 팀 - 애플리케이션 역할 관리
- 사전 정의된 역할 및 사용자 - 애플리케이션 역할 관리

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

`epmautomate removeUsersFromTeam FILE.CSV TEAM_NAME` 설명:

- `FILE`은 팀에서 제거할 사용자의 로그인 ID가 나열된 UTF-8 형식의 CSV 파일을 확인합니다.
- `TEAM_NAME`은 액세스 제어에 정의된 팀 이름을 확인합니다. 이 값은 대소문자를 구분하지 않습니다.

예

```
epmautomate removeUsersFromTeam example_users.csv example_team
```

renameSnapshot

환경에서 생성했거나 업로드한 스냅샷의 이름을 바꿉니다.

생성 또는 아카이브 중인 스냅샷의 이름을 바꾸기 위해 이 명령을 실행하는 경우 다음 오류 중 하나가 표시됩니다.

- 스냅샷이 생성되는 경우 `File not found` 오류 또는 `Invalid File` 오류
- 스냅샷이 아카이브되고 있는 경우 `Archive process is in progress. Unable to Rename or Delete` 오류

환경에서 유지관리 스냅샷의 이름은 바꾸지 마십시오. 유지관리 스냅샷의 백업을 유지관리하려면 환경의 아티팩트 스냅샷을 로컬 컴퓨터로 다운로드한 다음, 필요에 따라 이름을 바꾸어야 합니다. [관리자용 시작 가이드](#)에서 유지관리 스냅샷 개요를 참조하십시오.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

`epmautomate renameSnapshot SNAPSHOT_NAME NEW_SNAPSHOT_NAME`, 설명:

- `SNAPSHOT_NAME`은 기존 스냅샷의 이름입니다. 이 값은 공백, \ (백슬래시), / (슬래시), * (별표), ? (물음표), " (따옴표), < (보다 작음), > (보다 큼) 등의 특수 문자를 포함하면 안 됩니다.
- `NEW_SNAPSHOT_NAME`은 스냅샷에 지정할 고유 이름입니다.

예

```
epmautomate renameSnapshot "Example Snapshot" Example_Snapshot_18_09_25
```

replay

서비스가 지정된 로드 상태일 때 사용자 환경이 허용되는지 확인하기 위해 환경에서 Oracle Smart View for Office, REST API 또는 EPM Automate 로드를 재생하여 과도한 로드 상태에서 성능 테스트를 사용합니다.

서비스에서 실행해야 하는 활동을 확인하는 재생 파일을 생성해야 합니다. 재생 파일을 생성하는 방법에 대한 자세한 내용은 [Replay 명령 실행 준비](#)를 참조하십시오.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사용법

```
epmautomate replay REPLAY_FILE_NAME.csv [duration=N] [trace=true] [lagTime=t]
[encrypt=true|false] 설명:
```

- `REPLAY_FILE_NAME`은 환경에서 실행할 활동을 저장하는 CSV 파일입니다.
- `Duration`은 환경에서 활동이 실행되는 시간(분)을 나타냅니다(선택사항). 이 값을 설정하지 않으면 HAR 파일의 활동이 한 번 실행됩니다. 이 매개변수로 지정된 시간 내에 HAR 파일의 활동이 완료되지 않으면 활동이 완료될 때까지 명령이 HAR 파일을 재실행합니다. 예를 들어 실행하는 데 3분이 걸리는 HAR 파일을 재생하기 위해 `duration=10`을 설정한다고 가정합니다. 이 시나리오에서 재생 명령은 네 번째 반복이 완료될 때까지 HAR 파일 활동을 4번(12분 지속) 실행합니다.
- `trace=true`는 XML 형식으로 추적 파일을 생성하도록 명령에 지시하는 선택적 설정입니다. 이 선택적 설정을 지정하면 명령은 재생 CSV 파일에 포함된 각 HAR 파일마다 하나의 폴더를 생성하고 이 폴더에 관련된 추적 파일을 모두 저장합니다. HAR 파일의 각 작업에 대해 이 명령은 Smart View 응답을 포함하는 추적 파일 하나를 생성합니다. 추적 파일 이름은 `trace-N.xml`로 지정됩니다(예: `trace-1.xml`). 여기서 N은 1부터 시작하는 카운터입니다.

추적 파일을 저장하는 폴더는 EPM Automate를 실행한 디렉토리에 생성됩니다. 명령은 환경의 현재 시스템 시간과 HAR 파일 이름의 조합을 `YYYY_MM_DD_HH_MM_SS_HAR_FILE_NAME` 형식으로 사용하여 폴더 이름을 지정합니다. 예를 들어 HAR 파일 이름이 `forecast1.har`이면 폴더 이름은 `2016_06_08_10_21_42_forecast1`일 수 있습니다.

- `[lagTime=t]`는 선택사항으로, 재생 파일에 포함된 각 HAR 파일의 실행 사이에 명령이 대기해야 하는 시간(초)을 지정합니다. 기본값은 5초입니다. 5초 미만의 값을 지정하면 명령이 오류를 표시합니다. 음수(예: -1) 및 분수(예: 1/2)는 매개변수 값으로 허용되지 않습니다. 소수 자릿수 값은 지원됩니다.

첫 번째 HAR 파일의 실행을 시작한 후 명령은 이 매개변수로 지정된 시간(초) 동안 대기했다가 다음 HAR 파일의 처리를 시작합니다. 사용자 활동은 일반적으로 동시에 시작되지 않기 때문에 이 매개변수를 설정하면 환경의 로드 에 대해 보다 현실적인 시뮬레이션을 생성하는 데 도움이 됩니다.

예를 들어 최대 활동 수행 시간 중에 1,000명의 사용자가 환경에 사인온하는 로드를 시뮬레이션한다고 가정해보겠습니다. 해당 세션을 시뮬레이션할 HAR 파일을 생성하고 지연 시간이 6초인 다음 명령을 실행하여 환경에서 처리된 로드를 복제할 수 있습니다.

- `encrypt=true|false`는 재생 파일에 포함된 모든 비밀번호를 암호화할지 여부를 지정합니다 (선택사항). 기본값은 `true`입니다. 무작위 암호화 키를 사용하여 비밀번호를 암호화합니다.

이 명령의 실행과 관련된 자세한 단계는 [샘플 재생 세션](#)을 참조하십시오.

예

```
epmautomate replay forecast1.CSV duration=60 lagTime=5.6
```

resetService

환경을 재시작합니다. 선택적으로 환경을 재시작하기 전에 자동 조정하여 BSO(블록 저장영역 옵션) 큐브용 Oracle Essbase 인덱스 캐시가 애플리케이션에 맞게 최적화되도록 할 수 있습니다.

기본적으로 일별 유지관리가 완료된 후 즉시 환경이 재시작됩니다. 예를 들어 스냅샷을 환경으로 임포트한 후 환경을 자동 조정하는 것이 중요합니다. 이 명령은 심각한 성능 저하가 발견되거나 환경 사용 불가능이 표시된 오류 메시지를 수신하는 경우에만 사용합니다. 환경을 재시작해도 애플리케이션 사용자정의(예: 로케일 변경사항, 테마 및 통화와 관련된 설정 등)에는 영향을 주지 않습니다. 재시작하는 데 최대 15분이 걸립니다.

이 명령을 사용하기 전에 비즈니스 규칙이 환경에서 실행되고 있지 않은지 확인합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자

사용법

```
epmautomate resetService "comment" [AutoTune=true|false] [-f] 설명:
```

- `Comment`는 환경을 재설정하게 만든 이슈에 대한 설명입니다. 설명은 따옴표로 묶어야 합니다.
- `AutoTune`은 환경을 자동 조정하여 Essbase 캐시를 애플리케이션의 BSO 큐브에 맞게 최적화할지 여부를 나타냅니다(선택사항). 기본값은 `false`입니다.

이 매개변수는 Essbase BSO 큐브를 사용하는 환경인 Planning(Planning 모듈 포함), Financial Consolidation and Close, Tax Reporting에서만 사용합니다.

- `-f`는 선택사항으로, 추가 사용자 상호작용 없이 환경을 강제로 재시작하도록 지정합니다. 이 옵션을 사용하지 않으면 명령에서 작업을 확인하라는 메시지를 표시합니다. 이 옵션은 이 명령을 사용하는 스크립트를 스케줄링하는 경우에 유용합니다.

예

- `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

사용가능한 백업 스냅샷을 복사하여 환경으로 임포트하는 데 사용할 수 있도록 합니다.

[listBackups](#) 명령을 사용하여 복원할 백업을 사용할 수 있는지 판별하십시오. 환경에 대해 스냅샷 셀프 서비스 복원을 사용하면 처리 시간이 절약됩니다. 스냅샷을 복원한 후 [importSnapshot](#) 명령을 사용하여 환경으로 임포트하십시오.

① Note

Narrative Reporting만 해당: 다음 프로세스를 사용하여 복원된 스냅샷을 임포트합니다.

1. [downloadFile](#) 명령을 사용하여 복원된 스냅샷을 로컬 컴퓨터로 다운로드합니다.
2. 다운로드한 스냅샷의 이름을 EPRCS_Backup으로 바꿉니다.
3. [uploadFile](#) 명령을 사용하여 환경에 이름을 바꾼 스냅샷(EPRCS_Backup)을 로드합니다. 명령에 to_be_imported 업로드 위치를 지정해야 합니다.
4. Narrative Reporting에서 **일별 유지** 카드를 엽니다.
 - a. **업로드된 백업 스냅샷 사용**을 선택합니다.
 - b. **복원 스케줄링**을 눌러 다음 일별 유지관리 중 스냅샷을 임포트합니다.
 - c. **예**를 누릅니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

`epmAutomate restoreBackup SNAPSHOT_NAME [targetName=TARGET_SNAPSHOT_NAME]`, 설명:

- `SNAPSHOT_NAME`은 [listBackups](#) 명령에서 나열된 대로, 환경에서 사용할 수 있는 백업 스냅샷의 이름입니다.
- `targetName`은 선택사항으로, 타겟 환경에 있는 백업 스냅샷의 이름(확장자 제외)입니다. 이 값을 지정하지 않으면 백업 스냅샷이 `SNAPSHOT_NAME`을 사용하여 타겟 환경에 복원되지만 /(슬래시)가 _(밑줄)로 바뀝니다. 예를 들어 `SNAPSHOT_NAME`이 `2022-05-14T00:08:56/Artifact_Snapshot.zip`인 경우 `targetName`은 `2022-05-14T00:08:56_Artifact_Snapshot.zip`입니다.

예

- Narrative Reporting 이외의 서비스:

```
epmAutomate restoreBackup 2022-05-14T00:08:56/Artifact_Snapshot.zip
targetName=example_Artifact_Snapshot
```

- Narrative Reporting만 해당:
epmAutomate restoreBackup 2022-02-16T21:00:02/EPRCS_Backup
targetName=Example_EPRCS_Backup

restructureCube

블록 저장영역 큐브 전체 재구조화를 수행하여 조각화를 제거하거나 줄입니다. 또한, 재구조화는 비어 있는 블록을 제거하며 애플리케이션에서 큐브로 변경사항을 푸시하지 않습니다.

① 주

이 명령을 실행하기 전에 애플리케이션을 사용 중인 사람이 없도록 하십시오.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Sales Planning 및 Strategic Workforce Planning.

필수 역할

서비스 관리자

사용법

epmAutomate restructureCube *CUBE_NAME* 설명: *CUBE_NAME*은 애플리케이션에 있는 것과 동일한 큐브의 이름입니다.

예

```
epmAutomate restructureCube Plan1
```

roleAssignmentAuditReport

사용자 환경에 지정된 감사 데이터 보존 기간에 해당하는 기간 동안 발생한 사전 정의된 역할 및 애플리케이션 역할 지정 변경사항이 나열된 감사 보고서를 생성합니다. 기본 보존 기간은 30일입니다. Oracle Cloud Identity Console에서 **감사 데이터 보존 기간(일)** 설정을 변경하여 사전 정의된 역할 감사 데이터의 보존을 최대 90일로 연장할 수 있습니다. 애플리케이션 역할 감사 데이터는 최대 120일 동안 사용할 수 있습니다. 감사 데이터를 더 긴 기간 동안 보존하려면 이 보고서 및 [부적합한 로그인 보고서](#)를 정기적으로 다운로드 및 아카이브합니다.

역할 지정 감사 보고서에는 작업 열의 역할이 변경된 사용자 로그인 이름, IDCS 그룹 이름 또는 EPM 그룹 이름이 나열됩니다. 지정되거나 지정되지 않은 사전 정의된 역할 또는 애플리케이션 역할, 역할 변경을 수행한 사용자(수행자 열), 작업이 완료된 24시간 형식의 타임스탬프(UTC)도 포함되어 있습니다. 유형 열은 이름 열이 나타내는 것을 가리킵니다. 사용자(이름 열이 사용자의 로그인 이름을 나타내는 경우), IDCS 그룹(이름 열이 IDCS 그룹 이름을 표시하는 경우) 또는 EPM 그룹(이름 열에 EPM 그룹 이름이 나열된 경우) 세 값 중 하나가 포함될 수 있습니다.

	A	B	C	D	E	F
1	Name	Type	Role	Action	Performed By	Date and Time
2	IDCS-Group1	IDCS Group	Power User	Assign	serviceadmin@example.com	December 11, 2024 00:47:10 UTC
3	jane.doe@example.com	User	Power User	Assign	serviceadmin@example.com	December 11, 2024 00:47:27 UTC
4	epmIDCSGroup	User	Service Administrator	Assign	serviceadmin@example.com	December 11, 2024 00:48:47 UTC
5	john.doe@example.com	User	Power User	Assign	serviceadmin@example.com	December 11, 2024 00:59:06 UTC
6	john.doe@example.com	User	Ad Hoc - Create	Assign	serviceadmin@example.com	December 11, 2024 01:08:37 UTC
7	john.doe@example.com	User	Access Control - Manage	Unassign	serviceadmin@example.com	December 11, 2024 01:10:30 UTC
8	john.doe@example.com	User	Ad Hoc - Create	Unassign	serviceadmin@example.com	January 07, 2025 03:28:56 UTC

이전에 환경의 사전 정의된 역할에 지정되었던 삭제된 사용자에 대한 정보가 이름 열에 사용자의 표시 이름(이름 및 성)으로 나열됩니다. 이 경우 역할 열은 사용자 계정이 삭제되기 전에 보유했던 사전 정의된 역할을 나타냅니다. 이 변경사항은 삭제된 사용자에게 지정된 애플리케이션 역할(있는 경우)에는 적용되지 않습니다. 이러한 지정은 사용자의 사용자 로그인 이름으로 표시됩니다. 예를 들어 다음 그림의 빨간색 상자에 있는 정보를 참조하십시오.

	A	B	C	D	E	F
1	Name	Type	Role	Action	Performed By	Date and Time
2	Jane Doe	User	User	Assign	admin@example.com	January 08, 2025 05:42:47 UTC
3	jane.doe@example.com	User	Run Integrations	Assign	admin@example.com	January 08, 2025 05:44:18 UTC
4	john.smith@example.com	User	Power User	Assign	admin@example.com	January 08, 2025 05:44:18 UTC
5	john.smith@example.com	User	User	Unassign	admin@example.com	January 08, 2025 05:44:18 UTC
6	john.smith@example.com	User	Access Control - View	Assign	admin@example.com	January 08, 2025 05:44:18 UTC
7	epmIDCSGroup	IDCS Group	Viewer	Assign	admin@example.com	January 08, 2025 05:44:22 UTC
8	epmGroup	EPM Group	Ad Hoc - Create	Assign	admin@example.com	January 08, 2025 05:44:22 UTC

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 ID 도메인 관리자 역할
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할
- 사전 정의된 역할 및 액세스 제어 - 애플리케이션 역할 보기

사용법

epmAutomate roleAssignmentAuditReport *FROM_DATE TO_DATE FILE_NAME*.CSV, 설명:

- *FROM_DATE*는 보고서를 생성할 기간의 시작 날짜(YYYY-MM-DD 형식)를 나타냅니다.
- *TO_DATE*는 보고서를 생성할 기간의 종료 날짜(YYYY-MM-DD 형식)를 나타냅니다.
- *FILE_NAME*은 보고서 CSV 파일의 이름입니다. [downloadFile](#) 명령을 사용하여 생성된 보고서를 다운로드할 수 있습니다.

Note

Oracle Fusion Cloud Enterprise Performance Management에서는 보고서 생성 중 적합한 날짜 범위만 사용되도록 합니다. 이러한 검증은 *시작* 및 *종료* 날짜에 대해 수행됩니다.

- *FROM_DATE*는 현재 날짜부터 허용된 최대 보존 기간보다 이전일 수 없습니다. 이 날짜는 Oracle Cloud Identity Console의 **감사 데이터 보존 기간(일)** 설정 값 내에 있어야 합니다.
- *TO_DATE*는 *FROM_DATE* 날짜부터 최대 보존 기간보다 이후일 수 없습니다.
- *TO_DATE*는 *FROM_DATE*보다 이전일 수 없습니다.

예

```
epmAutomate roleAssignmentAuditReport 2024-12-11 2025-01-09 RoleAuditReport.CSV
```

roleAssignmentReport

역할 지정 보고서(.CSV)를 생성합니다.

이 보고서에는 사용자에게 지정된 사전 정의된 역할(예: 서비스 관리자) 및 애플리케이션 역할(예: Planning 애플리케이션 역할인 승인 소유권 지정자, 승인 수퍼바이저, 승인 관리자, 승인 프로세스 디자이너)이 나열됩니다. 필요에 따라 해당 환경의 ID 도메인 관리자를 나열하기 위해 이 보고서를 생성할 수도 있습니다. 이 보고서는 액세스 제어에서 생성된 역할 지정 보고서의 CSV 버전과 일치합니다.

	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

이 보고서를 생성하면 액세스 제어에서 사용할 수 있는 사용자 및 역할 정보가 새로 고쳐집니다.

Oracle Fusion Cloud Enterprise Performance Management는 사용자가 비활성화되어 있는 경우 사전 정의된 역할이 있더라도 비활성화된 사용자를 사전 정의된 역할이 지정되지 않은 사용자와 동일하게 간주합니다. 비활성화된 사용자에 대한 정보는 이 보고서에 포함되지 않습니다.

① 주

이 명령은 [provisionReport](#) 명령을 사용하여 생성한 것과 같은 보고서를 생성합니다.

[downloadFile](#) 명령을 사용하여 보고서를 다운로드할 수 있습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할
- 사전 정의된 역할 및 액세스 제어 - 애플리케이션 역할 보기

사용법

`epmautomate roleAssignmentReport REPORT_NAME.CSV [userType=IDAdmins|serviceUsers]`. 여기서:

- `REPORT_NAME`은 보고서의 이름입니다.
- `userType`은 선택사항으로, 보고서에 포함될 정보를 소유한 사용자 유형을 확인합니다. 기본값은 `serviceUsers`입니다. 적합한 값은 다음과 같습니다.
 - `serviceUsers`는 모든 기능 사용자에 대한 정보를 포함하는 보고서를 생성합니다 (애플리케이션에 대한 액세스 권한을 부여하는 사전 정의된 역할에 지정되지 않은 경우 ID 도메인 관리자는 포함되지 않음).
 - `IDAdmins`는 ID 도메인 관리자 역할에 지정된 사용자만 나열하는 보고서를 생성합니다.

예

- 기능 사용자만 나열하는 보고서를 생성합니다.
 - `epmautomate roleAssignmentReport myReport.CSV`
 - `epmautomate roleAssignmentReport myReport.CSV userType=serviceUsers`
- ID 도메인 관리자만 나열하는 보고서를 생성합니다.


```
epmautomate roleAssignmentReport myReport.CSV userType=IDAdmins
```

runAutoAlert

일치 유형에 대한 자동 알림 작업을 실행합니다. 이 작업은 알림 규칙을 처리한 다음, 필요한 경우 일치하지 않는 트랜잭션에 대한 알림을 자동으로 생성합니다. Account Reconciliation의 작업 내역 탭을 통해 자동 알림 작업의 상태를 모니터링할 수 있습니다.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

`epmAutomate runAutoAlert MATCH_TYPE`. 설명: `MATCH_TYPE`은 Account Reconciliation에 정의된 일치 유형의 식별자입니다.

예

```
epmautomate runAutoAlert INTERCOMPANY
```

runAutomatch

자동 일치 프로세스를 실행하여 서비스 관리자가 정의한 규칙을 통해 트랜잭션을 일치시킵니다.

① 주

[importTmPremappedTransactions](#) 또는 [runDataRule](#) 명령을 사용하여 트랜잭션 데이터를 트랜잭션 일치로 임포트한 후 이 명령을 실행합니다.

Account Reconciliation의 **작업 내역** 탭에서 자동 일치 프로세스의 상태를 모니터링할 수 있습니다.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

`epmautomate runAutomatch RECONCILIATION_TYPE`. 설명: RECONCILIATION_TYPE은 Account Reconciliation에 정의된 일치 유형입니다.

예

```
epmautomate runAutomatch INTERCOMPANY
```

runBatch

Data Management बै치를 실행합니다.

Data Management의 बै치 실행 모드가 [직렬]로 설정되는 경우 बै치의 모든 작업이 완료되면 제어가 반환됩니다. [병렬]로 설정되는 경우 बै치의 모든 작업이 실행을 위해 제출되면 제어가 반환됩니다.

① 주

이 명령을 사용하여 데이터 소스에서 Oracle Fusion Cloud Enterprise Performance Management로의 직접 데이터 로드 통합을 실행할 수 없습니다. 직접 데이터 로드를 통합하려면 EPM Integration Agent를 사용합니다. 자세한 내용은 *Data Integration 관리*에서 [EPM Integration Agent를 사용하여 직접 데이터 로드 수행](#)을 참조하십시오.

적용 대상

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

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

`epmautomate runBatch BATCH_NAME` 설명: `BATCH_NAME`은 Data Management에 정의된 배치의 이름입니다.

예

```
epmautomate runBatch Accounting_batch
```

runBusinessRule

비즈니스 규칙을 실행합니다.

적용 대상

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

필수 역할

- 서비스 관리자
- 고급 사용자(규칙에 대한 실행 액세스 권한이 부여된 경우)

사용법

`epmautomate runBusinessRule RULE_NAME [PARAMETER=VALUE]` 설명:

- `RULE_NAME`은 환경에서 정의된 것과 동일한 비즈니스 규칙의 이름입니다.
- `PARAMETER=VALUE`는 비즈니스 규칙을 실행하는 데 필요한 선택적 런타임 매개변수 및 해당 값을 나타냅니다.

① 주

- 이 명령은 단일 비즈니스 규칙만 실행할 수 있습니다. 규칙 세트를 실행하려면 [runRuleSet](#) 명령을 사용합니다.
- 규칙은 배포된 계획 유형에 대해 실행됩니다.
- 런타임 매개변수의 값을 제공하지 않으면 기본값이 사용됩니다. 이 명령은 규칙에 정의된 것과 정확히 일치하지 않는 런타임 프롬프트를 무시합니다.
- `PARAMETER=VALUE` 쌍을 사용하여 비즈니스 규칙에 필요한 개수만큼 런타임 프롬프트를 지정합니다. 다음 예에서는 두 개의 런타임 프롬프트(Period 및 Entity)와 해당 값(Q1 및 USA)을 사용합니다.
매개변수에 여러 값을 입력하는 방법은 [하나의 매개변수에 여러 값 지정](#)을 참조하십시오.

예

```
epmautomate runBusinessRule RollupUSSales Period=Q1 Entity=USA
```

runCalc

애플리케이션에서 계산을 실행합니다.

이 명령을 사용하면 POV 간에 규칙을 복사하지 않고 다른 데이터 POV의 데이터에 대해 모델 POV의 규칙을 사용하여 계산을 실행할 수 있습니다.

적용 대상

Profitability and Cost Management,

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

```
epmautomate runCalc APPLICATION_NAME POV_NAME [DATA_POV_NAME] PARAMETER=VALUE  
[comment="comment"] stringDelimiter="DELIMITER" 설명:
```

- `APPLICATION_NAME`은 계산할 POV가 포함된 Profitability and Cost Management 애플리케이션의 이름입니다.
- `POV_NAME`은 계산할 모델 POV의 이름입니다.
- `DATA_POV_NAME`은 모델 POV의 규칙을 사용하여 계산할 데이터 POV의 이름입니다(선택사항).
`DATA_POV_NAME`을 지정하지 않으면 기본적으로 `POV_NAME`이 사용됩니다.
`DATA_POV_NAME`을 지정하는 경우 `exeType=ALL_RULES`만 사용할 수 있습니다.
- `PARAMETER=VALUE`는 계산을 실행할 런타임 매개변수와 해당 값을 나타냅니다. 프로세스에 필요한 개수만큼 매개변수와 값 쌍을 지정합니다. 적합한 매개변수 및 해당 값은 다음과 같습니다.
 - `exeType=ALL_RULES|RULESET_SUBSET|SINGLE_RULE`은 규칙 실행 유형을 나타냅니다. 필수 매개변수입니다.
`exeType`에 설정된 값에 따라 다음 매개변수를 지정할 수 있습니다.

- * exeType=ALL_RULES를 지정하는 경우 subsetStart, subsetEnd, ruleSetName, ruleName과 같은 규칙 서브세트 또는 단일 규칙 관련 매개변수를 포함하지 마십시오. DATA_POV_NAME 매개변수를 설정하는 경우 이 exeType을 사용해야 합니다.
 - * exeType=SINGLE_RULE을 지정하는 경우 ruleSetName 및 ruleName 값만 지정합니다.
 - * exeType=RULESET_SUBSET을 지정하는 경우 subsetStart 및 subsetEnd의 값을 지정합니다.
- subsetStart는 실행할 규칙 세트에서 첫 번째 규칙의 순서 번호를 지정합니다.
 - subsetEnd는 실행할 규칙 세트에서 마지막 규칙의 순서 번호를 지정합니다.
 - ruleSetName은 실행할 계산이 포함된 규칙 세트를 식별합니다.
 - ruleName은 실행할 규칙 이름(단일 규칙 실행)입니다.
 - isClearCalculated=true|false는 기존 계산을 지울지 여부를 지정합니다.
 - isExecuteCalculations=true|false는 계산을 실행할지 여부를 지정합니다.
 - isRunNow=true|false 지금 바로 프로세스를 실행하려면 이 값을 true로 설정합니다.
 - optimizeReporting=true|false - 보고 최적화 없이 계산을 실행해야 하는 경우 이 선택적 값을 false로 설정합니다. 기본값은 true입니다.
- 모범 사례:
- * 처리 시간을 절약해야 하는 경우(예: 단일 규칙 또는 여러 POV의 순차 계열을 실행하는 경우)에만 optimizeReporting=false를 설정합니다.
 - * 동시 계산 작업을 여러 개 실행하는 경우 모든 작업에 대해 optimizeReporting=true를 설정합니다. 마지막으로 완료되는 작업만 집계를 수행하여 중복 처리를 피하고 실행 중인 작업이 느려지지 않도록 방지합니다.

① 주

매개변수 값(true 또는 false)은 모두 소문자여야 합니다.

- comment는 큰따옴표로 묶인 선택적 설명입니다.
- stringDelimiter는 POV 값에서 사용되는 구분자입니다. 구분자는 큰따옴표로 묶어야 합니다.

예

```
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

조정 준수에서 정의된 보고서를 생성합니다.

*Account Reconciliation 관리*에서 다음 정보 소스를 참조하십시오.

- 보고서 사용 - 보고서 정의 지침을 제공합니다.
- 조정 준수에서 사전 정의된 보고서 생성 - 사전 정의된 조정 준수 보고서 및 보고서를 생성하는 데 필요한 매개변수 목록을 제공합니다.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

`epmautomate runComplianceReport FILE_NAME GROUP_NAME REPORT_NAME REPORT_FORMAT [Param=value]` 설명:

- `FILE_NAME`은 생성할 보고서의 고유한 파일 이름입니다. 이 이름의 보고서가 서버에 있으면 덮어씁니다. [downloadFile](#) 명령을 사용하여 이 보고서를 로컬 컴퓨터로 다운로드할 수 있습니다.
- `GROUP_NAME`은 보고서가 연계된 그룹의 이름입니다.
- `REPORT_NAME`은 생성할 보고서의 고유 이름입니다.
- `REPORT_FORMAT`은 다음과 같은 보고서 형식 중 하나입니다.
 - PDF
 - HTML(그래프 및 차트에는 지원되지 않음)
 - XLSX(그래프에는 지원되지 않음)
 - CSV
 - CSV2

① 주

`REPORT_FORMAT` CSV는 템플릿 기반의 데이터 형식 지정을 허용하지 않지만 `csv2`는 허용합니다. `csv2`로 형식 지정된 보고서를 생성하는 경우 `csv` 출력에 비해 시간이 더 걸립니다.

- `Param=value`는 선택사항으로, 보고서를 생성하는 데 필요한 매개변수를 확인합니다. 예를 들어 Balance By Account Type 보고서는 값이 July 2017인 `Period`와 값이 Entered인 `Currency Bucket`의 두 매개변수를 사용합니다. 이 매개변수를 "`Period=July 2017`" "`Currency Bucket=Entered`"로 지정해야 합니다.

예

```
epmautomate runComplianceReport "Example_File Name" "Reconciliation Manager"
"Balance By Account Type" PDF "Period=July 2017" "Currency Bucket=Entered"
```

runDailyMaintenance

스케줄링된 일별 유지관리 기간을 기다리지 않고 일별 서비스 유지관리 프로세스를 바로 시작합니다.

이 명령을 사용하여 백업 스냅샷을 강제로 생성하고 환경을 업데이트할 수 있습니다. 이 명령을 실행하기 전에 환경을 사용 중인 사람이 없도록 하십시오. 이 명령은 일별 유지관리 스케줄에 영향을 미치지 않습니다. 환경 변경사항을 적용하기 위해 다음 유지관리 기간까지 기다리지 않으려는 경우 이 명령을 사용합니다(예: 일회용 패치를 적용한 후).

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자, 사용자(Cloud EDM만 해당)

사용법

`epmautomate runDailyMaintenance [skipNext=true|false] [-f]` 설명:

- `skipNext`는 선택적으로 일별 유지관리 프로세스의 다음 발생을 건너뛰지 여부를 나타냅니다. 기본값은 `false`입니다.
- `-f`는 선택적으로 사용자 확인 없이 유지관리 프로세스를 강제로 시작 할지 여부를 나타냅니다. `-f` 옵션을 사용하지 않으면 명령에서 작업을 확인하라는 메시지를 표시합니다.

예

- 다음 스케줄링된 유지관리를 건너뛰지 않고 비주기적인 일별 유지관리를 강제로 시작하려는 경우: `epmautomate runDailyMaintenance -f`
- 비주기적인 일별 유지관리를 강제로 시작하고 다음 스케줄링된 유지관리를 건너뛰려는 경우: `epmautomate runDailyMaintenance skipNext=true -f`
- 비주기적인 일별 유지관리를 시작하고 다음 스케줄링된 유지관리를 건너뛰려는 경우: `epmautomate runDailyMaintenance skipNext=true`

runDataRule

시작 기간 및 종료 기간과 지정한 임포트 또는 익스포트 옵션에 따라 Data Management 데이터 로드 규칙을 실행합니다.

① 주

이 명령을 사용하여 데이터 소스에서 Oracle Fusion Cloud Enterprise Performance Management로의 직접 데이터 로드 통합을 실행할 수 없습니다. 직접 데이터 로드를 통합하려면 EPM Integration Agent를 사용합니다. 자세한 내용은 *Data Integration 관리*에서 [EPM Integration Agent를 사용하여 직접 데이터 로드 수행](#)를 참조하십시오.

적용 대상

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

필수 역할

- 서비스 관리자

- 고급 사용자

사용법

`epmautomate runDataRule RULE_NAME START_PERIOD END_PERIOD IMPORT_MODE EXPORT_MODE [FILE_NAME]` 설명:

- *RULE_NAME*은 Data Management에 정의된 데이터 로드 규칙 이름입니다. 공백이 포함된 규칙 이름은 따옴표로 묶어야 합니다.
- *START_PERIOD*는 데이터를 로드할 첫번째 기간입니다. 이 기간 이름은 Data Management 기간 매핑에 정의되어 있어야 합니다.
- *END_PERIOD*는 복수 기간 데이터 로드에서 데이터를 로드할 마지막 기간입니다. 단일 기간 로드의 경우 시작 기간과 동일한 기간을 입력합니다. 이 기간 이름은 Data Management 기간 매핑에 정의되어 있어야 합니다.
- *IMPORT_MODE*는 데이터를 Data Management로 가져오는 방법을 결정합니다. 임포트 모드 설정은 대소문자를 구분합니다. 허용되는 값은 다음과 같습니다.
 - APPEND - Data Management의 기존 POV 데이터에 추가합니다.
 - REPLACE - POV 데이터를 삭제하고 파일의 데이터로 바꿉니다.
 - RECALCULATE - 데이터를 재계산합니다.
 - NONE - Data Management 스테이지 테이블로의 데이터 임포트를 건너뛵니다.
- *EXPORT_MODE*는 데이터를 애플리케이션으로 익스포트하는 방법을 결정합니다. 익스포트 모드 설정은 대소문자를 구분합니다. 허용되는 값은 다음과 같습니다.
 - STORE_DATA - 데이터 관리 스테이지 테이블의 데이터를 기존 데이터와 병합합니다. 메타데이터를 로드하는 데 사용되는 데이터 관리 작업에서는 항상 이 익스포트 옵션을 사용합니다.
 - ADD_DATA - 데이터 관리 스테이지 테이블의 데이터를 애플리케이션에 추가합니다.
 - SUBTRACT_DATA - 기존 데이터에서 데이터 관리 스테이지 테이블의 데이터를 뺍니다.
 - REPLACE_DATA - POV 데이터를 지우고 Data Management 스테이지 테이블의 데이터로 바꿉니다. 시나리오, 버전, 연도, 기간 및 엔티티의 데이터가 지워집니다.
 - NONE - 데이터 관리에서 애플리케이션으로의 데이터 익스포트를 건너뛵니다.

① 주

Financial Consolidation and Close의 경우 다음 익스포트 모드만 지원됩니다.

- MERGE - Data Management 스테이지 테이블의 데이터를 기존 데이터와 병합합니다.
- REPLACE - DM 스테이징 테이블에서 항목을 제거하고 데이터 로드의 항목으로 바꿉니다.
- NONE - Data Management에서 애플리케이션으로의 데이터 익스포트를 건너웁니다.

Oracle Fusion Cloud를 타겟으로 사용하는 경우 다음 익스포트 모드만 지원됩니다.

- MERGE - Data Management 스테이지 테이블의 데이터를 기존 데이터와 병합합니다.
- NONE - Data Management에서 애플리케이션으로의 데이터 익스포트를 건너웁니다.

- *FILE_NAME*은 선택적 파일 이름입니다. 파일 이름을 지정하지 않을 경우 이 명령은 로드 데이터 규칙에 지정된 파일 이름에 포함된 데이터를 임포트합니다. 이 파일은 인박스 폴더 또는 해당 하위 폴더에 있어야 합니다.

Account Reconciliation에 대한 BAI(Bank Administration Institute) 형식 파일을 로드하는 경우 이 매개변수 값을 지정하지 마십시오. 데이터 규칙 정의에 BAI 파일 로드에는 필요한 파일 이름을 항상 지정해야 합니다.

① 주

데이터 규칙에 경로가 지정된 경우 명령에 파일 경로를 지정하지 마십시오. 파일 이름만 지정하십시오. 데이터 규칙에 경로가 지정되지 않은 경우 데이터 파일의 전체 경로를 지정하십시오.

예

- 복수 기간 임포트:

```
epmautomate runDataRule VisionActual Mar-15 Jun-15 REPLACE STORE_DATA inbox/Vision/GLActual.dat
```
- 단일 기간 임포트:

```
epmautomate runDataRule "Vision Actual" Mar-15 Mar-15 REPLACE STORE_DATA inbox/Vision/GLActual.dat
```

runDMReport

Data Management 보고서를 생성하여 `outbox/reports` 폴더에 저장합니다.

생성된 보고서 이름은 보고서 및 보고서 형식을 생성하는 Data Management 작업의 ID에 따라 지정됩니다. 예를 들어 보고서 작업 ID가 2112이고 지정한 보고서 출력 형식이 PDF이면 보고서 이름은 2112.pdf입니다. 보고서가 생성되면 보고서 이름이 콘솔에 표시됩니다. Data Management의 [프로세스 세부정보] 탭에서 또는 [listFiles](#) 명령을 사용하여 보고서 이름을 확인할 수도 있습니다.

[downloadFile](#) 명령을 사용하여 보고서를 로컬 컴퓨터로 다운로드할 수 있습니다.

적용 대상

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

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

`epmautomate runDMReport REPORT_NAME PARAMETER=Value "Report Output Format=[PDF|HTML|XLS|XLSX]"` 설명:

- `REPORT_NAME`은 보고서 생성에 사용할 Data Management 보고서 템플릿의 이름입니다.
- `PARAMETER=Value`는 보고서 매개변수와 해당 값을 나타냅니다. `PARAMETER=Value` 형식으로 필요한 개수만큼 매개변수를 지정합니다. 필수 매개변수 목록은 생성할 보고서에 따라 다릅니다.

① 주

보고서 런타임 매개변수는 보고서를 디자인할 때 정의됩니다. 이 명령을 실행하려면 워크플로우 탭에서 이러한 매개변수와 값을 생성하고 사용해야 합니다. 보고서의 런타임 매개변수를 생성하려면 Data Management의 [워크플로우] 탭에서 **보고서 실행**을 누른 다음 **보고서 그룹**에서 그룹을 선택합니다. 매개변수를 생성할 보고서를 선택하고 **보고서 스크립트 생성**을 누릅니다. 선택적으로 보고서 매개변수 값을 지정하고 출력 형식을 선택한 다음 **확인**을 누릅니다. **보고서 스크립트 생성**에 표시된 매개변수를 통해 런타임 매개변수와 값을 지정하여 보고서를 생성합니다.

- `Report Output Format`은 보고서 출력 형식을 나타냅니다. 적합한 옵션은 PDF, HTML, XLS, XLSX입니다. 기본 보고서 형식은 PDF입니다.

예

```
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

Data Integration 작업을 실행하여 데이터를 Oracle Fusion Cloud Enterprise Performance Management 비즈니스 프로세스로 импорт하거나 비즈니스 프로세스의 데이터를 외부 시스템으로 익스포트합니다.

이 명령은 [runDataRule](#) 명령을 대체합니다. [runDataRule](#) 명령 대신 이 명령을 사용하는 것이 좋습니다.

적용 대상

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

필수 역할

- 서비스 관리자
- 고급 사용자

사용법

```
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]
```

- 표준 모드 통합의 경우 importMode, exportMode, periodName 값을 지정해야 합니다.
- 빠른 모드 통합의 경우 exportMode 값을 지정해야 합니다.
- 매개변수 이름과 해당 값은 대소문자를 구분합니다.

이 명령에서 각 요소는 다음을 나타냅니다.

- *JOB_NAME*은 Data Integration에서 정의된 통합 작업의 이름입니다.
 - importMode는 데이터를 Data Integration으로 임포트하는 방법을 결정합니다. 사용가능한 임포트 모드는 다음과 같습니다.
 - Append - Data Integration의 기존 POV 데이터에 추가합니다.
 - Replace - POV 데이터를 삭제하고 파일의 데이터로 바꿉니다.
 - Map and Validate - 데이터 임포트를 건너뛰지만 업데이트된 매핑 및 논리 계정을 사용하여 데이터를 재처리합니다.
 - No Import - Data Integration 스테이지 테이블로의 데이터 임포트를 건너뛵니다.
 - exportMode는 데이터를 타겟 애플리케이션에 로드하는 방법을 결정합니다. 빠른 모드 통합의 경우 exportMode 매개변수 값으로 Check 및 No Export를 사용할 수 없습니다. 사용가능한 익스포트 모드 값은 다음과 같습니다.
 - Merge - 기존 데이터를 업데이트하고 새 데이터를 추가합니다.
 - Replace - POV의 기존 데이터를 지우고 새 데이터를 로드합니다. 표준 모드의 경우 시나리오, 버전, 연도, 기간, 엔티티 차원의 데이터가 지워집니다. 빠른 모드의 경우 연도, 기간, 엔티티 차원의 데이터가 지워집니다. 두 모드에서 모두 사용자정의 지우기 영역을 정의할 수 있습니다.
 - Accumulate - 기존 데이터에 데이터를 추가합니다. Planning, Planning 모듈, Financial Consolidation and Close, Tax Reporting, Profitability and Cost Management, Enterprise Profitability and Cost Management에 적용할 수 있습니다.
 - Subtract - 기존 잔액에서 데이터를 뺍니다. Profitability and Cost Management 및 Enterprise Profitability and Cost Management에 적용됩니다.

빠른 모드 통합의 경우:

 - * Check 및 No Export는 이 매개변수의 값으로 사용할 수 없습니다.
 - * Planning, Planning Modules 및 Financial Consolidation and Close의 경우 유일한 적합한 값은 Replace, Merge 및 Accumulate입니다.
 - No Export - 데이터 익스포트를 건너뛵니다. 타겟 애플리케이션에 로드하기 전에 검토를 위해 스테이지 테이블에 데이터를 로드하려면 이 모드를 사용합니다.
 - Check - 데이터 검증 검사만 수행합니다.
- Oracle Fusion Cloud를 타겟으로 사용하는 경우 다음 익스포트 모드만 지원됩니다.
- * MERGE - Data Integration 스테이지 테이블의 데이터를 기존 데이터와 병합합니다.

- * NONE - Data Integration에서 애플리케이션으로의 데이터 익스포트를 건너웁니다.
- `periodName`은 데이터를 임포트 또는 익스포트할 하나 이상 기간 또는 기간 범위의 이름으로, 각각 중괄호로 묶여 있습니다. 사용가능한 기간 이름 지정 규칙은 다음과 같습니다.
 - 단일 기간 로드의 경우 중괄호로 묶어 기간 이름을 지정합니다(예: {Jan-21}).
 - 복수 기간 로드의 경우 시작 및 종료 기간 이름을 중괄호로 묶습니다. 예를 들어 {Jan-21}{Mar-21}은 Jan-21에 시작되어 Mar-21에 종료되는 모든 기간의 데이터를 로드합니다.
 - **Planning, Planning 모듈, Financial Consolidation and Close무형식, Tax Reporting:** 비즈니스 프로세스 기간 이름과 연도를 {Jan#FY21}{Mar#FY21} 형식으로 지정하여 Jan-21에 시작되어 Mar-21에 종료되는 모든 기간의 데이터를 로드할 수 있습니다. 기간 이름은 중괄호로 묶어야 합니다.
- * Single Period - 기간 매핑에서 정의된 단일 기간의 Data Management 기간 이름을 나타냅니다.
- * Multi-Period - 복수 기간 로드를 나타냅니다. 매개변수는 {Month-Year}{Month-Year} 형식으로 지정됩니다. 예를 들어 {Jan-20}{Mar-20}은 Jan-20부터 Mar-20까지의 복수 기간 로드입니다.
- * Planning Period Name - {Month#Year} 형식(예: {Jan#FY20}{Mar#FY20})의 Planning 기간 이름을 나타냅니다. 이 규칙을 사용하면 Data Integration 기간 이름을 지정할 필요가 없습니다. 대신, 연도 및 시나리오 차원의 멤버 이름을 지정합니다. 이 매개변수는 Planning, Tax Reporting, Financial Consolidation and Close 비즈니스 프로세스에서 지원됩니다. 온-프레미스 데이터 소스에서 파생된 서비스 애플리케이션 및 클라우드 배포에 모두 적용됩니다.

연도 및 기간 멤버 이름을 캡처하여 Cloud EPM Groovy 스크립트에서 트리거되는 경우 이 규칙을 사용하면 유용합니다. 기간 매핑의 타겟 값에 연도와 월이 포함된 애플리케이션 기간 매핑 또는 글로벌 기간 매핑이 있어야 합니다.
- * Substitution Variable - 이전 Planning 기간 이름 형식의 확장입니다. 실제 연도 및 월 멤버 이름 대신 대체 변수를 {Month#&CurYr}{&FcstMonth#&CurYr} 형식(예: {Jan#&CurYr}{&FcstMonth#&CurYr})으로 지정할 수 있습니다. 실제 멤버 이름과 대체 변수를 조합할 수 있습니다.

이 형식은 Planning, Tax Reporting, Financial Consolidation and Close 비즈니스 프로세스에서 지원됩니다.

기간 매핑의 타겟 값에 연도 및 월 값이 제공된 애플리케이션 기간 매핑 또는 글로벌 기간 매핑이 명령이 실행된 환경의 Data Integration에 있어야 합니다. 이 경우, 연도와 월은 실행 중 대체 변수의 현재 값을 나타냅니다.
- * GLOBAL POV - 글로벌 POV 기간의 데이터 로드를 실행합니다. {GLOBAL_POV} 형식을 사용합니다.

① Note

이 토론에서 설명한 매개변수 이외의 기간 이름 지정 매개변수를 사용하면 부적합한 입력 - HTTP 400 오류 메시지가 표시됩니다.

- {GLOBAL_POV} - 시스템의 글로벌 POV 또는 Data Integration의 애플리케이션 설정에서 정의된 기간의 데이터 로드를 실행합니다.

Note

{Month#Year} 기간 이름 지정 규칙 형식은 Planning, Planning 모듈, Financial Consolidation and Close, Tax Reporting에서 지원됩니다. 이 규칙에서는 Data Integration 기간 이름 대신 연도 및 시나리오 차원의 멤버 이름을 지정할 수 있습니다. 이 방법은 연도 및 기간 멤버 이름을 캡처하여 Groovy 스크립트에서 명령이 트리거되는 경우에 유용합니다.

{Jan#&CurYr}{&FcstMonth#&CurYr} 대체 변수 이름 지정 규칙은 이전 기간 이름 지정 규칙의 확장입니다. Planning, Planning 모듈, Financial Consolidation and Close, Tax Reporting에서 이 명령을 실행하는 경우 연도 및 월 멤버 이름 대신 대체 변수를 지정할 수 있습니다. 멤버 이름과 대체 변수를 조합할 수도 있습니다.

이전 기간 이름 지정 및 대체 변수 이름 지정 규칙은 타겟 값에 연도와 월이 포함된 애플리케이션 기간 매핑 또는 글로벌 기간 매핑이 Data Integration에 이미 있는 경우에만 작동합니다.

- `inputFileName` - 파일 기반 데이터 로드와 같은 경우 인박스에서 사용가능한, 데이터를 임포트할 파일의 이름을 지정합니다. 통합 정의에서 디렉토리 이름을 지정하는 경우 파일 이름만 전달합니다. 통합 정의에 디렉토리 이름을 포함하지 않는 경우 `inbox/DIR_NAME/FILE_NAME` 형식(예: `inbox/GLBALANCES.txt` 또는 `inbox/EBSGL/GLBALANCES.txt`)을 사용합니다. 파일이 환경의 기본 위치에 업로드된 경우 `#epminbox/FILE_NAME` 규칙(예: `#epminbox/GLBALANCES.txt`)을 사용하여 입력 데이터 파일을 파악합니다. 이 매개변수는 기본 파일 기반 데이터 로드에만 적용할 수 있습니다. 파일 기반 데이터 로드에서 이 매개변수 값을 지정하지 않으면 이 명령은 통합 정의에서 지정된 파일의 데이터를 임포트합니다. 파일 기반이 아닌 데이터 로드에서 이 매개변수 값을 지정하면 명령에서 무시합니다.
- `PARAMETERS`는 선택사항으로, 런타임 매개변수를 `PARAMETER_NAME="PARAMETER"` 형식으로 확인합니다. 매개변수에는 소스 필터와 타겟 옵션이 둘 다 포함됩니다.

Note

현재 타겟 애플리케이션의 차원(메타데이터) 유형에 사용할 수 있는 매개변수는 "Refresh Database"=Yes|No뿐입니다.

예

- 단일 기간 임포트:
`epmAutomate runIntegration VisionDataLoad importMode=Replace exportMode=Merge periodName="{Mar-15}"`
- 복수 기간 임포트:
`epmAutomate runIntegration VisionDataLoad importMode=Replace exportMode=Merge periodName="{Mar-15}{Jun-15}"`
- 증분 파일 기반 데이터 통합:
`epmAutomate runIntegration IncrementalFileLoad importMode=Replace exportMode=Merge periodName="{Jan-20}{Mar-20}" inputFileName=File1.txt`

runIntercompanyMatchingReport

통합 프로세스의 일부로 관련 엔티티 및 파트너 간 거래를 올바르게 매칭하고 분석 및 감사 목적으로 사용하는 데 도움이 되는 내부거래 매칭 보고서를 생성합니다.

적용 대상

재무 통합 및 마감

필수 역할

- 서비스 관리자
- 고급 사용자(ACL을 통해 추가 보안이 부여된 경우)
- 사용자(ACL을 통해 추가 보안이 부여된 경우)

사용법

```
epmautomate runIntercompanyMatchingReport JOB_NAME FILE_NAME [scenario=scenario]
[years=years] [period=period_name] [ReportFormat=HTML]:
```

- JOB_NAME은 애플리케이션에 있는 내부거래 보고서 작업 정의의 이름입니다. 자세한 내용은 *Financial Consolidation and Close 작업*에서 내부거래 매칭 보고서 관리를 참조하십시오. 이 명령은 작업에서 사용가능한 설정을 사용하여 내부거래 매칭 보고서를 생성합니다. 생성된 보고서의 위치로 아웃박스를 선택해야 합니다.

① Note

이 명령을 실행하는 동안 이러한 선택적 값을 입력하여 작업에 지정된 시나리오, 연도, 기간 및 보고서 형식 설정을 대체할 수 있습니다.

- FILE_NAME은 보고서 파일의 이름입니다. 이 보고서는 아웃박스에서 생성됩니다. [downloadFile](#) 명령을 사용하여 보고서를 로컬 컴퓨터에 다운로드하거나 [sendMail](#) 명령을 사용하여 전자메일로 보냅니다.
- SCENARIO는 선택사항으로, 보고서를 생성할 시나리오(애플리케이션에 정의되어 있음)의 이름입니다.
- YEARS는 선택사항으로, 보고서를 생성할 연도를 지정합니다.
- PERIOD는 선택사항으로, 보고서를 생성할 기간입니다.
- ReportFormat은 선택사항으로, 보고서의 형식입니다. 허용되는 값은 HTML, PDF, XLSX입니다.

① Note

- 시나리오, 연도, 기간을 하나씩 조합하여 이 보고서를 생성할 수 있습니다. 시나리오, 연도 또는 기간을 여러 개 지정하지 마십시오.
- 시나리오, 연도, 기간 및 보고서 형식에 대한 값을 지정하지 않으면 보고서 작업 정의에 지정된 해당 값(JOB_NAME으로 확인)이 사용됩니다. 값을 지정하면 보고서 작업 정의에 설정된 값이 이러한 선택적 값으로 대체됩니다.

예

- 작업 정의에 설정된 값을 대체하여 보고서 생성:
epmautomate runIntercompanyMatchingReport IC_Job_01 SampleICReport.html
scenario=Actual years=FY22 period=Jan reportFormat=HTML
- 작업 정의에 설정된 값을 사용하여 보고서 생성:
epmautomate runIntercompanyMatchingReport IC_Job_01 SampleICReport

runMatchingReport

트랜잭션 일치에서 정의된 보고서를 생성합니다.

사전 정의된 트랜잭션 일치 보고서 및 보고서를 생성하는 데 필요한 매개변수 목록은 *Account Reconciliation 관리*에서 트랜잭션 일치에서 사전 정의된 보고서 생성을 참조하십시오.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

epmautomate runMatchingReport *FILE_NAME* *GROUP_NAME* *REPORT_NAME* *REPORT_FORMAT*
[Param=value] 설명:

- *FILE_NAME*은 생성할 보고서의 고유한 파일 이름입니다. 이 이름의 보고서가 서버에 있으면 덮어씁니다. [downloadFile](#) 명령을 사용하여 이 보고서를 로컬 컴퓨터로 다운로드할 수 있습니다.
- *GROUP_NAME*은 보고서가 연계된 그룹의 이름입니다.
- *REPORT_NAME*은 생성할 보고서의 고유 이름입니다.
- *REPORT_FORMAT*은 다음과 같은 보고서 형식 중 하나입니다.
 - PDF
 - HTML(그래프 및 차트에는 지원되지 않음)
 - XLSX(그래프에는 지원되지 않음)
 - CSV
 - CSV2

① 주

`REPORT_FORMAT` CSV는 템플릿 기반의 데이터 형식 지정을 허용하지 않지만 csv2는 허용합니다. CSV2로 형식 지정된 보고서를 생성하는 경우 csv 출력에 비해 시간이 더 걸립니다.

- `Param=Value`는 선택사항으로, 보고서를 생성하는 데 필요한 매개변수를 확인합니다. 예를 들어 `status` 매개변수를 `approved` 값으로 사용하는 Match Type Configuration 보고서의 경우, 매개변수와 값을 `status=Approved`로 지정합니다.

예

```
epmautomate runMatchingReport Example_FileName "Transaction Matching" "Match Type Configuration" HTML "status=Approved"
```

runPipeline

파이프라인에 대해 정의된 변수를 기반으로 Data Integration 파이프라인(일련의 작업을 단일 프로세스로 조정)을 실행합니다. 파이프라인에 대한 자세한 내용은 *Oracle Fusion Cloud Enterprise Performance Management용 Data Integration 관리*의 파이프라인 사용을 참조하십시오.

적용 대상

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

필수 역할

서비스 관리자 또는 파이프라인 정의를 실행하도록 프록시 사용자로 설정된 사용자(위치 보안을 사용하거나 시스템 설정의 관리자가 아닌 사용자를 위한 파이프라인 실행 사용 설정을 예로 지정)

사용법

`epmautomate runPipeline PIPELINE_CODE [PARAMETER=VALUE]` 설명:

- `PIPELINE_CODE`는 Data Integration에서 파이프라인을 생성하는 중 정의된 코드입니다. 자세한 내용은 *Data Integration 관리*의 파이프라인 프로세스 설명을 참조하십시오.
- `PARAMETER=VALUE`는 선택사항으로, 파이프라인을 실행할 매개변수 및 해당 값을 나타냅니다. 프로세스에 필요한 개수만큼 매개변수와 값 쌍을 지정합니다. 매개변수 목록은 Data Integration의 파이프라인 변수 화면에 정의된 변수의 수에 따라 다릅니다. 기본 파이프라인 매개변수 및 허용되는 값은 다음과 같습니다.
 - `STARTPERIOD` - 데이터를 로드할 첫번째 기간입니다. 이 기간 이름은 Data Integration 기간 매핑에 정의되어 있어야 합니다.
 - `ENDPERIOD` - 데이터를 로드할 마지막 기간입니다. 이 기간 이름은 Data Integration 기간 매핑에 정의되어 있어야 합니다.
 - `IMPORTMODE` - 데이터를 Data Integration으로 임포트하는 방법을 결정합니다. 허용되는 값은 다음과 같습니다.
 - * Append - Data Integration의 기존 POV 데이터에 추가합니다.
 - * Replace - POV 데이터를 삭제하고 파이프라인 정의에 지정된 파일의 데이터 또는 매개변수로 바꿉니다.

- * Map and Validate - 파이프라인 정의에 지정된 파일의 데이터 또는 매개변수로 임포트하지 않고 업데이트된 매핑 및 논리 계정으로 데이터를 재처리합니다.
- * No Import - Data Integration 스테이지 테이블로의 데이터 임포트를 건너뛴니다.
- EXPORTMODE - 데이터를 Data Integration으로 익스포트하는 방법을 결정합니다. 허용되는 익스포트 모드는 다음과 같습니다.
 - * Merge - Data Integration 스테이지 테이블의 데이터를 비즈니스 프로세스의 기존 데이터와 병합합니다.
 - * Accumulate - Data Integration 스테이지 테이블의 데이터를 비즈니스 프로세스에 추가합니다.
이 익스포트 모드는 Planning, Planning Modules, FreeForm, Sales Planning, Strategic Workforce Planning에만 적용됩니다.
 - * Replace - 비즈니스 프로세스의 POV 데이터를 삭제하고 Data Integration 스테이지 테이블의 데이터로 바꿉니다. 시나리오, 버전, 연도, 기간 및 엔티티 차원의 데이터가 지워집니다.
 - * No Export - Data Integration에서 비즈니스 프로세스로의 데이터 익스포트를 건너뛴니다.
- ATTACH_LOGS - 로그 파일을 압축하여 파이프라인 실행과 관련된 통지에 첨부할지 여부를 지정합니다. 허용되는 값은 예인 경우 Y이고 아니요인 경우 N입니다.
- SEND_MAIL - 파이프라인 실행 완료 시 전자메일을 전송할지 여부를 지정합니다. 허용되는 값은 항상, 안함(기본값), 실패 시, 성공 시입니다.
- SEND_TO - 전자메일을 전송할 전자메일 ID의 심표로 구분된 목록입니다.

예

```
epmautomate runPipeline DAILYLOAD "STARTPERIOD=Jan-24" "ENDPERIOD=Jan-24"
"IMPORTMODE=Replace" "EXPORTMODE=Merge" "SEND_MAIL=Always"
"SEND_TO=John.Doe@example.com, Jane.Doe@example.com" "ATTACH_LOGS=Y"
```

runPlanTypeMap

plan type map 유형의 작업에 지정된 설정에 따라 블록 저장영역 데이터베이스에서 집계 저장영역 데이터베이스로 데이터를 복사하거나 블록 저장영역에서 다른 블록 저장영역으로 데이터를 복사합니다.

적용 대상

Planning, Planning Modules, FreeForm, Sales Planning 및 Strategic Workforce Planning.

필수 역할

서비스 관리자

사용법

epmautomate runPlanTypeMap JOB_NAME [clearData=true|false] 설명:

- JOB_NAME은 애플리케이션에 정의된 plan type map 유형의 작업 이름입니다.
- clearData는 데이터를 복사하기 전에 대상 데이터베이스의 데이터를 제거해야 하는지를 표시하는 선택적 설정입니다. 이 매개변수 값이 설정되지 않은 경우 기본값인 true가 사용됩니다.

매개변수 값(true 또는 false)은 모두 소문자여야 합니다.

예

```
epmautomate runPlanTypeMap CampaignToReporting clearData=false
```

runRuleSet

비즈니스 규칙 세트를 실행합니다.

적용 대상

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

필수 역할

서비스 관리자, 고급 사용자(규칙 실행 액세스 권한이 부여된 경우)

사용법

epmautomate runRuleSet *RULESET_NAME* [*PARAMETER=VALUE*] 설명:

- *RULESET_NAME*은 환경에서 정의된 것과 동일한 비즈니스 규칙 세트의 이름입니다.
- *PARAMETER=VALUE*는 규칙 세트를 실행하는 데 필요한 선택적 런타임 매개변수 및 해당 값을 나타냅니다.

① 주

규칙 세트는 배포된 계획 유형에 대해 실행됩니다.

PARAMETER=VALUE 쌍을 사용하여 규칙 세트에 필요한 개수만큼 런타임 프롬프트를 지정합니다. 다음 예에서는 두 개의 런타임 프롬프트(Period 및 Entity)와 해당 값(Q1 및 USA)을 사용합니다.

런타임 매개변수의 값을 제공하지 않으면 기본값이 사용됩니다. 이 명령은 규칙 세트에 대해 정의된 것과 정확히 일치하지 않는 런타임 프롬프트는 무시합니다.

매개변수에 여러 값을 입력하는 방법은 [하나의 매개변수에 여러 값 지정](#)을 참조하십시오.

예

```
epmautomate runRuleSet RollupUSSales Period=Q1 Entity=USA
```

runSupplementalDataReport

Supplemental Data Manager의 데이터를 표시하는 관계형 보고서를 생성합니다.

보충 데이터 보고서는 Financial Consolidation and Close 및 Tax Reporting에서 비연결 보고서로 그룹화됩니다. 생성할 수 있는 보고서와 보고서 생성에 사용되는 매개변수 목록은 *Oracle Fusion Cloud EPM용 REST API 가이드*에서 Financial Consolidation and Close 및 Tax Reporting 보고서 생성의 "사전 정의된 보고서 및 매개변수 목록" 섹션을 참조하십시오.

적용 대상

Financial Consolidation and Close, 및 Tax Reporting.

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사용법

`epmautomate runSupplementalDataReport FILE_NAME GROUP_NAME REPORT_NAME REPORT_FORMAT [Param=value]` 설명:

- `FILE_NAME`은 보고서의 고유한 파일 이름입니다.
- `GROUP_NAME`은 보고서가 연계된 그룹의 이름입니다.
- `REPORT_NAME`은 생성할 보고서의 고유 이름입니다.
- `REPORT_FORMAT`은 다음과 같은 보고서 형식 중 하나입니다.
 - PDF
 - HTML(그래프 및 차트에는 지원되지 않음)
 - XLSX(그래프에는 지원되지 않음)
 - CSV
 - CSV2

`REPORT_FORMAT` CSV는 템플릿 기반의 데이터 형식 지정을 허용하지 않지만 CSV2는 허용합니다. CSV2로 형식 지정된 보고서를 생성하는 경우 CSV 출력에 비해 시간이 더 걸립니다.

- `Param=value`는 선택사항으로, 보고서를 생성하는 데 필요한 매개변수를 확인합니다. 예를 들어 값이 `monthly`인 `schedule name`과 값이 `Jan`인 `period`를 사용하여 `At Risk Tasks` 보고서를 생성하려면 `"schedule name"=monthly period=Jan`을 지정합니다.

예

```
epmautomate runSupplementalDataReport Example_File_name Group1 "At Risk Tasks"
html "schedule name"=monthly period=Jan
```

runTaskManagerReport

태스크 관리자의 데이터를 표시하는 관계형 보고서를 생성합니다.

적용 대상

Enterprise Profitability and Cost Management, Financial Consolidation and Close, Tax Reporting.

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사용법

epmautomate runTaskManagerReport *FILE_NAME* *GROUP_NAME* *REPORT_NAME* *REPORT_FORMAT*
[Param=value] 설명:

- *FILE_NAME*은 보고서의 고유한 파일 이름입니다.
- *GROUP_NAME*은 보고서가 연계된 그룹의 이름입니다.
- *REPORT_NAME*은 생성할 보고서의 고유 이름입니다.
- *REPORT_FORMAT*은 다음과 같은 보고서 형식 중 하나입니다.
 - PDF
 - HTML(그래프 및 차트에는 지원되지 않음)
 - XLSX(그래프에는 지원되지 않음)
 - CSV
 - CSV2

① 주

REPORT_FORMAT CSV는 템플릿 기반의 데이터 형식 지정을 허용하지 않지만 csv2는 허용합니다. CSV2로 형식 지정된 보고서를 생성하는 경우 CSV 출력에 비해 시간이 더 걸립니다.

- Param=value는 선택사항으로, 보고서를 생성하는 데 필요한 매개변수를 확인합니다. 예를 들어 값이 monthly인 schedule name과 값이 Jan인 period를 사용하여 Early Tasks 보고서를 생성하려면 "schedule name"=monthly period=Jan을 지정합니다.

예

```
epmautomate runTaskManagerReport Example_File_name Group1 "Early Tasks" PDF
"schedule name"=monthly period=Jan
```

sendMail

Oracle Fusion Cloud Enterprise Performance Management에서 파일을 첨부하는 옵션을 사용하여 전자메일을 전송합니다.

이 명령을 스크립트에 통합하여 사용자에게 다양한 조건을 통지하거나 보고서를 전송할 수 있습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Narrative Reporting, Oracle Fusion Cloud Enterprise Data Management, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자

사용법

epmautomate sendMail ToAddress Subject [Body="MessageBody"]
[Attachments=FILE1,FILE2] 설명:

- ToAddress는 세미콜론으로 구분된 수신자 전자메일 주소를 큰따옴표로 묶어서 나타냅니다. 예를 들어 "jdoe@example.com;jane.doe@example.com"이 있습니다.
- Subject는 전자메일 제목을 나타냅니다.
- Body="MessageBody"는 선택사항으로, 전자메일 콘텐츠입니다. 지정하지 않으면 전자메일에 본문이 없습니다.

① Note

적합한 HTML 태그를 사용하여 메시지 본문의 형식을 지정하고 원하는 전자메일 형식을 생성합니다. 전체 메시지 본문(모든 HTML 태그 포함)은 한 줄로 지정해야 하며 새 줄 문자는 포함되지 않아야 합니다. [예](#)를 참조하십시오.

- Attachments는 선택사항으로, Cloud EPM에서 사용할 수 있으며 전자메일에 첨부할 파일의 심표로 구분된 목록을 나타냅니다. 예를 들어 outbox/errorFile.txt, inbox/users.csv가 있습니다.

① Note

- 파일 이름의 한 문자에 대해 별표(*)를 와일드카드로 사용합니다. 예를 들어 outbox/user*.csv를 지정하여 패턴에 맞는 5글자의 파일 이름으로 아웃박스에 있는 모든 파일을 첨부합니다.
- [listFiles](#) 명령으로 나열된 스냅샷 이외의 모든 파일을 이메일 첨부파일로 첨부할 수 있습니다. 첨부파일 크기는 10MB를 초과하지 않아야 합니다.

예

- **형식 미지정 전자메일:** `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`
- **형식 지정 전자메일:** `epmautomate sendMail jdoe@example.com "Send Formatted Email" "Body=<!DOCTYPE html> <html> <head> <title>EpmAutomate Email Formatting</title> </head> <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

애플리케이션에 대한 액세스가 서비스 관리자만으로 제한되도록 애플리케이션을 관리 모드로 전환합니다.

이 명령은 서비스 관리자가 관리 작업을 수행할 때 사용자가 애플리케이션에서 작업하지 못하도록 하는 데 유용합니다. 사용자가 모드를 다시 변경하여 모든 사용자가 액세스할 수 있도록 할 때까지 애플리케이션은 관리 모드로 유지됩니다.

① Note

이 명령은 [applicationAdminMode](#) 명령을 대체하며, 더 이상 사용되지 않지만 EPM Automate에서 제거되지는 않았습니다.

[getApplicationAdminMode](#) 명령을 사용하여 환경의 현재 상태를 확인합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

```
epmautomate setApplicationAdminMode true|false
```

이 명령에서 애플리케이션을 관리 모드로 전환하려면 `true`를 지정하고, 모든 사용자가 액세스할 수 있도록 일반 모드로 되돌리려면 `false`를 지정합니다.

예

- 애플리케이션을 관리 모드로 전환:
`epmautomate setApplicationAdminMode true`
- 애플리케이션을 일반 운영으로 되돌리기:
`epmautomate setApplicationAdminMode false`

setDailyMaintenanceStartTime

환경의 일별 유지관리가 시작되는 시간(UTC 또는 다른 시간대)을 설정합니다. 환경 유지관리를 꼭 정각에 시작할 필요는 없으며, 유지관리를 시작할 시간 및 분을 설정할 수 있습니다.

이 명령을 사용해도 백업 생성에 대한 Oracle 요구사항을 준수하는 데 방해가 되지 않도록 최근 36시간 동안 일별 유지관리 프로세스가 실행되지 않은 경우 이 명령은 유지관리 시작 시간을 변경하지 않습니다.

① 주

현재 브라우저를 사용하여 환경에 로그인한 서비스 관리자는 로그아웃했다가 로그인한 후에만 새 일별 유지관리 시작 시간을 볼 수 있습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

`epmautomate setDailyMaintenanceStartTime StartTime:StartTime`은 유지관리 프로세스가 시작되어야 하는 시간(24시간제를 사용하는 HH:MM 형식) 및 시간대(선택사항)입니다. 허용 가능한

시작 시간 값 범위는 00:00 - 23:59입니다. 시작 시간이 UTC로 설정되어 있지 않으면 유효한 표준 시간대를 지정하십시오. 예를 들어 태평양 표준시 2:35 pm은 "14:35 America/Los_Angeles"입니다.

예

- 일별 유지관리 시작을 2:20 PM UTC로 설정:
`epmautomate setDailyMaintenanceStartTime 14:20`
- 일별 유지관리 시작을 2:35 PM 태평양 표준시로 설정:
`epmautomate setDailyMaintenanceStartTime "14:35 America/Los Angeles"`

setDemoDates

필요에 따라 Oracle 내부 데모 데이터를 업데이트합니다.

이 명령은 Oracle 내부 데모 데이터를 사용하는 설치 설정에만 사용됩니다.

Account Reconciliation만 해당: 이 명령은 Demo Code 속성 값이 `setdemodates` 또는 `setdemodatesnostatuschange`인 모든 조정의 일자를 원래대로 설정합니다. 이 명령은 현재 기간과 11개의 이전(과거) 기간 등 최대 12개 기간에 대한 조정을 처리합니다. 세 개 이상 기간의 조정에서 Demo Code 속성으로 태그가 지정된 경우 이 명령은 해당 기간을 이전 기간에 있는 것으로 처리합니다. 이 속성 값이 없는 조정은 영향을 받지 않습니다.

- 값이 `setdemodates`인 경우 이 명령은 지정된 일자와 임의의 상태를 기준으로 조정 일자를 재설정합니다.
- 값이 `setdemodatesnostatuschange`인 경우 이 명령은 조정 상태를 변경하지 않고 지정된 일자에 따라 조정 일자를 재설정합니다.

다른 모든 비즈니스 프로세스: 이 명령은 태스크의 시작 및 종료 날짜와 기타 관련 날짜 정보를 재설정하여 데모에 적합한 태스크로 변경합니다. 제공한 Demo Date 값과 함께 태스크 스케줄에서 설정된 `SETDEMODATES` 속성 값을 기반으로 하여 새 태스크 일자를 계산합니다. Demo Date 값을 지정하지 않은 경우 이 명령은 오늘 일자를 사용하여 새 태스크 일자를 계산합니다.

① 주

`SETDEMODATES` 값이 없는 스케줄의 태스크는 영향을 받지 않습니다.

지정한 Demo Date에 따라 이 명령은 태스크와 연계된 모든 일자를 앞으로 이동합니다. 여기에는 코어 실행 시간 일자(시작 일자, 종료 일자 등) 및 내역 일자, 개별 워크플로우 만기 일자, 시작 일자(실제)를 비롯한 보조 일자가 포함됩니다. 태스크 상태는 영향을 받지 않습니다.

적용 대상

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

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

`epmautomate setDemoDates [DEMO_DATE]`, 설명: `DEMO_DATE`는 YYYY-MM-DD 형식의 선택적 날짜입니다. 이 값을 지정하지 않는 경우 조정은 현재 날짜로 재설정됩니다.

예

```
epmautomate setDemoDates 2020-02-15
```

setEJJournalStatus

Financial Consolidation and Close에서 ERP 시스템의 기업 분개 전기 결과를 설정합니다. 이 명령을 사용하여 워크플로우 상태와 관계없이 Post in Progress 상태인 분개의 전기 상태를 업데이트합니다.

이 명령은 ERP 시스템으로의 임포트 상태를 확인할 수 있는 CSV 파일을 사용합니다. [uploadFile](#) 명령을 사용하여 임포트 파일을 환경으로 업로드할 수 있습니다. CSV 파일 형식은 다음과 같습니다.

```
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"
```

메시지 열은 선택사항이며 생략할 수 있습니다.

이 명령은 기업 분개 데이터를 Financial Consolidation and Close에서 익스포트하거나 ERP 시스템으로 임포트하지 않습니다.

적용 대상

재무 통합 및 마감

필수 역할

서비스 관리자

사용법

`epmautomate setEJJournalStatus FILE_NAME.csv [OPERATION=posting|validation]` 설명:

- `FILE_NAME`은 ERP 시스템으로의 임포트 상태가 포함된 CSV 파일을 나타냅니다.
- Operation은 선택사항으로, 수행할 작업입니다. 가능한 값은 `posting` 및 `validation`입니다. 기본값은 `posting`입니다.
 - `operation=posting`은 연도, 기간, 분개 ID, 전기 상태(전기됨/실패), 메시지, 오류 코드 및 오류 메시지 열이 사용된 분개 전기 결과가 포함되어 있는 CSV 파일을 제공합니다. 메시지, 오류 코드, 오류 메시지 열은 선택사항입니다. 파일에 제공된 상태를 기준으로 분개 전기 상태가 업데이트됩니다. 전기 상태가 *전기 진행* 중인 분개만 업데이트됩니다.
 - `operation = validation`은 연도, 기간, 분개 ID, 검증 상태(적합/실패), 메시지, 오류 코드 및 오류 메시지 열이 사용된 분개 검증 결과가 포함되어 있는 CSV 파일을 생성합니다. 메시지, 오류

코드, 오류 메시지 열은 선택사항입니다. 파일에 제공된 상태를 기준으로 분개 검증 상태가 업데이트됩니다. 검증 상태가 *검증 진행* 중인 분개만 업데이트됩니다.

예

- 파일에서 기업 분개 전기 결과 설정:
`epmautomate setEJJournalStatus JournalStatus.csv`
- 파일에서 기업 분개 전기 결과 설정 및 검증 작업 수행:
`epmautomate esetEJJournalStatus JournalStatus.csv Operation=validation`

setEncryptionKey

데이터베이스 액세스에 대한 사용자정의 암호화 키를 설정합니다.

이 명령을 사용하면 고객이 표준 키 관리 회전에 Oracle Fusion Cloud Enterprise Performance Management를 포함하도록 BYOK(Bring Your Own Key) 솔루션이 제공됩니다.

사용자정의 암호화 키는 환경의 다음 일별 유지관리 후에 적용됩니다. [resetService](#) 명령을 실행하여 즉시 활성화할 수 있습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자

사용법

`epmautomate setEncryptionKey key=key` 설명: *key*는 암호화 키로 사용할 사용자정의 문자열로, 길이는 상관이 없습니다.

예

- 암호화 키 설정: `epmautomate setEncryptionKey key=se!m+a2J`
- 암호화 키 제거: `epmautomate setEncryptionKey key=`

setEssbaseQryGovExecTime

쿼리가 종료되기 전에 Oracle Essbase 쿼리가 정보를 검색 및 전달하기 위해 사용할 수 있는 현재 최대 시간(초)을 설정합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

`epmautomate setEssbaseQryGovExecTime TIME`, 여기서 `TIME`는 Essbase 쿼리를 종료할 시간(초)을 나타냅니다. 이 값은 70000을 초과하지 않는 정수여야 합니다.

Oracle은 Essbase 쿼리가 무기한 실행되는 것을 방지하기 위해 이 값을 0(영)으로 설정하지 않는 것을 권장합니다.

예

```
epmautomate setEssbaseQryGovExecTime 600
```

setIdleSessionTimeout

Oracle Fusion Cloud Enterprise Performance Management 환경의 유휴 세션 시간 초과(분)를 변경합니다. 새 세션 시간 초과는 환경의 다음 일별 유지관리 후에 활성화 상태가 됩니다. 이 명령을 사용하여 기본 시간 초과(30분)를 다른 값으로 변경합니다. 이 명령을 사용하여 지정된 기간 동안 세션이 유휴 상태인 경우 사용자를 로그인 페이지로 리디렉션합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자

사용법

`epmautomate setIdleSessionTimeout MINUTES`, 여기서 `MINUTES`는 분 단위의 새 유휴 세션 시간 초과(최소 15 및 최대 480)를 나타냅니다.

예

```
epmautomate setIdleSessionTimeout 30
```

setIPAllowlist(사용되지 않음)

Oracle Fusion Cloud Enterprise Performance Management에 액세스할 수 있는 IP 주소와 CIDR(Classless Inter-Domain Routings)의 허용 목록을 구성합니다. 이 명령은 IPv4 주소 및 CIDR을 추가하거나 제거합니다.

Note

이 명령은 더 이상 사용되지 않으며 이후 업데이트에서 제거될 예정입니다. 이 명령을 사용하는 대신 Oracle Identity Cloud Service에서 네트워크 경계를 설정하여 전체 도메인에 대한 특정 IP 주소 또는 범위로 로그인을 제한합니다. 다음 정보 소스를 참조하십시오.

- *관리자용 시작 가이드*의 네트워크 경계 설정
- Oracle Identity Cloud Service *관리*의 [Oracle Identity Cloud Service 네트워크 경계 관리](#)

Note

- Cloud EPM 환경의 IP 허용 목록을 설정하는 경우 해당 특정 IP 주소에서만 연결을 허용합니다. 이 시나리오에서는 요청하는 환경이 있는 지역의 아웃바운드 IP 주소를 IP 허용 목록에 추가하지 않으면 다른 Cloud EPM의 액세스 요청이 작동하지 않습니다. IP 허용 목록을 설정하는 환경과 다른 환경이 통신할 수 있도록 추가해야 하는 IP 주소는 *작업 가이드*의 Cloud EPM 데이터 센터 및 지역의 아웃바운드 IP 주소를 참조하십시오.
- 환경이 있는 Identity Cloud Service에서 IDCS 네트워크 경계를 사용하는 경우 이 네트워크 경계에도 동일한 IP 주소를 추가해야 합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자

사용법

epmAutomate setIPAllowlist add|remove *FILE_NAME.txt* 설명:

- add는 텍스트 파일에 나열된 IP 주소와 CIDR을 허용 목록에 추가합니다.
- remove는 텍스트 파일에 나열된 IP 주소와 CIDR을 허용 목록에서 삭제합니다.
- *FILE_NAME*은 허용 목록에 추가하거나 제거할 IP 주소와 CIDR을 나열하는 텍스트 파일의 이름입니다. 파일의 각 항목이 라인 바꿈 문자로 구분되어야 합니다. 이 파일을 환경에 업로드하려면 [uploadFile](#) 명령을 사용합니다. 파일의 각 행은 IPv4 주소 또는 다음 형식의 CIDR이어야 합니다.

```
xxx.xxx.xxx.xxx
xxx.xxx.xxx.xxx/n
```

Note

- IPv4 IP 주소만 지원됩니다.
- 개별 IP 주소 대신 CIDR 형식을 사용하여 연속된 IP 주소 범위를 지정합니다.
- 모든 IP 주소에서의 액세스를 허가하는 허용 목록을 사용 안함으로 설정하려면 [getIPAllowlist\(사용되지 않음\)](#) 명령을 사용하여 기존 IP 주소와 CIDR을 파일에 모두 씁니다. 파일을 환경에 업로드한 후, 다음 예제에 표시된 대로 remove 옵션을 사용하여 이 명령을 실행합니다.

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

예

- 허용 목록에 일부 IP 주소와 CIDR 추가:

```
epmAutomate setIPAllowlist add myAddList.txt
```

- 허용 목록에서 일부 IP 주소 제거:

```
epmAutomate setIPAllowlist remove myRemoveList1.txt
```

setManualDataAccess

환경이 응답하지 않으며 고객이 환경을 조사하고 사용할 수 있게 지정하는 서비스 요청을 아직 제공하지 않은 응급 상황에서 Oracle이 환경의 관계형 데이터베이스 및 Oracle Essbase 데이터베이스에 수동으로 액세스할 수 있는지 여부를 지정합니다.

응급 상황이 발생하면 Oracle은 상위 레벨 개발 임원이 독립적인 검증 프로세스 후 관계형 데이터베이스 및 Essbase 데이터베이스에 대한 수동 액세스를 허용하는 내부 프로세스를 따릅니다. Oracle이 명시적 승인 없이는 데이터베이스 액세스를 중지하게 하려면 이 명령을 사용합니다. 또한, 서비스 요청이 열려 있어도 응급 상황이 발생했을 때 Oracle이 관계형 데이터베이스 및 Essbase 데이터베이스에 수동으로 액세스하는 것을 금지하는 옵션도 있습니다.

이 명령을 사용하여 지정한 설정은 즉시 적용됩니다. Oracle의 수동 데이터 액세스는 기본적으로 허용되어 있습니다. 해당 액세스를 취소하려면 이 명령을 사용합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자

사용법

```
epmAutomate setManualDataAccess Allow|Revoke [disableEmergencyAccess=true|false],
설명: disableEmergencyAccess는 선택사항으로, 관계형 데이터베이스 및 Essbase
```

데이터베이스에 대한 수동 액세스를 모두 금지할 것인지 지정합니다. 이 등록정보 값을 true로 설정하면 서비스 요청이 열려 있어도 Oracle이 데이터베이스에 수동으로 액세스하지 못합니다. 기본값은 false입니다.

`disableEmergencyAccess` 값을 true로 지정하면 작동 중단된 환경 문제를 해결하는 데 액세스가 필요한 경우에도 Oracle이 관계형 데이터베이스 및 Essbase 데이터베이스에 액세스할 수 없습니다. 환경이 작동 중단 상태인 경우에는 이 명령을 실행하여 Oracle이 데이터베이스에 수동으로 액세스하도록 허용할 수 없습니다.

예

- 응급 상황에서 명시적 승인 없이 관계형 데이터베이스 및 Essbase 데이터베이스에 수동으로 액세스하도록 부여된 권한을 취소합니다.
`epmautomate setManualDataAccess revoke`
- 응급 시 관계형 데이터베이스 및 Essbase 데이터베이스에 대한 수동 액세스를 허용합니다.
`epmautomate setManualDataAccess allow`
- 서비스 요청이 열려 있는 경우에도 관계형 데이터베이스 및 Essbase 데이터베이스에 대한 수동 액세스를 금지합니다.
`epmautomate setManualDataAccess revoke disableEmergencyAccess=true`

setPeriodStatus

기간에 특정 상태를 설정합니다.

적용 대상

Account Reconciliation

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사전 정의된 고급 사용자, 사용자, 조회자 역할의 사용자는 추가 애플리케이션 역할이 필요할 수 있습니다.

사용법

`epmautomate setPeriodStatus PERIOD STATUS` 설명:

- `PERIOD`는 기간 이름입니다.
- `STATUS`는 OPEN, CLOSED 또는 LOCKED입니다.

예

`epmautomate setPeriodStatus "January 2015" OPEN`

setRestrictedDataAccess

서비스 관리자가 피드백 제공 유틸리티를 사용하는 동안 Oracle에 애플리케이션 스냅샷을 제출하는 데 동의하지 못하도록 Oracle Fusion Cloud Enterprise Performance Management 환경을 구성합니다.

Oracle이 사용자 데이터에 액세스할 수 없도록 하는 정책이 있는 경우 Oracle에 피드백을 제출하는 동안 이 명령을 사용하여 데이터 공유를 관리합니다. 사용자 환경에 대해 제한된 데이터 액세스 권한을 설정하면 피드백 제공 유틸리티에서 표시되는 [애플리케이션 스냅샷 제출] 옵션이 사용 안함으로 설정됩니다.

적용 대상

Account Reconciliation, Oracle Fusion Cloud Enterprise Data Management, Enterprise Profitability and Cost Management, Financial Consolidation and Close, FreeForm, Planning, Planning Modules, Profitability and Cost Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning, Tax Reporting

필수 역할

서비스 관리자

사용법

```
epmautomate setRestrictedDataAccess true|false
```

현재 데이터 액세스 제한 상태를 보려면 [getRestrictedDataAccess](#)를 사용합니다.

예

- 서비스 관리자가 애플리케이션 스냅샷을 제출하는 데 동의하지 못하도록 방지:
`epmautomate setRestrictedDataAccess true`
- 서비스 관리자가 애플리케이션 스냅샷을 제출하는 데 동의하도록 허용:
`epmautomate setRestrictedDataAccess false`

setSubstVars

애플리케이션 또는 큐브 레벨에서 대체 변수를 생성하거나 업데이트합니다.

이 명령은 대체 변수의 여러 값 및/또는 함수를 설정하는 데 사용할 수 없습니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

```
epmautomate setSubstVars CUBE_NAME SUBSTVAR=VALUE [SUBSTVAR=VALUE] 설명:
```

- *CUBE_NAME*은 대체 변수를 생성하거나 업데이트할 대상 큐브(예: Plan1, Plan2)입니다. 애플리케이션 레벨에서 대체 변수를 설정하거나 업데이트하려면 큐브 이름 대신 ALL을 사용합니다.
- *SUBSTVAR*은 값을 설정하거나 업데이트할 대체 변수 이름입니다.
- *VALUE*는 새 대체 변수 값입니다.

예

- 애플리케이션 레벨에서 하나의 대체 변수 생성 또는 업데이트: `epmautomate setSubstVars ALL CurYear=2015 CurPeriod=Jan`

- 큐브 레벨에서 대체 변수 생성 또는 업데이트: `epmautomate setSubstVars Plan2 CurYear=2013 CurPeriod=Jan`

setVirusScanOnFileUploads

환경에서 파일이 Oracle Fusion Cloud Enterprise Performance Management에 업로드되기 전에 바이러스를 검사할 수 있도록 합니다.

Cloud EPM 환경은 바이러스 백신 프로그램을 사용하여 보호됩니다. 이 명령은 업로드 파일에 대한 바이러스 검사를 사용하여 추가적인 보안을 제공합니다. 파일을 업로드하기 전에 검사하면 환경에 바이러스가 업로드될 가능성을 방지할 수 있습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자

사용법

`epmautomate setVirusScanOnFileUploads true|false`
기본적으로 바이러스 검사는 사용하도록 설정되어 있지 않습니다(false로 설정됨). 이 값을 true로 설정하면 Cloud EPM이 모든 업로드 파일을 검사합니다. 파일이 바이러스에 감염된 경우 해당 파일은 환경에 업로드되지 않습니다.

예

- 바이러스 스캔 사용: `epmautomate setVirusScanOnFileUploads true`
- 바이러스 스캔 사용 안함: `epmautomate setVirusScanOnFileUploads false`

simulateConcurrentUsage

사용자를 시뮬레이션하여 환경에서 다양한 동시 작업을 실행합니다.

이 명령은 특정수의 사용자가 특정 작업을 실행하는 동안 서비스가 부하를 받고 있을 때 응답 시간이 허용 가능한지 확인하기 위해 환경 성능을 검증하는 데 사용할 수 있습니다. 예를 들어 사용자 50명이 동시에 서로 다른 POV를 사용하여 양식을 여는 경우 이 명령을 사용하여 성능을 측정할 수 있습니다. 이 명령을 통해 환경의 셀프 서비스 로드 테스트를 수행할 수 있습니다.

이 명령은 특정수의 사용자 및 반복에 대해 지정된 작업을 실행하여 시뮬레이션을 수행합니다. 이 명령은 여러 반복을 실행하여 특정 작업에 대한 최소 시간, 최대 시간, 평균 시간을 계산합니다. 이 명령은 동시 사용 로드 테스트를 수행하기 위해 다음 작업을 지원합니다.

- 양식 열기
- 양식 저장
- 비즈니스 규칙 실행
- 비즈니스 규칙 세트 실행
- 데이터 규칙 실행
- 임시 그리드 열기

- 보고서 실행
- 장부 실행

① Note

이 명령은 Financial Reporting 보고서 및 장부를 지원하지 않습니다. Reports(이전 Management Reports)에 속한 장부 및 보고서만 지원합니다.

⚠ Caution

이 명령은 현재 환경에서 지정된 작업을 실행하며, 작업에 따라 해당 환경에서 데이터를 업데이트합니다. 테스트 환경에서 이 명령을 실행합니다. 프로덕션 환경에서는 이 명령을 실행하지 않는 것이 좋습니다.

이 명령은 ZIP 파일을 사용하며, 이 파일은 입력으로 환경의 인박스에 이미 업로드되었어야 합니다. 하나의 requirement.csv 파일 및 requirement.csv에 포함된 사용 사례를 지원하는 여러 입력 파일이 ZIP 파일에 포함됩니다. 선택사항으로, ZIP 파일에는 사용자 변수 멤버 매핑을 제공하는 UserVarMemberMapping.csv 파일, 일부 사용 사례에 대한 Oracle Smart View for Office 옵션을 제공하는 options.xml 파일, 새 사용자를 생성하는 대신 기존 사용자의 사용자 이름 및 비밀번호를 제공하는 users.csv 파일이 포함될 수 있습니다. 이후에 이 명령은 사용 사례를 시뮬레이션하고 하나 이상의 수신자에게 전자메일로 전송될 수 있는 보고서를 생성합니다.

① Note

이 명령은 피드백 제공 제출을 생성하지 않습니다. 비즈니스 프로세스 화면의 피드백 제공 (*관리자용 시작 가이드*의 피드백 제공 유틸리티를 사용하여 Oracle이 진단 정보를 수집할 수 있도록 지원 참조) 옵션, 피드백 제공 REST API 또는 [feedback](#) 명령을 사용하여 시뮬레이션을 실행한 후 환경에 대한 세부정보를 표시하는 피드백 제공 제출을 생성할 수 있습니다.

사용 시나리오 1: 동시에 양식을 여는 50명 사용자에게 대한 애플리케이션 성능의 수락 테스트.

해결 방법:

1. Library/Global Assumption/ 폴더에 저장된 Exchange Rates라는 양식을 열려는 경우 다음과 비슷한 항목이 포함된 requirement.csv를 생성합니다.


```
# 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. [양식 입력 파일 열기](#)에 지정된 형식을 사용하여 open_form_input.csv를 생성합니다. 50회 사용될 하나의 항목이 이 파일에 포함됩니다. 다른 POV가 포함된 동일한 양식을 열려고 하면 사용하려는 POV 수만큼 많은 항목이 포함됩니다.
3. 사용자 변수 멤버 매핑을 설정해야 하는 경우 [UserVarMemberMapping.csv 파일 생성](#)에 지정된 형식을 사용하여 UserVarMemberMapping.csv를 생성합니다.
4. Smart View 옵션을 사용해야 하는 경우 Smart View 옵션을 options.xml로 익스포트합니다. 자세한 내용은 [options.xml 파일 생성](#)을 참조하십시오.
5. 이전 단계의 파일이 포함된 ZIP 파일을 생성하고 인박스에 업로드합니다.

6. 이전 단계의 ZIP 파일을 입력 파일로 사용하여 `simulateConcurrentUsage` 명령을 실행합니다.

사용 시나리오 2: 계절 사용 증가에 대한 성능 시뮬레이션(예: 회계연도 종료 시). 가정: 사용자 100명이 각 사용자 간에 6초의 지연 시간을 두고 양식을 저장합니다.

해결 방법:

1. `Library/Dashboards/` 폴더에 저장된 `Accessories Revenue`라는 양식을 저장하려는 경우 다음과 비슷한 항목이 포함된 `requirement.csv`를 생성합니다.

```
# Type of Operation,Artifact Name,Number of Users,Input File,Additional Info
Save Form, Library/Dashboards/Accessories Revenue,100,save_form_input.csv,
```

2. [양식 입력 파일 저장](#)에 지정된 형식을 사용하여 `save_form_input.csv`를 생성합니다.
3. 사용자 변수 멤버 매핑을 설정해야 하는 경우 [UserVarMemberMapping.csv 파일 생성](#)에 지정된 형식을 사용하여 `UserVarMemberMapping.csv`를 생성합니다.
4. Smart View 옵션을 사용해야 하는 경우 Smart View 옵션을 `options.xml`로 익스포트합니다. 자세한 내용은 [options.xml 파일 생성](#)을 참조하십시오.
5. 이전 단계의 파일이 포함된 ZIP 파일을 생성하고 인박스에 업로드합니다.
6. 이전 단계의 ZIP 파일을 입력 파일로 사용하고 등록정보 값 `iteration=1` 및 `lagTime=6`을 사용하여 `simulateConcurrentUsage` 명령을 실행합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Strategic Workforce Planning 및 Sales Planning.

필수 역할

서비스 관리자. `testModes 0, 1, 2`를 사용하려면 ID 도메인 관리자 역할도 필요합니다.

사용법

```
epmautomate simulateConcurrentUsage INPUT_FILE.zip [testName=TEST_NAME]
[iterations=COUNT] [notificationEmails="EMAIL_ADDRESS"] [testMode=0|1|2|3|4]
[lagTime=LAG_TIME]. 설명:
```

- `INPUT_FILE.zip`은 사용 사례를 확인하는 ZIP 파일의 이름입니다. 이 명령을 실행하기 전에 [uploadFile](#) 명령(명령어 구문 예: `epmautomate uploadFile "C:/uploads/INPUT_FILE.zip" inbox`)을 사용하여 이 파일을 인박스에 업로드합니다. 이 ZIP 파일에는 다음 파일이 포함되어야 합니다.
 - `requirement.csv`라는 사용 사례 CSV 파일. 이 CSV 파일의 각 라인은 수행할 작업 유형, 아티팩트 이름, 동시 사용자 수, 작업 세부정보를 지정하는 입력 파일, 각 사용 사례 관련 추가 정보를 확인합니다. [requirement.csv 파일 생성](#)을 참조하십시오.
 - 각 작업의 세부정보가 포함된 입력 파일입니다. 다음 항목을 참조하십시오.
 - * [양식 입력 파일 열기](#)
 - * [양식 입력 파일 저장](#)
 - * [비즈니스 규칙 입력 파일 실행](#)
 - * [비즈니스 규칙 세트 입력 파일 실행](#)
 - * [데이터 규칙 입력 파일 실행](#)
 - * [임시 그리드 입력 파일](#)

- * [장부 입력 파일 실행](#)
- * [보고서 입력 파일 실행](#)
- 선택사항으로, UserVarMemberMapping.csv를 입력 ZIP 파일에 추가하여 사용자 변수 멤버 매핑을 제공합니다. [UserVarMemberMapping.csv 파일 생성](#)을 참조하십시오.
- 선택사항으로, Smart View 옵션을 사용하려면 options.xml을 입력 ZIP 파일에 추가합니다. [options.xml 파일 생성](#)을 참조하십시오.
- 선택사항으로, 기존 사용자의 사용자 이름 및 비밀번호를 제공하려면 users.csv를 입력 ZIP 파일에 추가합니다. [users.csv 파일 생성](#)을 참조하십시오.
- testName은 이 테스트의 현재 호출을 확인하기 위한 이름(최대 길이는 50자)입니다. 이 매개변수는 테스트 모드 0, 1, 2, 3에 필요하며 테스트 모드 4에 대해서는 선택사항입니다.

Note

동일한 입력 파일을 사용하여 테스트 모드 1(사용자 생성), 3(테스트 모드 1에서 생성된 사용자 시뮬레이션 실행), 2(테스트 모드 1에서 생성된 사용자 삭제) 순으로 순차적으로 시뮬레이션을 분할하는 경우 동일한 테스트 이름을 사용해야 합니다. 테스트 이름이 *Sim02*인 사용 예:

- 사용자를 생성합니다.

```
epmautomate simulateConcurrentUsage SimTest.zip testName=Sim02
testMode=1
```

- 이전 예에서 생성한 사용자 시뮬레이션을 실행합니다.

```
epmautomate simulateConcurrentUsage SimTest.zip testName=Sim02
testMode=3
```

- 이 시뮬레이션에 대해 생성된 사용자를 삭제합니다.

```
epmautomate simulateConcurrentUsage SimTest.zip testName=Sim02
testMode=2
```

- iterations는 선택사항으로, 응답 시간을 측정하기 위해 requirement.csv에서 확인된 각 사용자 사례를 실행할 횟수를 나타내는 양수입니다. 이 값을 지정하지 않으면 작업이 한 번만 실행됩니다.
- notificationEmails는 선택사항으로, 이 명령의 결과를 전자메일로 보낼 타겟 전자메일 주소를 나타냅니다. 전자메일 주소를 두 개 이상 지정하는 경우에는 세미콜론을 사용하여 구분합니다. 또한 주소 목록을 큰따옴표로 묶습니다. 이 값을 지정하지 않으면 명령을 시작한 사용자에게 결과가 전자메일로 전송됩니다. 이 보고서에 관한 자세한 내용은 [샘플 동시 사용 시뮬레이션 보고서](#)를 참조하십시오.
- [testMode]는 선택사항으로, 동시 사용 시뮬레이션 모드를 지정합니다. 기본값은 0입니다. 허용되는 값은 다음과 같습니다.
 - 0: 시뮬레이션된 사용자를 환경에 추가하고 서비스 관리자 역할에 지정하는 기본 시뮬레이션 모드는 시뮬레이션을 실행한 다음, 시뮬레이션된 사용자를 삭제합니다. 이 모드는 테스트를 한 번만 실행하려는 경우 유용합니다.
시뮬레이션된 사용자 이름은 <testName>User<userNumber> 패턴을 사용합니다. 다음 예에서는 사용되는 테스트 이름이 *시뮬레이션*이라고 가정합니다.
이름: SimulationUser1, SimulationUser2 등

상: SimulationUser1, Simulationser2 등

전자메일 주소: SimulationUser1@example.oracle.com,
SimulationUser2@example.oracle.com 등

사용자 이름: SimulationUser1, SimulationUser2 등

- 1: 시뮬레이션된 사용자를 환경에 추가하고 서비스 관리자 역할에 지정합니다. 시뮬레이션을 실행하거나 시뮬레이션된 사용자를 삭제하지 마십시오. 이 모드를 사용한 후, 모드 3으로 명령을 실행하여 필요한 횟수만큼 시뮬레이션을 실행합니다. 마지막으로, 모드 2로 명령을 실행하여 시뮬레이션된 사용자를 삭제합니다.
- 2: 시뮬레이션된 사용자를 삭제합니다. 사용자를 생성하거나 시뮬레이션을 실행하지 마십시오.
- 3: 사용자를 추가하거나 삭제하지 않고 기존 시뮬레이션된 사용자를 사용하여 시뮬레이션을 실행합니다.
- 4: 입력 ZIP 파일에 포함되어 있는 users.csv 파일에 정의된 사용자를 사용하여 명령을 실행합니다. [users.csv 파일 생성](#)을 참조하십시오. 이 모드는 시뮬레이션할 사용자를 생성하지 않습니다. 대신 기존 사용자를 사용합니다.

동시 사용을 한 번만 실행하려면 testMode=0을 사용합니다. 일련의 테스트를 실행하려면:

- 먼저 testMode=1을 사용하여 명령을 실행하여 시뮬레이션된 사용자를 추가하고 서비스 관리자 역할에 지정합니다.
 - 그런 다음, testMode=3을 사용하여 명령을 실행하여 필요한 횟수만큼 시뮬레이션을 실행합니다.
 - 마지막으로, testMode=2를 사용하여 명령을 실행하여 시뮬레이션된 사용자를 삭제합니다.
- [lagTime]은 선택사항으로, requirement.csv의 각 사용 사례 실행 간에 명령이 대기해야 하는 시간(초, 5초 이상)을 지정합니다. 기본값은 5초입니다. 음수(예: -1), 분수(예: 1/2), 소수 값을 사용하지 마십시오. 한 사용자가 requirement.csv에서 사용 사례 실행을 시작한 후 해당 명령은 다음 사용자의 사용 사례 실행을 시작하기 전에 이 매개변수에 지정된 시간(초) 동안 대기합니다. 사용자 활동은 일반적으로 동시에 시작되지 않기 때문에 이 매개변수를 설정하면 환경의 로드와 관련하여 보다 현실적인 시뮬레이션을 생성하는 데 도움이 됩니다.

예

```
epmautomate simulateConcurrentUsage test_simulation.zip testName=Simulation
iterations=5
notificationEmails="jane.doe@example.com;john.doe@example.com;example@example.com"
lagTime=6
```

skipUpdate

Oracle에서 최대 세 개의 연속 주기 동안 환경에 대한 월별 업데이트 적용을 건너뛰도록 요청하거나, 환경이 기본 코드 라인으로 업데이트되도록 이전에 이 명령을 사용하여 수행된 업데이트 건너뛰기 요청을 모두 제거합니다.

이 명령을 사용하여 현재 환경에 대해 지정된 업데이트 건너뛰기 요청을 나열할 수도 있습니다. 환경의 업데이트 건너뛰기 상태는 이 명령을 사용하여 환경 업데이트를 건너뛴 후 생성되는 활동 보고서(운영 메트릭의)에 포함됩니다. [관리자용 시작 가이드](#)에서 운영 메트릭을 참조하십시오.

현재 월의 주별 패치 및 응급 패치(있는 경우)가 계속 환경에 적용됩니다. 업그레이드 지연이 요청된 월에는 업데이트되지 않습니다.

일회용 패치가 적용된 환경의 업데이트는 건너뛴 수 없습니다. 또한 환경에 현재 적용된 업데이트보다 3개월 넘게 차이가 있는 월별 업데이트는 건너뛴 수 없습니다. 예를 들어 현재 환경에 23.12가 적용된 경우 24.01, 24.02 및 24.03을 건너뛴 수 있지만 24.04는 건너뛴 수 없습니다. 업데이트 지원 작동 방식에 대한 자세한 내용은 *작업 가이드*에서 프로덕션 환경의 업그레이드 지원 요청을 참조하십시오.

① 주

3개월 동안 환경 중 하나에 대해서만 업데이트를 건너뛰는 경우(예를 들어 프로덕션 환경에서는 업데이트를 건너뛰고 테스트 환경에서는 건너뛰지 않는 경우) 해당 환경은 3개 버전 차이가 있습니다. 이 시나리오에서는 이러한 환경 간에 스냅샷을 마이그레이션할 수 없습니다.

예를 들어 테스트 및 프로덕션 환경의 현재 버전이 23.12이며 프로덕션 환경에서만 버전 24.01, 24.02, 24.03에 대한 업데이트를 건너뛴다고 가정해 보겠습니다. 버전 24.03을 사용할 수 있게 되면 테스트 환경의 버전은 24.03이 되지만 프로덕션 환경의 버전은 계속 23.12입니다. 이 경우 테스트 및 프로덕션 환경 간 마이그레이션이 지원되지 않습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

서비스 관리자

사용법

```
epmautomate skipUpdate add|remove|list [version=UPDATE_NUMBER comment="COMMENT"]
```

설명:

- add는 특정 월별 업데이트에 대해 업데이트 건너뛰기 요청을 설정합니다. 다음 매개변수를 지정해야 합니다.
 - version: 건너뛰어야 하는 월별 업데이트입니다. 세 개의 예정된 월별 업데이트 중 하나, 두 개 또는 세 개를 건너뛴 수 있습니다. 예를 들어 환경에 23.12 월별 업데이트가 적용된 경우 24.01, 24.02 또는 24.03 업데이트를 건너뛰거나 이러한 업데이트를 모두 건너뛴 수 있습니다. 세 개의 월별 업데이트를 건너뛰려면 명령을 세 번 실행합니다. 예를 들어 version=24.01, version=24.02, version=24.03을 차례로 사용하여 실행할 때마다 건너뛴 하나의 특정 업데이트를 지정합니다. 이 시나리오에서는 환경이 24.04 월별 주기의 기본 코드 라인으로 업데이트됩니다.
 - 업데이트 건너뛰기가 요청되는 월별 주기와 현재 월별 주기에 간격이 있는 경우 Oracle은 필요에 따라 환경을 업데이트한 다음 지정된 월별 주기의 업데이트를 건너뛵니다. 예를 들어 환경에 23.12 월별 업데이트가 적용되어 있고 버전 24.02, 24.03에 대해 업데이트 건너뛰기를 지정합니다. 이 경우에는 환경이 24.01에서 업데이트되고 24.02, 24.03 업데이트는 건너뛵니다. 환경이 24.04의 기본 코드 라인으로 업데이트됩니다.
 - comment: 업데이트 건너뛰기가 필요한 이유를 설명하는 텍스트입니다. 설명은 큰따옴표로 묶어야 합니다.
- remove: 다음 일별 유지관리 중에 기본 코드 라인으로 업데이트될 수 있도록 환경에 대해 지정된 모든 업데이트 건너뛰기 요청을 제거합니다. 환경에 대한 업데이트 건너뛰기 요청이 두 개 이상이면 이 명령은 해당 요청을 모두 제거합니다.

- list: 현재 환경에 대해 설정된 업데이트 건너뛰기 요청(업데이트 건너뛰기 요청을 수행한 사용자의 로그인 ID, 설명, 업데이트를 건너될 버전, 요청이 생성된 날짜)을 다음 그림과 같이 표시합니다.

```
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
```

예

- 업데이트 건너뛰기 요청: epmautomate skipUpdate add version=24.01 comment="We are in the process of closing the quarter"
- 업데이트 건너뛰기 세부정보 보기: epmautomate skipUpdate list
- 모든 업데이트 건너뛰기 요청 제거: epmautomate skipUpdate remove

snapshotCompareReport

두 개의 스냅샷을 비교하고 스냅샷에 포함된 계산 규칙 및 규칙 세트와 데이터 양식의 차이를 확인하는 스냅샷 비교 보고서를 생성합니다.

보고서는 규칙 및 양식 정의의 차이를 보여 줍니다. 한 스냅샷이 다른 스냅샷에 있는 규칙, 규칙 세트 또는 양식을 포함하고 있는지 여부도 표시되어 있습니다. 규칙 세트의 차이는 포함된 규칙에 따라 확인됩니다. 생성된 보고서는 큐브 또는 차원 등록정보의 차이에 대해 보고하지 않습니다.

- 최근 환경 성능이 저하되었습니다. 이전 스냅샷과 현재 스냅샷을 비교하여 성능 저하를 야기했을 수 있는 차이를 확인할 수 있습니다.
- 기능 동작 또는 성능이 동일할 것으로 예상되는 두 환경 간의 동작 또는 성능 차이가 나타납니다. 이 경우 두 환경의 스냅샷을 비교하여 이 둘 간의 차이를 이해할 수 있습니다.
- 일부 규칙 또는 양식이 환경에서 사라진 것으로 의심됩니다. 이 보고서에서는 사용된 아티팩트를 기존 및 현재 아티팩트와 비교합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

`epmAutomate snapshotCompareReport SOURCE_SNAPSHOT TARGET_SNAPSHOT [reportName=REPORT_NAME.html]`, 설명:

- `SOURCE_SNAPSHOT`은 비교할 소스 스냅샷의 이름입니다. 보고서에는 이 스냅샷에 있는 규칙 및 양식의 차이에 대한 데이터가 포함됩니다.
- `TARGET_SNAPSHOT`은 비교할 타겟 스냅샷의 이름입니다.

① Note

- 스냅샷 이름은 .ZIP 확장자를 포함하거나 포함하지 않고 지정할 수 있습니다.
- 두 스냅샷 모두 환경에서 사용할 수 있어야 합니다. [uploadFile](#), [copyFromObjectStorage](#) 또는 [copySnapshotFromInstance](#) 명령을 사용하여 스냅샷을 환경에 업로드합니다.

- `REPORT_NAME`은 선택사항으로, 보고서 파일의 이름입니다. 기본 보고서 이름은 `SnapshotCompare.html`입니다. [downloadFile](#) 명령을 사용하여 보고서를 다운로드합니다.

예

- `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

엔티티, 계정, 시나리오 및 버전 차원 멤버와 사용자정의 차원 멤버를 정렬합니다.

이 명령은 애플리케이션에 멤버를 로드한 후 차원 멤버를 정렬하는 데 유용합니다.

① 주

이 명령을 사용하여 기간, 연도 및 통화 차원의 멤버를 정렬할 수 없습니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

`epmAutomate sortMember Member [type=children|descendants] [order=ascending|descending]` 설명:

- `Member`는 해당 하위 또는 1차 하위가 정렬되는 상위 멤버의 이름입니다.

- type은 선택적으로 정렬될 멤버를 지정합니다. 허용되는 값은 다음과 같습니다.
 - descendants는 Member 값으로 지정된 상위 멤버의 모든 하위 멤버(1차 하위 및 하위)를 정렬합니다.
 - children은 기본값이며, Member 값으로 지정된 상위 멤버 바로 아래 레벨의 멤버만 정렬합니다.
- order는 선택적으로 정렬 순서를 나타냅니다. 허용되는 값은 다음과 같습니다.
 - ascending은 기본 정렬 순서입니다.
 - descending

예

- 엔티티 차원의 1차 하위를 오름차순으로 정렬: `epmautomate sortMember Entity`
- 엔티티 차원의 모든 하위 멤버를 내림차순으로 정렬: `epmautomate sortMember Entity type=descendants order=descending`

unassignRole

이 명령을 실행하는 사용자를 비롯하여 이 명령에 사용되는 ANSI 또는 UTF-8 인코딩 CSV 파일에 로그인 ID가 포함된 사용자에게서 현재 지정된 역할 하나를 제거합니다. 이 명령을 사용하여 사전 정의된 역할 또는 애플리케이션 역할의 지정을 제거할 수 있습니다.

CSV 파일 형식은 다음과 같습니다.

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

서비스 관리자는 [uploadFile](#) 명령을 사용하여 파일을 환경에 업로드합니다.

① 주

큰따옴표를 사용하여 공백 문자를 포함하는 역할 이름을 묶습니다.

완료되면 명령이 실패한 각 항목에 대한 정보를 콘솔에 인쇄합니다. 이 정보를 검토하여 CSV 파일의 일부 항목에서 명령 실행이 실패한 이유를 알아봅니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 사전 정의된 역할 지정을 제거하는 경우:
 - 서비스 관리자 (또는 사전 정의된 역할 삭제)
 - ID 도메인 관리자 및 사전 정의된 역할
- 애플리케이션 역할을 제거하려면 다음을 수행합니다.

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할

사용법

epmautomate unassignRole *FILE_NAME* *ROLE* 설명:

- *FILE_NAME*은 역할 지정을 취소할 사용자의 로그인 ID가 포함된 CSV 파일의 이름입니다. CSV 확장자는 소문자로 지정하십시오.
사용자 로그인 값은 대소문자를 구분하지 않습니다. 예를 들어 jane.doe@example.com은 Jane.Doe@Example.com 또는 대소문자의 모든 변형과 동일한 것으로 처리됩니다.
- *ROLE*은 다음 중 하나를 나타냅니다. 역할 이름은 대소문자를 구분하지 않습니다.
 - 사전 정의된 역할에 대한 사용자 지정을 제거하는 경우 *ROLE*은 서비스에 적용 가능한 사전 정의된 역할을 확인해야 합니다. *관리/사용 시작 가이드*에서 사전 정의된 역할 이해를 참조하십시오.
 - 애플리케이션 역할에 대한 사용자 지정을 제거하는 경우 *ROLE*은 현재 환경에서 애플리케이션에 속하는 역할을 확인해야 합니다. 애플리케이션 역할은 액세스 제어의 **역할** 탭에 나열됩니다. 각 비즈니스 프로세스의 애플리케이션 역할에 대한 설명은 *Oracle Enterprise Performance Management Cloud용 Access Control 관리*에서 다음 항목을 참조하십시오.
 - * Account Reconciliation
 - * Enterprise Profitability and Cost Management
 - * Planning, FreeForm, Financial Consolidation and Close 및 Tax Reporting
 - * Profitability and Cost Management,
 - * Oracle Enterprise Data Management
 - * Narrative Reporting

예

- 사전 정의된 ID 도메인 역할에서 사용자를 지정 취소합니다.
epmautomate unassignRole remove_roles.csv "Service Administrator"
- 애플리케이션 역할에서 사용자를 지정 취소합니다.
epmautomate unassignRole example_file.csv "Task List Access Manager"

updateGuidedLearningSettings

환경에서 OGL(Oracle Guided Learning)을 사용으로 설정하는 데 필요한 값을 설정합니다.

적용 대상

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

필수 역할

서비스 관리자

사용법

epmautomate updateGuidedLearningSettings [oglAppId=*APP_ID*]
[oglServerUrl=*OGL_SERVER_URL*] 설명:

- `OglAppId`는 선택사항으로, OGL 애플리케이션 ID입니다. OGL 서버에 있는 가이드(콘텐츠)의 논리적 그룹화 ID입니다. 이 값에는 최대 100자가 포함될 수 있습니다.
- `OglServerUrl`은 선택사항으로, 활성 OGL 계정이 있는 OGL 서버의 URL입니다. 기본값은 `https://guidedlearning.oracle.com`입니다. 이 URL은 프로토콜을 지정해야 하며, 최대 300자를 포함할 수 있습니다.

Note

이전 매개변수 중 하나 이상이 필요합니다.

예

```
epmautomate updateGuidedLearnignSettings "oglAppId=123xyz" "oglServerUrl=https://guidedlearning.oracle.com"
```

updateUsers

환경에 업로드된 ANSI 또는 UTF-8 인코딩 CSV(쉼표로 구분된 값) 파일에서 확인된 새 값을 사용하여 ID 도메인에 있는 Oracle Fusion Cloud Enterprise Performance Management 사용자의 전자메일, 이름, 성과 같은 속성을 수정합니다.

서비스 관리자는 [uploadFile](#) 명령을 사용하여 파일을 환경에 업로드합니다. CSV 파일의 모든 열은 필수입니다. 각 열에 적합한 항목을 제공해야 합니다. 이 명령은 필수 값의 각 정의를 검증하고, 누락되었거나 부적합한 각각의 값을 확인할 수 있는 오류 메시지를 표시합니다. 입력 파일 형식은 다음과 같습니다.

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

CSV 파일의 사용자 로그인 값이 ID 도메인에 있는 계정과 일치하는 경우 이 명령은 사용자 계정이 입력 파일의 값과 일치하도록 수정합니다. 사용자 계정은 ID 도메인이 지원하는 모든 환경에 공통적으로 적용되므로 ID 도메인을 공유하는 모든 환경에서 업데이트된 사용자 정보를 사용할 수 있습니다. 사용자에게 지정된 사전 정의된 역할 및 애플리케이션별 역할은 이 명령의 영향을 받지 않습니다.

Note

- 이 명령을 사용하여 사용자 로그인 값을 수정할 수는 없습니다.
- 자체 계정 속성은 수정할 수 없습니다.
- 멀티바이트 문자가 포함된 입력 파일은 UTF-8 문자 인코딩을 사용해야 합니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Strategic Workforce Planning, Sales Planning

필수 역할

ID 도메인 관리자 및 사전 정의된 역할

사용법

`epmautomate updateUsers FILE_NAME` 설명: `FILE_NAME`은 수정할 사용자 정보가 포함된 CSV 파일의 이름입니다.

예

```
epmautomate updateUsers update_user_info.csv
```

upgrade

최신 버전의 EPM Automate를 자동으로 다운로드하고 자동 설치합니다.

[login](#) 명령을 실행하여 세션을 시작하면 EPM Automate가 현재 설치된 버전을 확인합니다. 설치된 버전이 최신 버전이 아닌 경우 최신 버전을 사용할 수 있다는 메시지가 표시됩니다.

① 주

- Windows에서는 기본적으로 EPM Automate가 `C:\Program Files\Oracle\EPM Automate` 폴더에 설치됩니다. 관리자 Windows 그룹에 속한 사용자만 쓰기 액세스 권한을 가지는 보호된 폴더입니다. 관리자 Windows 그룹의 멤버가 아닌 경우 이 명령은 실패합니다.
- EPM Automate가 `C:\Program Files\Oracle\EPM Automate`에 설치되지 않은 경우 이 명령은 현재 설치 디렉토리를 변경하지 않습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 고급 사용자
- 사용자
- 조회자

사용법

```
epmautomate upgrade
```

예

```
epmautomate upgrade
```

uploadFile

로컬 컴퓨터의 파일을 서비스에 업로드합니다. 이 명령을 사용하여 데이터, 메타데이터, 규칙 정의, 차원 정의, 매핑된 트랜잭션, 템플릿 및 백업 스냅샷이 포함된 파일을 업로드합니다.

이 명령은 환경의 기존 파일을 덮어쓰지 않습니다. 업로드 중인 파일의 이름이 업로드 위치에 있는 파일의 이름과 동일한 경우 오류를 표시합니다.

이 명령은 이진 파일(예: .exe, .com, .cmd, .bin, .bat, .msi, .vbs 등의 확장자가 사용된 파일) 및 허용되지 않는 확장자가 사용된 파일 업로드를 허용하지 않습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 임의 사전 정의된 역할 및 마이그레이션 - 관리 애플리케이션 역할

사용법

`epmautomate uploadFile "FILE_NAME" [UPLOAD_LOCATION]` 설명:

- `FILE_NAME`은 파일 이름이며, 파일이 EPM Automate를 실행하는 디렉토리에 없는 경우 절대 경로가 포함됩니다.
- `UPLOAD_LOCATION`은 선택사항으로, 파일을 업로드할 Oracle Fusion Cloud Enterprise Performance Management 위치입니다. 기본 업로드 위치로 파일을 업로드하려면 업로드 위치를 지정하지 마십시오. 자세한 내용은 [기본 파일 위치](#)를 참조하십시오. 지원되는 값은 다음과 같습니다.
 - `inbox` - 인박스로 파일을 업로드합니다. Profitability and Cost Management를 제외한 Cloud EPM 비즈니스 프로세스는 이 위치에서 처리할 파일을 찾습니다.
 - `profitinbox` - Profitability and Cost Management에서 처리할 파일을 업로드합니다.
 - `to_be_imported` - 환경의 다음 일별 유지관리 중 임포트할 Narrative Reporting 스냅샷을 업로드합니다.
 - `inbox/directory_name` - Data Management에서 처리할 파일을 인박스 내 디렉토리로 업로드합니다.
 - `outbox` - Profitability and Cost Management 이외의 비즈니스 프로세스에서 사용되는 아웃박스로 파일을 업로드합니다.
 - `profitoutbox` - Profitability and Cost Management에서 사용되는 아웃박스로 파일을 업로드합니다.

예

- 기본 위치로 스냅샷 업로드:
`epmautomate uploadFile "C:/snapshots/backup_snapshot.zip"`
- Data Management 인박스로 파일 업로드:
`epmautomate uploadFile "C:/pbcdata/quarterlydata.csv" inbox`
- 인박스 내 폴더로 파일 업로드(Data Management의 경우):

```
epmautomate uploadFile "C:/fdmee_data/data.zip" inbox/dm_folder
```

- profitinbox로 파일 업로드(Profitability and Cost Management):
epmautomate uploadFile "C:/profitability_data/data.zip" profitinbox
- C:\temp 디렉토리에서 to_be_imported 위치로 Narrative Reporting 스냅샷 업로드:
epmautomate uploadFile "C:\temp\EPRCS_Backup.tar.gz" to_be_imported

userAuditReport(사용되지 않음)

사용자 감사 보고서(.CSV 파일)를 생성하여 기본 다운로드 위치에 저장합니다.

사용자 감사 보고서에는 지정된 기간(최대 최근 120일) 동안 환경에 로그인한 사용자에 대한 정보가 포함되어 있습니다. 사용자 로그인 ID, 사용자가 로그인한 컴퓨터의 IP 주소, 사용자가 환경에 액세스한 날짜 및 시간(예: July 28, 2022 18:43:21 UTC)이 나열됩니다.

① 주

사용자 감사 보고서에서는 5분 기간 내에 Oracle Fusion Cloud Enterprise Performance Management 환경에 여러 번 로그인한 사용자의 경우 로그인 항목을 하나만 나열합니다.

	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

[downloadFile](#) 명령을 사용하여 생성된 보고서를 컴퓨터로 다운로드할 수 있습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할
- 사전 정의된 역할 및 액세스 제어 - 애플리케이션 역할 보기

사용법

```
epmautomate userAuditReport FROM_DATE TO_DATE REPORT_NAME 설명:
```

- *FROM_DATE*는 감사 보고서를 생성할 기간의 시작 날짜(YYYY-MM-DD 형식)를 나타냅니다.
- *TO_DATE*는 감사 보고서를 생성할 기간의 종료 날짜(YYYY-MM-DD 형식)를 나타냅니다.
- *REPORT_NAME*은 보고서 파일의 이름입니다.

① 주

Cloud EPM에서는 보고서 생성 중에 적합한 날짜 범위만 사용되도록 합니다. 이러한 검증은 시작 및 종료 날짜에 대해 수행됩니다.

- `FROM_DATE`는 현재 날짜부터 허용된 최대 보존 기간(120일)보다 이전일 수 없습니다.
- `TO_DATE`는 `FROM_DATE` 날짜부터 최대 보존 기간보다 이후일 수 없습니다.
- `TO_DATE`는 `FROM_DATE`보다 이전일 수 없습니다.

① 주

이 보고서는 최근 120일에 대해서만 생성될 수 있습니다.

예

```
epmautomate userAuditReport 2016-10-15 2016-12-15 myAuditReport.CSV
```

userGroupReport

액세스 제어에서 사용자가 지정된 그룹을 나열하는 보고서(CSV 파일)를 생성하여 기본 다운로드 위치에 저장합니다.

이 보고서는 그룹에 대한 사용자 지정이 직접(그룹의 멤버) 또는 간접(중첩 그룹의 1차 하위인 그룹의 멤버)인지를 표시합니다.

이 보고서는 사용자의 로그인 이름, 이름, 성, 전자메일 주소, 지정된 그룹 및 지정 유형을 다음 형식으로 확인합니다. 액세스 제어의 [사용자 그룹 보고서] 탭에서 생성된 보고서의 CSV 버전과 동일합니다. 예를 들어 사용자 `jdoue`가 중첩 그룹 `Test2`의 1차 하위인 `Test1` 그룹의 멤버라고 가정합니다. 이 시나리오에서 보고서는 `jdoue`에 대해 다음 정보를 표시합니다.

```
User Login,First Name,Last Name,Email,Direct,Group
jdoue, John, Doe, jdoue@example.com, Yes, test1
jdoue, John, Doe, jdoue@example.com, No, test2
```

[downloadFile](#) 명령을 사용하여 생성된 보고서를 컴퓨터로 다운로드할 수 있습니다.

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할
- 사전 정의된 역할 및 액세스 제어 - 애플리케이션 역할 보기

사용법

`epmautomate userGroupReport REPORT_NAME`, 설명: `REPORT_NAME`은 보고서 파일의 이름입니다.

예

epmautomate userGroupReport UsrGrpReport.CSV

userLoginReport

지정된 날짜 범위 내에 환경에 액세스한 사용자에 대한 정보가 포함된 사용자 로그인 보고서를 생성합니다. 보고서에는 사용자 이름, 환경에 액세스하는 데 사용되는 장치의 IP 주소, 사용자가 환경에 액세스한 날짜 및 시간(UTC)이 나열됩니다.

최근 120일 이내 모든 날짜 범위에 대해 이 보고서를 생성할 수 있습니다.

	A	B	C
1	User Login	IP Address	Access Date and Time
2	jdoe	111.222.22.4	September 11, 2025 19:18:06 UTC
3	jane.doe@example.com	222.333.44.5	September 11, 2025 17:56:46 UTC
4	example@example.com	222.333.55.7	September 11, 2025 17:31:13 UTC

적용 대상

Planning, Planning Modules, FreeForm, Financial Consolidation and Close, Tax Reporting, Account Reconciliation, Profitability and Cost Management, Enterprise Profitability and Cost Management, Oracle Fusion Cloud Enterprise Data Management, Narrative Reporting, Sales Planning, Strategic Workforce Planning

필수 역할

- 서비스 관리자
- 사전 정의된 역할 및 액세스 제어 - 관리 애플리케이션 역할

사용법

epmautomate userLoginReport *FROM_DATE TO_DATE REPORT_NAME*. 설명:

- *FROM_DATE*는 보고서를 생성할 기간의 시작 날짜(YYYY-MM-DD 형식)를 나타냅니다
- *TO_DATE*는 보고서를 생성할 기간의 종료 날짜(YYYY-MM-DD 형식)를 나타냅니다
- *REPORT_NAME*은 보고서 파일(.CSV 형식)의 이름입니다.

① Note

Oracle Fusion Cloud Enterprise Performance Management에서는 보고서 생성 중 적합한 날짜 범위만 사용되도록 합니다. 이러한 검증은 *시작* 및 *종료* 날짜에 대해 수행됩니다.

- *FROM_DATE*는 오늘 날짜부터 허용된 최대 보존 기간(120일)보다 이전일 수 없습니다.
- *TO_DATE*는 *FROM_DATE* 날짜부터 최대 보존 기간보다 이후일 수 없습니다.
- *TO_DATE*는 *FROM_DATE*보다 이전일 수 없습니다.

Note

Cloud EPM은 최근 120일 동안만 사용자 로그인 감사 내역을 유지관리하므로 이 보고서는 최근 120일에 대해서만 생성할 수 있습니다.

예

```
epmautomate userLoginReport 2025-08-01 2025-09-25 myEPMLoginReport.CSV
```

validateConsolidationMetadata

데이터베이스 새로그침 및 통합에 오류가 없도록 환경의 메타데이터를 검증합니다.

[importMetadata](#) 명령을 사용하여 메타데이터를 임포트한 후에 [refreshCube](#) 명령을 실행할 때 데이터베이스 새로그침에 오류가 없도록 이 명령을 실행하여 메타데이터를 검증합니다. 통합 메타데이터가 올바르지 않은 경우 통합도 실패할 수 있습니다.

이 명령은 영(0) 또는 검증 오류 수를 검증이 실행된 콘솔에 표시합니다. 검증 오류를 CSV 파일에 기록하며, 이 파일을 사용하여 메타데이터 오류를 해결할 수 있습니다. [downloadFile](#) 명령을 사용하여 결과 CSV 파일을 로컬 서버로 다운로드합니다.

적용 대상

재무 통합 및 마감

필수 역할

서비스 관리자

사용법

epmautomate validateConsolidationMetadata *LOG_FILE_NAME*, 여기서 *LOG_FILE_NAME*은 이 명령으로 확인된 오류에 대한 정보가 포함되는 파일의 이름입니다.

예

```
epmautomate validateConsolidationMetadata validation_error.csv
```

validateModel

Enterprise Profitability and Cost Management 모델을 검증하고 검증 출력을 파일에 씁니다.

적용 대상

Enterprise Profitability and Cost Management

필수 역할

서비스 관리자

사용법

① Note

대체 변수가 있으면 실제 매개변수 값 대신 대체 변수를 사용할 수 있습니다. 자세한 내용은 [일부 Enterprise Profitability and Cost Management 명령에서 대체 변수 사용](#)을 참조하십시오.

```
epmautomate validateModel "modelName" FILE_NAME.txt [messageType=All|Warning|Error] [ruleStatus=All|Enabled|Disabled]:
```

- `modelName`은 검증할 Enterprise Profitability and Cost Management 모델의 이름입니다. 이 값은 큰따옴표로 묶어야 합니다.
- `FILE_NAME`은 명령에서 모델 검증 출력을 써야 하는 텍스트 파일의 고유 이름입니다. 아웃박스에 생성된 이 파일은 [downloadFile](#) 명령을 사용하여 다운로드할 수 있습니다.
- `messageType`은 선택적으로 모델 유효성 검사 출력에 포함할 정보의 상태입니다. 가능한 매개변수 값은 다음과 같습니다.
 - All, 검증 출력 파일에 오류와 경고를 모두 기록합니다.
 - Error, 검증 출력 파일에 오류만 기록합니다. 기본값입니다.
 - Warning, 검증 출력 파일에 모델 검증 경고만 기록합니다.
- `ruleStatus`는 선택사항으로, 다음 중 하나입니다.
 - All: 선택한 모델에 대한 모든 규칙을 검증합니다. 기본값입니다.
 - Enabled: 사용으로 설정된 규칙만 검증합니다.
 - Disabled: 사용 안함으로 설정된 규칙만 검증합니다.

예

```
epmautomate validateModel "10 Actuals Allocation Process" results.txt
messageType=All ruleStatus=Enabled
```

종료 코드

EPM Automate에서 작업 상태를 나타내는 종료 코드 및 메시지를 반환합니다. 종료 코드는 5개 코드 번호 아래에 그룹화됩니다. 각 코드는 여러 오류 조건을 나타낼 수 있습니다. 수신되는 메시지를 검토하여 오류를 일으킨 특정 조건을 식별합니다.

또한 EPM Automate는 명령 실행에 실패할 때마다 `COMMANDNAME_TIMESTAMP.log` 로그 파일 (예: `uploadfile_16_11_2016_11_27_10.log`)을 생성합니다. 로그 파일은 EPM Automate를 실행하는 컴퓨터에 생성됩니다.

종료 코드 1 오류

Command failed to execute EPM Automate는 이 종료 코드를 사용하여 HTTP 상태 코드 200 및 400과 관련된 메시지를 표시합니다. 이러한 코드는 EPM Automate에서 사용하는 REST API에 의해 반환됩니다.

Insufficient privileges to perform the operation 이 오류는 해당 인증서가 서비스 사인인에 사용된 사용자에게 시도한 작업을 수행할 수 있는 권한이 없는 경우에 표시됩니다.

작업을 수행할 수 있는 권한이 있는 계정으로 사인인합니다. 일반적으로 서비스 관리자만 서비스에서 작업을 수행할 수 있습니다.

Resource does not exist 이 오류는 삭제하거나 다운로드할 파일 또는 스냅샷이 서비스에 없는 경우에 표시됩니다.

`listfiles` 명령을 사용하여 파일 이름 및 해당 위치를 확인합니다.

Invalid snapshot *SNAPSHOT* 이 오류는 서비스에서 익스포트 또는 임포트 작업에 지정된 스냅샷을 검증할 수 없는 경우에 표시됩니다.

적합한 스냅샷을 사용 중인지 확인합니다.

Internal server error. Unable to delete file: *FILE_NAME* Please issue "Provide Feedback" with details 이 오류는 서버 오류 때문에 서비스에서 파일 또는 스냅샷을 삭제할 수 없는 경우에 표시됩니다.

`Feedback` 명령 또는 피드백 제공 기능을 사용하여 이 이슈를 Oracle에 보고합니다.

Invalid file: *FILE_NAME* 이 오류는 삭제하거나 다운로드할 파일 또는 스냅샷이 서비스에 없거나 파일 이름이 필요한 형식이 아닌 경우에 표시됩니다.

`listfiles` 명령을 사용하여 파일 이름 및 해당 위치를 확인합니다.

Recreate is running for a long time. Please contact support 이 오류는 시작한 재생성 작업이 1시간 내에 완료되지 않은 경우에 표시됩니다.

`feedback` 명령 또는 피드백 제공 기능을 사용하여 이 이슈를 Oracle에 보고합니다.

Reset service is running for a long time. Please contact support 이 오류는 시작한 서비스 재설정 작업이 1시간 내에 완료되지 않은 경우에 표시됩니다.

`feedback` 명령 또는 피드백 제공 기능을 사용하여 이 이슈를 Oracle에 보고합니다.

Cannot perform operation. Another instance is in progress. Please try after some time 이 오류는 명령의 다른 인스턴스가 활성 상태일 때 `copysnapshotfrominstance` 명령을 실행하는 경우에 표시됩니다.

다시 명령을 실행하기 전에 `copysnapshotfrominstance` 명령이 완료될 때까지 기다립니다.

Cannot perform operation. Another maintenance script is in progress. Please try after some time 이 오류는 일별 유지관리 또는 서비스 재설정 프로세스가 실행 중일 때 `copysnapshotfrominstance`, `recreate` 또는 `resetservice` 명령을 실행하는 경우에 표시됩니다.

유지 관리 또는 재설정 프로세스가 완료된 후 작업을 재실행하십시오.

Login to source instance failed: *SOURCE_URL* 이 오류는 EPM Automate에서 `copysnapshotfrominstance` 명령을 시작하기 위해 소스 환경에 로그인할 수 없는 경우에 표시됩니다.

소스 환경 액세스에 사용되는 인증서, ID 도메인 및 URL이 적합한지 확인합니다.

Internal server error. Copy snapshot from source instance failed. Please issue "Provide Feedback" with details 이 오류는 `copysnapshotfrominstance` 명령을 실행하는 동안 EPM Automate에서 예기치 않은 문제가 발생한 경우에 표시됩니다.

`feedback` 명령 또는 피드백 제공 기능을 사용하여 이 이슈를 Oracle에 보고합니다.

Internal server error. Please issue "Provide Feedback" with details 이 오류는 Oracle의 정정 작업이 필요한 여러 내부 서버 조건을 나타내기 위해 표시됩니다.

feedback 명령 또는 피드백 제공 기능을 사용하여 이 이슈를 Oracle에 보고합니다.

Snapshot *SNAPSHOT_NAME* already exists. Please delete the snapshot and rerun the command 이 오류는 동일한 이름의 다른 스냅샷이 있는 위치에 스냅샷을 다운로드하거나 업로드하는 경우에 표시됩니다.

기존 스냅샷을 삭제하거나 제거한 후 명령을 재시도합니다.

Error in extracting the snapshot. Please retry with a proper snapshot 이 오류는 importsnapshot 명령을 실행할 때 EPM Automate에서 스냅샷 콘텐츠를 추출할 수 없는 경우에 표시됩니다.

스냅샷이 적합한지 확인합니다.

Internal server error. Unable to open file for write. Please issue "Provide Feedback" with details 이 오류는 가령 감사 보고서를 생성하는 동안 오류가 발생하여 CSV 파일이 생성되거나 업데이트되는 경우에 표시됩니다.

feedback 명령 또는 피드백 제공 기능을 사용하여 이 이슈를 Oracle에 보고합니다.

No matching records found, please select a different date range 이 오류는 userauditreport 명령을 실행하여 감사 데이터를 사용할 수 없는 날짜 범위의 감사 보고서를 생성하는 경우에 표시됩니다.

적합한 날짜 범위를 지정한 후 userauditreport 명령을 재실행합니다. 서비스에서는 최근 365일간의 감사 내역만 유지 관리합니다.

File with same name exists: *FILE_NAME*, please choose a different filename 이 오류는 지정한 감사 보고서 이름을 가진 보고서가 서비스에 있는 경우에 표시됩니다.

서비스에서 기존 파일을 삭제하거나, 다른 파일 이름을 지정한 후 userauditreport 명령을 재실행합니다.

Operation failed with status \$!. Please issue "Provide Feedback" 이 메시지는 서비스 재설정 또는 서비스 재생성 프로세스가 실패하게 하는 내부 서버 오류를 나타냅니다.

feedback 명령 또는 피드백 제공 기능을 사용하여 이 이슈를 Oracle에 보고합니다.

EPMCSS-20643: Failed to add users. File *FILE_NAME.csv* is not found. Please provide a valid file name 이 오류는 추가할 사용자에게 대한 정보가 포함된 지정한 CSV 파일을 인박스에서 사용할 수 없는 경우에 표시됩니다.

listfiles 명령을 사용하여 파일 이름 및 해당 위치를 확인합니다. 파일이 인박스에 없는 경우 [uploadFile](#) 명령을 사용하여 파일을 업로드합니다.

EPMCSS-20644: Failed to remove users. File *FILE_NAME.csv* is not found. Please provide a valid file name 이 오류는 삭제할 사용자에게 대한 정보가 포함된 지정한 CSV 파일을 인박스에서 사용할 수 없는 경우에 표시됩니다.

listfiles 명령을 사용하여 파일 이름 및 해당 위치를 확인합니다. 파일이 인박스에 없는 경우 [uploadFile](#) 명령을 사용하여 파일을 업로드합니다.

20645: Failed to assign role for users. Invalid role name *role*. Please provide a valid role name 이 오류는 CSV 파일에 지정한 역할이 지원되지 않는 경우에 표시됩니다.

파일에 지정된 역할 이름이 Service Administrator, Power User, User 또는 Viewer인지 확인합니다.

listfiles 명령을 사용하여 파일 이름 및 해당 위치를 확인합니다. 파일이 인박스에 없는 경우 [uploadFile](#) 명령을 사용하여 파일을 업로드합니다.

종료 코드 6 오류

Service Unavailable HTTP 오류 404 때문에 서비스를 사용할 수 없습니다.

EPM Automate를 실행 중인 컴퓨터의 브라우저에서 서비스에 액세스하여 서비스 가용성을 확인합니다. 어떤 이유로든 서비스 작동이 중단된 경우 잠시 기다렸다가 다시 시도하거나 Oracle 지원 센터로 문의하십시오.

Read/Write timeout 이 오류는 느린 네트워크 또는 방화벽 이슈로 인해 읽기/쓰기 작업을 수행하는 중 클라이언트 소켓 시간 초과(소켓 시간 초과는 15분임)가 발생하는 경우에 표시됩니다.

네트워크 처리량이 높은 경우 실패한 명령을 재실행합니다. 방화벽 설정으로 인해 실패하는 경우 네트워크 관리자에게 문의하십시오.

종료 코드 7 오류

명령을 실행할 수 없는 경우 EPM Automate에서 이 오류를 표시합니다. 오류 메시지(예: Invalid command)는 오류가 발생한 이유를 지정합니다.

비밀번호 파일 FILE_NAME 부적합한 암호화된 비밀번호 파일, 예를 들어, **PASSWORD_FILE.EPW**를 열 수 없습니다. EPM Automate가 지정된 위치에서 파일을 찾지 못했거나 파일이 필수 형식이 아닙니다.

파일 이름 및 경로를 확인합니다. 부적합한 형식 때문에 파일을 구문분석할 수 없는 경우 [encrypt](#) 명령을 사용하여 파일을 재생성합니다.

Unable to parse password file FILE_NAME 부적합한 형식이거나 파일이 손상되어 EPM Automate에서 암호화된 비밀번호 파일을 구문분석할 수 없습니다.

[encrypt](#) 명령을 사용하여 파일을 재생성할 수 있습니다.

Unable to connect to URL. Root cause MESSAGE 이 오류는 잘못된 URL 때문에 연결을 설정할 수 없는 경우에 표시됩니다. 근본 원인으로 표시되는 메시지에서는 부정확한 URL을 사용하여 발생하는 기본 오류를 자세히 설명합니다.

- 적합한 URL을 사용 중인지 확인합니다.
- 프록시 설정에 의해 인터넷에 연결하기 위해 프록시 서버에 인증해야 하는 경우 프록시 서버 사용자 이름, 도메인 및 비밀번호를 지정하거나 프록시 서버 비밀번호가 포함된 암호화된 비밀번호 파일을 사용하여 사인인합니다. 도움이 필요하면 네트워크 관리자에게 문의하십시오.

Unable to connect to URL Unsupported Protocol 지정한 URL이 지원되지 않는 프로토콜을 사용하기 때문에 login 명령이 실패했습니다. 함께 표시되는 오류 메시지에서 지원되지 않는 프로토콜을 식별합니다.

login 명령과 함께 사용 중인 URL이 보안 프로토콜(HTTPS)을 사용하는지 확인합니다.

Session is not authenticated. Please execute the "login" command before executing any other command login 명령을 사용하여 세션을 설정하기 전에 다른 명령을 실행하려고 했습니다.

다른 명령을 실행하기 전에 [login](#) 명령을 실행하여 환경에 대한 보안 연결을 설정합니다.

Invalid parameter 이 메시지는 명령 매개변수의 부정확한 순서나 필수 명령 매개변수 값이 없어서 발생하는 명령 사용 오류를 나타냅니다.

명령 매개변수 및 지정 순서를 검토하여 수정합니다.

COMMAND_NAME 명령은 **SERVICE_TYPE** 에서 지원되지 않습니다. 비즈니스 프로세스에서 명령을 지원하지 않으므로 EPM Automate에서 연결된 환경에 대해 명령을 실행할 수 없습니다.

각 비즈니스 프로세스에서 지원하는 명령 목록을 보려면 [명령 참조](#)를 참조하십시오.

File does not exist in location: PATH EPM Automate에서 가령 upload 또는 replay 명령을 사용하여 처리하려는 파일을 찾을 수 없습니다.

파일 이름과 경로가 정확한지 확인합니다.

Unable to open file for read: PATH EPM Automate에서 지정한 파일을 읽을 수 없습니다.

파일이 필수 형식인지 확인합니다. EPM Automate를 실행하는 사용자에게 파일에 대한 읽기 액세스 권한이 있는지 확인합니다.

Unable to open file for write: PATH EPM Automate에서 지정한 파일에 쓸 수 없습니다.

파일이 다른 프로세스에서 잠기지 않았는지 확인합니다. EPM Automate를 실행하는 사용자에게 파일에 대한 쓰기 액세스 권한이 있는지 확인합니다.

Invalid command EPM Automate가 지원되지 않는 명령을 발견했습니다.

EPM Automate에서 명령을 지원하는지 확인합니다. 명령 이름의 철자가 정확한지도 확인합니다.

Invalid date format 틀이 부적합한 날짜 형식을 발견했습니다.

보고서 생성 날짜를 지원되는 날짜 형식으로 지정합니다.

FROMDATE DATE cannot be greater than TODATE DATE EPM Automate에서 시작 날짜보다 이전 종료 날짜를 발견했습니다.

지정된 일자 범위의 to date가 from date보다 이후 일자인지 확인합니다.

Exceeded maximum number of feedbacks(6) for a day 이 오류는 feedback 명령을 사용하여 제출할 수 있는 피드백 수를 초과할 경우에 표시됩니다.

File with the same name already exists in the download path PATH.

Please delete the file and rerun the command 이 오류는 다운로드되는 파일의 이름과 일치하는 파일이 이미 있는 위치로 파일을 다운로드하는 경우에 표시됩니다.

기존 파일을 삭제하거나 이름을 바꾸거나 이동한 후 명령을 다시 실행합니다.

File is empty: PATH 이 오류는 재생 파일에 콘텐츠가 없는 경우에 표시됩니다.

재생 파일(CSV 파일)에 replay 명령을 실행하는 데 사용할 HAR 파일의 이름과 인증서(사용자 이름 및 비밀번호)가 나열되어 있는지 확인합니다.

Unable to encrypt the password as localhost cannot be resolved. Ensure that hostnames are mapped properly to IP addresses 이 오류는 컴퓨터의 호스트 파일에 127.0.0.1 주소의 localhost 대신 서버 이름이 포함되어 있으므로 EPM Automate가 localhost 정의를 MAC 주소로 확인할 수 없는 경우에 표시됩니다.

호스트 파일에서 127.0.0.1의 서버 이름으로 localhost가 지정되어 있는지 확인합니다.

Snapshot Name is invalid 이 오류는 이름을 바꿀 스냅샷의 이름을 지정하지 않은 경우에 표시됩니다.

환경에서 사용가능한 스냅샷의 이름을 지정합니다.

New Snapshot Name is invalid 이 오류는 스냅샷의 새 이름을 지정하지 않은 경우에 표시됩니다.

스냅샷의 새 이름을 지정합니다.

Invalid snapshot name: {0}. Characters \\/*?"<>| are not allowed 이 오류는 스냅샷 이름에 공백, \ (백슬래시), / (슬래시), * (별표), ? (물음표), " (따옴표), < (보다 작음), > (보다 큼) 등의 특수 문자가 포함된 경우에 표시됩니다.

이러한 특수 문자를 포함하지 않는 새 스냅샷 이름을 지정합니다.

Unable to rename snapshot : {0}. There could be another process accessing it. Please try after sometime 이 오류는 다른 프로세스에서 사용 중이므로 EPM Automate에서 스냅샷에 대한 배타적 잠금을 획득할 수 없는 경우에 표시됩니다.

스냅샷을 사용 중인 현재 작업이 완료될 때까지 기다렸다가 다시 시도하십시오.

Snapshot {0} already exist. Please delete the snapshot and re-run the command 이 오류는 새 스냅샷 이름이 환경의 기존 스냅샷 이름과 동일한 경우에 표시됩니다.

다른 스냅샷 이름을 사용하거나, `deletefile` 명령을 사용하여 기존 스냅샷을 삭제합니다.

종료 코드 9 오류

Invalid credentials 이 오류는 `login` 명령에 사용한 사용자 이름 또는 비밀번호가 부정확한 경우에 표시됩니다.

연결하려는 환경에 적합한 인증서를 지정합니다.

Authentication failed while executing command. 을(를) 다시 시도하십시오. 이 오류는 `login` 이외의 명령을 실행하는 동안 기본 인증이 실패한 경우에 표시됩니다. 이 오류는 명령 실행을 재시도할 때(최대 3회) HTTP 호출에 대해서도 발생할 수 있습니다.

종료 코드 11 오류

내부 서버 오류입니다. 수동 재설정 서비스로 인해 현재 Oracle EPM Cloud Service 환경을 사용할 수 없습니다. 환경 재설정이 진행 중일 때 EPM Automate 명령이 실행되는 경우 이 오류가 표시됩니다.

Internal server error MESSAGE 이 오류는 EPM Automate에서 HTTP 연결과 관련 없는 알 수 없는 예외가 발생하는 경우에 표시됩니다. 서버 오류 503 및 500을 포함합니다.

Unable to connect to URL: MESSAGE 이 오류는 서버를 사용할 수 없는 경우에 표시됩니다. 오류 메시지는 명령을 실패하게 한 예외를 나타냅니다.

서버를 사용할 수 없는 경우 Oracle 지원 센터로 문의하십시오. 메시지에서 URL 관련 이슈를 나타내는 경우 사용 중인 URL이 적합한지 확인합니다.

3

명령 실행 샘플 시나리오

EPM Automate는 여러 공통 Oracle Fusion Cloud Enterprise Performance Management 관리 태스크를 자동화하는 데 사용될 수 있습니다.

- [모든 서비스의 샘플 시나리오](#)
- [Planning, Consolidation, Tax Reporting, Enterprise Profitability and Cost Management 샘플 시나리오](#)
- [Account Reconciliation의 샘플 시나리오](#)
- [Profitability and Cost Management의 샘플 시나리오](#)
- [Cloud EDM의 샘플 시나리오](#)

샘플 스크립트 복사 정보

이 문서의 PDF 버전에서 샘플 스크립트를 복사하지 마십시오. 스크립트를 사용할 수 없게 만드는 라인 바꿈 및 바닥글 정보를 방지하려면 [EPM Automate 작업](#)의 HTML 버전에서 샘플 스크립트를 복사하는 것이 좋습니다.

모든 서비스의 샘플 시나리오

다음 시나리오에서는 Oracle Fusion Cloud Enterprise Performance Management 환경에서 특정 작업을 수행하는 데 일반적으로 사용할 수 있는 일련의 명령을 설명합니다.

참조:

- [애플리케이션 스냅샷을 컴퓨터에 백업](#)
이 시나리오에서는 일별 서비스 유지 관리 시 생성된 스냅샷을 로컬 컴퓨터에 백업하는 프로세스를 자동화하는 방법을 설명합니다.
- [사용자에게 일별 유지관리 완료에 대해 알리기](#)
Oracle Fusion Cloud Enterprise Performance Management 환경의 일별 유지관리는 일반적으로 해당 작업에 배정된 1시간보다 훨씬 더 짧은 시간이 소요됩니다.
- [Oracle Object Storage로 또는 Oracle Object Storage에서 스냅샷 복사](#)
- [사용자 생성 및 사전 정의된 역할에 사용자 지정](#)
이 섹션의 스크립트를 사용하여 사용자를 생성하고 ID 도메인의 사전 정의된 역할에 지정합니다.
- [라이센스를 가진 사용자\(역할에 지정된 사용자\) 수 계산](#)
이 섹션의 스크립트를 사용하여 환경의 사용자 수를 계산하는 역할 지정 보고서를 생성할 수 있습니다.
- [역할에 지정된 사용자 감사 보고서 생성](#)
이 섹션의 스크립트를 사용하여 환경의 사전 정의된 역할에 지정된 사용자에 대한 감사 보고서 생성 프로세스를 자동화할 수 있으며 선택적으로 수신자에게 전자메일로 보낼 수 있습니다.
- [역할 지정 및 취소 감사 보고서 생성](#)
이 섹션의 PowerShell 스크립트를 사용하여 환경의 역할 지정 및 역할 취소를 자세히 설명하는 감사 보고서를 생성하는 프로세스를 자동화할 수 있습니다.

- [개인 정보 보호 법률을 준수하도록 액세스 로그 및 활동 보고서 마스크](#)
이 섹션의 스크립트를 사용하여 개인정보보호 법률을 준수하도록 활동 보고서 또는 액세스 로그의 정보를 마스킹하는 프로세스를 자동화할 수 있으며 선택적으로 수신자에게 전자메일로 보낼 수 있습니다.
- [로컬 컴퓨터로 활동 보고서 다운로드 자동화](#)
이 섹션의 스크립트를 사용하여 환경의 활동 보고서를 로컬 컴퓨터로 다운로드하는 프로세스를 자동화할 수 있습니다.
- [환경에서 액세스 로그 다운로드](#)
이 섹션의 스크립트를 사용하여 환경의 액세스 로그를 로컬 컴퓨터로 다운로드하는 프로세스를 자동화할 수 있습니다.
- [환경 복제 자동화](#)
이 섹션의 스크립트를 사용하여 환경 복제를 자동화할 수 있습니다.
- [기본 환경에서 일별 유지관리가 완료된 후 매일 기본 환경에서 대기 환경으로 복제](#)
대기 환경을 기본 환경과 같은 최신 상태로 유지하려면 기본 환경에서 일별 유지관리가 완료된 직후 이 스크립트를 사용하여 Oracle Fusion Cloud Enterprise Performance Management 기본 환경을 대기 환경으로 복제합니다.
- [환경에서 불필요한 파일 제거](#)
이러한 스크립트를 사용하여 환경에서 불필요한 파일을 제거합니다.
- [환경에서 파일 찾기 및 다운로드](#)
이 섹션의 샘플 스크립트를 사용하면 텍스트 문자열을 와일드카드로 사용하여 Oracle Fusion Cloud Enterprise Performance Management 환경에서 하나 이상의 파일을 다운로드하는 프로세스를 자동화할 수 있습니다.
- [감사를 위해 이전 Cloud EPM 환경 재생성](#)
이 섹션의 스크립트를 사용하여 Oracle Fusion Cloud Enterprise Performance Management 환경에 대한 최신 스냅샷 라이브러리를 유지관리하는 셀프 서비스 솔루션을 생성합니다. 최신 스냅샷 라이브러리를 업그레이드하고 유지관리하기 위한 전용 환경이 필요합니다.
- [데이터베이스 액세스 감사 및 준수 자동화](#)
이 섹션의 PowerShell 및 Bash 셸 스크립트를 사용하여 EPM Automate 명령을 통해 수동 데이터베이스 액세스에 대한 감사 및 준수 데이터를 수집할 수 있습니다.
- [사용자 및 사전 정의된 역할 지정 복제](#)
이 섹션의 스크립트는 환경의 사용자 및 사전 정의된 역할 지정을 다른 환경으로 마이그레이션하는 데 도움이 됩니다.
- [분기별 Cloud EPM 업그레이드 주기 생성](#)
이러한 스크립트를 사용하여 Cloud EPM 환경이 2주 테스트 주기에 따라 분기별로 업데이트되도록 업데이트를 건너뛰는 셀프 서비스 솔루션을 생성합니다. 이 경우 프로덕션 환경은 테스트 환경 2주 후에 업데이트됩니다.
- [6주 테스트 주기로 분기별 Cloud EPM 업그레이드 주기 생성](#)
이 섹션의 스크립트를 사용하여 Oracle Fusion Cloud Enterprise Performance Management 환경이 6주 테스트 주기에 따라 분기별로 업데이트되도록 업데이트를 건너뛰는 셀프 서비스 솔루션을 생성합니다. 이 경우 프로덕션 환경은 테스트 환경 6주 후에 업데이트됩니다.

애플리케이션 스냅샷을 컴퓨터에 백업

이 시나리오에서는 일별 서비스 유지 관리 시 생성된 스냅샷을 로컬 컴퓨터에 백업하는 프로세스를 자동화하는 방법을 설명합니다.

- 유지 관리 기간 중에 생성된 애플리케이션 스냅샷(Artifact Snapshot)을 다운로드합니다.
- 타임스탬프를 추가하여 다운로드한 스냅샷의 이름을 바꿉니다.
- 필요한 경우 가장 오래된 백업을 삭제하여 10개의 백업을 유지 관리합니다.

Note

- 이 스크립트는 Narrative Reporting을 백업하는 데 사용할 수 없습니다.
- 이 스크립트를 사용자의 용도에 맞게 고치는 경우 필요에 따라 런타임 매개변수(url, user, password, NumberOfBackups)의 값을 수정합니다.

Windows 태스크 스케줄러를 사용하여 스크립트 스케줄을 예약하는 방법에 대한 자세한 내용은 [스크립트 실행 자동화](#)를 참조하십시오.

Windows 샘플 스크립트

아래와 유사한 스크립트가 포함된 배치(.bat) 또는 셸(.sh) 파일을 생성하여 스냅샷 다운로드를 자동화합니다.

```
@echo off
rem Sample script to download and maintain 10 maintenance backups
rem Update the following parameters

SET url=https://example.oraclecloud.com
SET user=ServiceAdmin
SET password=Example.epw
SET SnapshotName="Artifact Snapshot"
SET NumberOfBackups=10

rem EPM Automate commands
call epmautomate login %user% %password% %url%
IF %ERRORLEVEL% NEQ 0 goto :ERROR
    call epmautomate downloadfile %SnapshotName%
IF %ERRORLEVEL% NEQ 0 goto :ERROR
    call epmautomate logout
IF %ERRORLEVEL% NEQ 0 goto :ERROR

rem Rename downloaded Artifact Snapshot, keep the last 10 backups
Set Timestamp=%date:~4,2%_%date:~7,2%_%date:~10,2%%
Set Second=%time:~0,2%%time:~3,2%
ren %SnapshotName%.zip %SnapshotName%_%Timestamp%_%Second%.zip

SET Count=0
FOR %%A IN (%SnapshotName%*.*) DO SET /A Count += 1
IF %Count% gtr %NumberOfBackups% FOR %%A IN (%SnapshotName%*.*) DO del "%%A"
&& GOTO EOF
:EOF

echo Scheduled Task Completed successfully
exit /b %errorlevel%
:ERROR
echo Failed with error #%errorlevel%.
exit /b %errorlevel%
```

Linux/UNIX 샘플 스크립트

아래와 유사한 스크립트가 포함된 쉘(.sh) 파일을 생성하여 스냅샷 다운로드를 자동화합니다. 비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오.

```
#!/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/jdk_17.0.12/

export JAVA_HOME=${javahome}

printResult()
{
    op="$1"
    opoutput="$2"
    returncode="$3"

    if [ "${returncode}" -ne 0 ]
    then
        echo "Command failed. Error code: ${returncode}. ${opoutput}"
    else
        echo "${opoutput}"
    fi
}

processCommand()
{
    op="$1"
    date=`date`

    echo "Running ${epmautomatescript} ${op}"
    operationoutput=`eval "$epmautomatescript $op"`
    printResult "$op" "$operationoutput" "$?"
}

op="login ${user} ${password} ${url}"
processCommand "${op}"

op="downloadfile \"${snapshotname}\""
processCommand "${op}"

op="logout"
processCommand "${op}"

# Renames the downloaded artifacts, keeps the last 10 backups
timestamp=`date +%m_%d_%Y_%I%M`
mv "${snapshotname}.zip" "${snapshotname}_${timestamp}.zip"
```

```
((numberofbackups+=1))
ls -tp ${snapshotname}*.zip | grep -v '/$' | tail -n +${numberofbackups} |
xargs -d '\n' -r rm --
```

사용자에게 일별 유지관리 완료에 대해 알리기

Oracle Fusion Cloud Enterprise Performance Management 환경의 일별 유지관리는 일반적으로 해당 작업에 배정된 1시간보다 훨씬 더 짧은 시간이 소요됩니다.

해당 환경의 실제 일별 유지관리 기간은 활동 보고서의 "작업 메트릭" 섹션에서 "일별 유지관리 기간 (분)" 메트릭의 값으로 기록됩니다. 환경을 사용하기 전에 전체 시간 동안 기다리지 않으려면 이 스크립트의 사용자정의 버전을 사용하여 사용자가 활동을 재개할 수 있도록 일별 유지관리가 완료됨을 사용자에게 알립니다.

Windows 스크립트

다음 PowerShell 스크립트를 복사하여 `daily_maintenance_completed.ps1`을 생성합니다. 사용할 스크립트를 업데이트하는 방법에 대한 내용은 [스크립트 실행](#)을 참조하십시오.

```
# Daily Maintenance Completed Notification script
#
# Update the following parameters
# -----
$emailaddresses=user1@oracle.com,user2@oracle.com
# -----

$username=$args[0]
$password=$args[1]
$url=$args[2]

if (($args.count) -ne 3) {
    echo "Usage: ./daily_maintenance_completed.ps1 <USERNAME> <PASSWORD>
<URL>"
    exit 1
}

$amw_time=""

function getDailyMaintenanceStartTime {
    $amwstring=$(epmautomate.bat getDailyMaintenanceStartTime)
    $elements=$amwstring.split(' ')
    $amwtime=$elements[0]
    return $amwtime
}

function goToSleep ($amw_time){
    $current_mdy=Get-Date -AsUTC -UFormat "%m/%d/%Y"
    $current_date_time=Get-Date -AsUTC -UFormat "%m/%d/%Y %H:%M:%S"
    $current_epoch=Get-Date -Date $current_date_time -UFormat "%s"
    $target_date_time=[DateTime]"${current_mdy} ${amw_time}"
    $target_epoch=Get-Date -Date $target_date_time -UFormat "%s"
    $sleep_seconds=$target_epoch - $current_epoch

    # Today's AMW start time has already passed, so add 24 hours to
```

```

sleep_seconds
  if ($sleep_seconds -lt 0) {
    $sleep_seconds=$sleep_seconds + 86400
  }

  $sleep_ts=New-TimeSpan -Seconds ${sleep_seconds}
  $sleep_hms="${sleep_ts}" -replace '^\d+?\.'

  echo "Current time is ${current_date_time}. Sleeping for ${sleep_hms},
until daily maintenance start time of ${amw_time}."
  Start-Sleep -Seconds $sleep_seconds
}

function attemptLogin {
  $serverdown=$False
  while ($true) {
    epmautomate.bat login ${username} ${password} ${url}
    if ($?) { # login succeeded
      if ($serverdown) { # server has been brought down
        echo "Daily maintenance processing has completed ..."
        break
      } else { # server has not yet been brought down
        echo "Daily maintenance processing has not yet started.
Sleeping for 2 minutes before the next check ..."
        Start-Sleep -Seconds 120
      }
    } else { # login failed
      if ($serverdown) { # server has been brought down
        echo "Waiting for daily maintenance processing to complete.
Sleeping for 2 minutes before the next check ..."
        Start-Sleep -Seconds 120
      } else { # server has not yet been brought down
        echo "Daily maintenance processing is now beginning. Sleeping
for 2 minutes before the next check ..."
        Start-Sleep -Seconds 120
        $serverdown=$True
      }
    }
  }
}

function sendNotification {
  $servername=$url.split("/")[2];
  $subject="Daily maintenance processing has completed"
  $formattedmessage="Daily maintenance processing has completed for server $
{servername}"
  $emailaddresses=${emailaddresses}.replace(',',';')

  echo "Mailing report"
  epmautomate.bat sendmail "${emailaddresses}" "${subject}" Body="$
{formattedmessage}"
}

echo "Beginning daily maintenance completion notification script."
echo "Logging into server ..."
epmautomate.bat login ${username} ${password} ${url}

```

```

$amwtime=getDailyMaintenanceStartTime
goToSleep ($amwtime)
attemptLogin
sendNotification
echo "Logging out of server ..."
epmautomate.bat logout
echo "Script processing has completed."

```

Linux/UNIX 스크립트

다음 스크립트를 복사하여 `daily_maintenance_completed.sh`를 생성합니다. 사용할 스크립트를 업데이트하는 방법에 대한 내용은 [스크립트 실행](#)을 참조하십시오.

```

#!/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
        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."
```

서버측 Groovy 스크립트

다음 코드를 복사하여 `daily_maintenance_completed.sh` Groovy 스크립트를 생성합니다. 사용할 스크립트를 업데이트하는 방법에 대한 내용은 [스크립트 실행](#)을 참조하십시오.

```
// 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.")

```

스크립트 실행

Windows 및 Linux/UNIX

- 이전 섹션에서 스크립트를 복사하여 `daily_maintenance_completed.ps1` 또는 `daily_maintenance_completed.sh`를 생성합니다.
- 스크립트를 업데이트합니다.
 - Windows:** 일별 유지관리가 완료될 때 통지를 받아야 하는 전자메일 주소의 심표로 구분된 목록으로 `emailaddresses` 값을 업데이트합니다.
 - Linux/UNIX:** 다음 변수를 업데이트합니다.
 - `epmautomatescript`는 EPM Automate 실행파일의 위치로 업데이트합니다. 예: `epmautomatescript="/home/utils/EPMAutomate/bin/epmautomate.sh"`
 - `javahome`은 EPM Automate에서 사용되는 JDK가 설치된 디렉토리로 업데이트합니다. 예: `"/home/user1/jdk_17.0.12"`
 - `emailaddresses`는 일별 유지관리가 완료될 때 통지를 받아야 하는 전자메일 주소의 심표로 구분된 목록으로 업데이트합니다. 예: `jdoh@example.com,jane_doh@example.com`
- 명령 창 또는 콘솔에서 `daily_maintenance_completed` 스크립트가 저장된 폴더로 이동합니다.
- 다음 명령을 실행합니다.
 - Windows:** `./daily_maintenance_completed.ps1 USERNAME PASSWORD URL`
 - Linux/UNIX:** `./daily_maintenance_completed.sh USERNAME PASSWORD URL`. 여기서:
 - `USERNAME`은 서비스 관리자의 사용자 이름입니다
 - `PASSWORD`는 서비스 관리자의 비밀번호입니다
 - `URL`은 Cloud EPM 환경의 URL입니다.

서버측 Groovy:

- 이전 섹션에서 스크립트를 복사하여 `daily_maintenance_completed.groovy` Groovy 스크립트를 생성합니다.
- 다음 값을 업데이트합니다.
 - `username`은 서비스 관리자의 사용자 이름으로 업데이트합니다.
 - `password`는 서비스 관리자의 비밀번호로 업데이트합니다.
 - `url`은 일별 유지관리 완료 통지를 해야 하는 Cloud EPM 환경의 URL로 업데이트합니다. 예를 들면 다음과 같습니다. 예: `https://testExample-idDomain.pbcs.us1.oraclecloud.com`
 - `emailaddresses`는 일별 유지관리가 완료될 때 통지를 받아야 하는 전자메일 주소의 쉼표로 구분된 목록으로 업데이트합니다.
- Cloud EPM 비즈니스 프로세스에서 Groovy 화면을 사용하거나 [runBusinessRule](#)을 사용하여 스크립트 실행을 자동화합니다. 자세한 내용은 다음 정보 소스를 참조하십시오.
 - [EPM Automate를 설치하지 않고 명령 실행](#)
 - Planning* 관리의 Groovy 규칙 사용

Oracle Object Storage로 또는 Oracle Object Storage에서 스냅샷 복사

이 항목에는 다음 태스크를 완료하기 위한 샘플 스크립트가 포함되어 있습니다.

- Oracle Fusion Cloud Enterprise Performance Management에서 Oracle Object Storage 버킷으로 Artifact Snapshot(유지관리 스냅샷)을 복사하고 스냅샷 복사 날짜를 추가하여 이름을 바꿉니다.
- Oracle Object Storage 버킷에서 Cloud EPM으로 백업 스냅샷을 복사합니다.

이 섹션의 스크립트에서는 Oracle Object Storage에 스냅샷을 보관할 버킷이 이미 생성되어 있다고 가정합니다. 이러한 스크립트를 실행하기 전에 다음 자리 표시자를 업데이트하여 용도에 맞게 스크립트를 사용자정의하십시오.

Table 3-1 매개변수 및 해당 값

자리 표시자	예상 값
<code>JAVA_HOME</code>	EPM Automate에서 사용하는 JDK가 설치된 디렉토리입니다. 예: <code>./home/JDK/bin</code>
<code>epmautomateExe</code>	EPM Automate가 설치된 디렉토리입니다. 예: <code>./home/utills/EPMAutomate/bin</code>
<code>cloudServiceUser</code>	Cloud EPM 서비스 관리자의 사용자 ID입니다. 예: <code>John.doe@example.com</code>
<code>cloudServicePassword</code>	서비스 관리자의 비밀번호 또는 비밀번호 파일의 위치입니다. 비밀번호에 특수 문자가 포함된 경우 특수 문자 처리 를 참조하십시오. 예: <code>ex_PWD_213</code>
<code>cloudServiceUrl</code>	Artifact Snapshot을 복사할 Cloud EPM 환경의 URL입니다. 예: <code>https://test-cloud-id_Dom.pbcs.us1.oraclecloud.com</code>
<code>objectStorageUser</code>	Oracle Object Storage에 있는 사용자의 사용자 ID입니다. 스냅샷을 Object Storage로 복사하려면 이 사용자에게 스냅샷이 복사되는 버킷에 대한 쓰기 액세스 권한이 있어야 합니다. 스냅샷을 Object Storage에서 복사하려면 이 사용자에게 스냅샷이 복사되는 버킷에 대한 읽기 액세스 권한이 있어야 합니다. 예: <code>jDoe</code>

Table 3-1 (Cont.) 매개변수 및 해당 값

자리 표시자	예상 값
objectStoragePassword	objectStorageUser의 비밀번호입니다. 예: example_PWD
objectStorageBucketUrl	스냅샷이 복사되는 Oracle Object Storage 버킷의 URL입니다. URL 형식에 대한 내용은 다음 정보 소스를 참조하십시오. <ul style="list-style-type: none"> copyToObjectStorage copyFromObjectStorage 예: https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/axaxnpcrorw5/bucket-20210301-1359
snapshot	Oracle Object Storage 버킷에서 복사할 스냅샷의 이름입니다. 예: Artifact Snapshot20210429.zip

Cloud EPM에서 Oracle Object Storage로 스냅샷을 복사하는 샘플 EPM Automate 스크립트

다음 스크립트를 사용자정의하고 실행하여 Artifact Snapshot 이름을 바꾸고 Cloud EPM에서 Oracle Object Storage 버킷으로 복사합니다.

```
#!/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
```

Oracle Object Storage에서 Cloud EPM으로 스냅샷을 복사하는 샘플 EPM Automate 스크립트

다음 스크립트를 사용자정의하고 실행하여 특정 날짜의 Artifact Snapshot을 Oracle Object Storage 버킷에서 Cloud EPM으로 복사합니다.

```
#!/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>
$pmautomateExe login ${cloudServiceUser} ${cloudServicePassword} $
${cloudServiceUrl}
$pmautomateExe copyFromObjectStorage ${objectStorageUser} $
${objectStoragePassword} "${objectStorageBucketUrl}/${snapshot}"
$pmautomateExe logout
exit 0
```

사용자 생성 및 사전 정의된 역할에 사용자 지정

이 섹션의 스크립트를 사용하여 사용자를 생성하고 ID 도메인의 사전 정의된 역할에 지정합니다.

이러한 스크립트는 EPM Automate 명령을 사용하여 다음 활동을 완료합니다.

- ID 도메인 관리자 역할을 가진 서비스 관리자로 환경에 로그인합니다.
- 환경에서 그룹 및 멤버십 정보를 익스포트하여 지정하는 스냅샷(예: example_snapshot.zip)을 재생성합니다. 이 스냅샷을 생성하기 위해 마이그레이션을 사용하여 이전에 그룹 및 멤버십 아티팩트를 익스포트했다고 가정합니다.
- 스냅샷(example_snapshot.zip)을 로컬 디렉토리에 다운로드합니다.
- 환경에서 스냅샷(example_snapshot.zip)을 삭제합니다.
- 환경에서 사인아웃합니다.
- 사용자정의 코드를 추가한 작업을 수행합니다. 해당 작업에는 다음이 포함될 수 있습니다.
 - example_snapshot.zip의 콘텐츠 추출
 - 이름,성,전자메일,사용자 로그인 형식으로 HSS-Shared Services\resource\External Directory\Users.csv에 새 사용자 정보 추가. 자세한 정보는 *Oracle Cloud 시작하기*에서 사용자 계정 배치 임포트를 참조하십시오.
 - HSS-Shared Services\resource\External Directory\Roles\에서 새로운 사용자의 역할 지정에 대한 정보(이름,성,전자메일,사용자 로그인 형식)를 해당 역할 파일에 추가. 예를 들어 서비스 관리자 역할에 대한 지정은 <service_name> Service Administrator.csv에 추가해야 하는 반면 뷰어 역할에 대한 지정은 HSS-Shared Services\resource\External Directory\Roles\<service_name> Viewer.csv에 추가해야 합니다. 자세한 정보는 *Oracle Cloud 시작하기*에서 하나의 역할을 여러 사용자에게 지정을 참조하십시오.
 - HSS-Shared Services 디렉토리 및 해당 콘텐츠를 압축하여 업데이트된 파일로 스냅샷 재생성.
- ID 도메인 관리자 역할도 있는 서비스 관리자로 환경에 로그인합니다.
- 수정한 example_snapshot.zip을 환경에 업로드합니다.
- example_snapshot.zip을 환경에 임포트합니다.
- 업로드한 example_snapshot.zip을 환경에서 삭제합니다.
- 사인아웃

Windows 샘플 스크립트

다음 스크립트를 복사하여 createUsersAndAssignRoles.ps1이라는 파일을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -raw)
$username="$($inputproperties.username)"
$passwordfile="$($inputproperties.passwordfile)"
$serviceURL="$($inputproperties.serviceURL)"
$snapshotName="$($inputproperties.snapshotName)"
$userspassword="$($inputproperties.userspassword)"
$resetPassword="$($inputproperties.resetPassword)"

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate exportsnapshot ${snapshotName}
epmautomate downloadfile ${snapshotName}.zip
epmautomate deletefile ${snapshotName}.zip
epmautomate logout

# Add custom code to extract the contents of example_snapshot.zip
# Add custom code to append new user information to HSS-Shared
Services\resource\External Directory\Users.csv
# Optional: Add custom code to append role information to the appropriate
role file(s) in HSS-Shared Services\resource\External Directory\Roles\
# Add custom code to zip HSS-Shared Services and its contents as
example_snapshot.zip

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile ${snapshotName}.zip
epmautomate importsnapshot ${snapshotName} userPassword=${userspassword}
resetPassword=${resetPassword}
epmautomate deletefile ${snapshotName}.zip
epmautomate logout
```

Linux/UNIX 샘플 스크립트

다음 스크립트를 복사하여 createUsersAndAssignRoles.sh라는 파일을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
#!/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

```

샘플 input.properties 파일

createUsersAndAssignRoles 스크립트를 실행하려면 input.properties 파일을 생성하고 해당 환경에 대한 정보로 업데이트합니다. createUsersAndAssignRoles.ps1 또는 createUsersAndAssignRoles.sh가 저장된 디렉토리에 파일을 저장합니다.

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

```

표 3-2 input.properties 매개변수

매개변수	설명
javahome	JAVA_HOME 위치입니다. Linux/UNIX에만 해당합니다.
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다. Linux/UNIX에만 해당합니다.
username	ID 도메인 관리자 역할도 가진 서비스 관리자의 사용자 이름입니다.
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
serviceURL	스냅샷을 생성하려는 소스 환경의 URL입니다.
snapshotName	생성하려는 스냅샷의 이름입니다. 이 스냅샷을 생성하기 위해 마이그레이션을 사용하여 이전에 그룹 및 멤버십 아티팩트를 익스포트했다고 가정합니다.
userspassword	새 사용자의 초기 비밀번호입니다.
resetPassword	새 사용자가 첫번째 로그인 시 비밀번호를 재설정해야 하는지 여부입니다. 새 사용자가 처음 로그인할 때 해당 비밀번호를 강제로 변경하게 하려면 이 값을 true로 설정합니다.

스크립트 실행

- 이전 섹션에서 스크립트를 복사하여 `createUsersAndAssignRoles.ps1` 또는 `createUsersAndAssignRoles.sh`를 생성합니다.
- 사용자정의 코드를 추가하여 다음 작업을 수행합니다.
 - 스냅샷의 콘텐츠를 추출합니다.
 - HSS-Shared Services\resource\External Directory\Users.csv에 새 사용자 정보를 추가합니다.
 - 필요에 따라 HSS-Shared Services\resource\External Directory\Roles\에서 새로운 사용자의 역할 지정에 대한 정보(이름,성,전자메일,사용자 로그인 형식)를 해당 역할 파일에 추가합니다.
 - 업데이트된 파일로 스냅샷을 재생성합니다.
- `input.properties` 파일을 생성하고 `createUsersAndAssignRoles` 스크립트가 있는 디렉토리에 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. [샘플 input.properties 파일](#)을 참조하십시오.
이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 **관리자 권한으로 실행** 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
- 스크립트를 실행합니다.
 - Windows PowerShell:** `createUsersAndAssignRoles.ps1`을 실행합니다.
 - Linux/UNIX:** `./createUsersAndAssignRoles.sh`를 실행합니다.

라이선스를 가진 사용자(역할에 지정된 사용자) 수 계산

이 섹션의 스크립트를 사용하여 환경의 사용자 수를 계산하는 역할 지정 보고서를 생성할 수 있습니다.

다음 스크립트를 복사하여 `provisionedUsersCount.bat`를 생성합니다.

① 주

- `provisionedUsersCount.bat`를 실행하기 위한 입력 매개변수는 `username`, `password/password_file`, `service_url`, `report_file_name`입니다. 예를 들어 명령 프롬프트에서 다음과 유사한 명령을 입력합니다.
`provisionedUsersCount.bat jdoe password.epw https://example.oraclecloud.com myRole_assign.CSV`
- 비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오.

```
@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 "%4"
if %errorlevel% NEQ 0 exit /b 1
    call epmautomate.bat downloadFile "%4"
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%.
)

```

역할에 지정된 사용자 감사 보고서 생성

이 섹션의 스크립트를 사용하여 환경의 사전 정의된 역할에 지정된 사용자에 대한 감사 보고서 생성 프로세스를 자동화할 수 있으며 선택적으로 수신자에게 전자메일로 보낼 수 있습니다.

이 감사 보고서는 보고서가 마지막으로 생성된 이후 변경한, 사전 정의된 역할 또는 그룹에 지정된 사용자를 보여 줍니다. 일별 감사 보고서를 생성하려면 이 스크립트를 매일 실행하십시오.

다음 스크립트를 복사하여 provisioningAuditReport.bat를 생성합니다. 이 래퍼 배치 스크립트에서는 이 시나리오에서 나중에 제공되는 소스 코드인 PowerShell 스크립트 provisioningAuditReport.ps1을 호출합니다.

① 주

- provisioningAuditReport.bat 실행에 사용되는 입력 매개변수는 username, password 또는 password_file, service_url, report_email_to_address(선택사항으로, 보고서를 전자메일 주소로 보내려는 경우에만 필요함)입니다.
- 비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오.

```

@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.bat에서는 다음 스크립트를 복사하여 생성하는 provisioningAuditReport.ps1을 호출합니다.

provisioningAuditReport.ps1은 감사 보고서를 생성합니다. provisioningAuditReport.bat가 있는 디렉토리에 두십시오.

```

$username=$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-audittest-" + $date + ".csv"
$provisionreporttemp="./provreport-audittest-temp.csv"
$provisionreportunique="./provreport-audittest-unique.csv"
$provisionreportbaselineunique="./provreport-audittest-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"

```

역할 지정 및 취소 감사 보고서 생성

이 섹션의 PowerShell 스크립트를 사용하여 환경의 역할 지정 및 역할 취소를 자세히 설명하는 감사 보고서를 생성하는 프로세스를 자동화할 수 있습니다.

다음 스크립트를 복사하여 AuditReportRoleAssignment.bat를 생성합니다. 이 래퍼 배치 스크립트는 이 시나리오에서 나중에 제공되는 소스 코드인 PowerShell 스크립트 AuditReportRoleAssignment.ps1을 호출합니다.

① 주

- AuditReportRoleAssignment.bat를 실행하기 위한 입력 매개변수는 username, password 또는 password_file, service_url입니다.
- 비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오.

스크립트: 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 %*
```

스크립트: 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-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 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"

```

개인 정보 보호 법률을 준수하도록 액세스 로그 및 활동 보고서 마스킹

이 섹션의 스크립트를 사용하여 개인정보보호 법률을 준수하도록 활동 보고서 또는 액세스 로그의 정보를 마스킹하는 프로세스를 자동화할 수 있으며 선택적으로 수신자에게 전자메일로 보낼 수 있습니다.

일부 국가의 엄격한 개인 정보 보호 법률 때문에 사용자의 개인 정보 보호를 위해 활동 보고서 및 액세스 로그에 제공되는 정보가 서비스 관리자에게 표시되지 않도록 숨겨야 할 수도 있습니다.

anonymizeData.bat를 사용하여 개인정보보호 법률을 준수하도록 활동 보고서 또는 액세스 로그의 정보를 마스킹하고 선택적으로 전자메일로 보낼 수 있습니다. 정보를 마스킹하려면 매일 각 환경에 대한 일별 유지관리 프로세스가 완료된 후 바로 실행되도록 Windows 스케줄러를 사용하여 이 스크립트나 스크립트의 변형을 스케줄링합니다.

다음과 같은 정보 소스를 사용합니다.

- *관리자용 시작 가이드*의 활동 보고서 및 액세스 로그로 사용량 모니터

- [스크립트 실행 자동화](#)

다음 절차에 제공된 Windows 스크립트를 복사하여 anonymizeData.bat를 수동으로 생성하고 Windows 스케줄러를 사용하여 스케줄링합니다. Windows를 사용하여 스케줄링하지 않는 경우 플랫폼에 적합한 유사 스크립트를 생성하고 실행할 수도 있습니다.

anonymizeData.bat는 다음 절차에 설명된 대로 생성하고 업데이트하는 anonymizeData.ps1 스크립트를 실행하는 래퍼 스크립트입니다.

사용하는 비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오.

1. 다음 스크립트를 포함하는 anonymizeData.bat라는 배치(BAT) 파일을 생성한 후 C:\automate_scripts 등의 편리한 위치에 저장합니다.

```
@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. 다음 스크립트를 포함하는 anonymizeData.ps1이라는 PowerShell 스크립트(PS1) 파일을 생성한 후 C:\automate_scripts 등의 편리한 위치에 저장합니다.

```
# 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
{
    $aprdir=$args[0]
    $datestampdir=$args[1]
    $path="$aprdir/$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('/')
        $aprrdir=$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 $aprdir/$datestampdir
were processed earlier. Skipping these files."
    }
}

function callSendMail
{
    $elements=$logfile.split('/')
    $logfile=$elements[3]

    if (${emailtoaddress} -match "@") {
        epmautomate.bat login ${username} ${password} ${url}
        epmautomate.bat uploadFile "$logfile"
        epmautomate.bat sendMail $emailtoaddress "Mask Access Logs and
Activity Reports results" Body="The results of running the Mask Access
Logs and Activity Reports script are attached." Attachments=$logfile
        epmautomate.bat deleteFile "$logfile"
        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. Windows 스케줄러를 사용하여 anonymizeData.bat를 스케줄링합니다. 자세한 단계는 [스크립트 실행 자동화](#)를 참조하십시오.

anonymizeData.bat를 실행하려면 다음 매개변수 값을 제공해야 합니다.

- 서비스 관리자의 사용자 이름
- 서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일을 사용할 수 있는 위치
- 액세스 로그 및 활동 보고서를 마스킹할 서비스 환경의 URL
- **선택사항:** 보고서를 보낼 전자메일 주소. 이 값이 지정된 경우에만 보고서를 전자메일로 보냅니다.

로컬 컴퓨터로 활동 보고서 다운로드 자동화

이 섹션의 스크립트를 사용하여 환경의 활동 보고서를 로컬 컴퓨터로 다운로드하는 프로세스를 자동화할 수 있습니다.

syncAprReports.bat를 사용하여 활동 보고서를 다운로드합니다. Windows 스케줄러를 통해 배치 파일을 스케줄하여 활동 보고서 다운로드를 자동화할 수 있습니다. 활동 보고서에 대한 자세한 내용은 *관리자용 시작 가이드*에서 활동 보고서 및 액세스 로그로 사용량 모니터를 참조하십시오.

다음 절차에 제공된 스크립트를 복사한 다음 연결 매개변수를 업데이트하여 수동으로 syncAprReports.bat를 생성합니다. 이 스크립트는 환경을 확인하고 로컬 컴퓨터의 다운로드 디렉토리에서 사용할 수 있는 보고서보다 최신 버전인 보고서만 다운로드합니다.

① 주

- Windows 컴퓨터에서만 스크립트를 실행해야 합니다.
- 이 스크립트는 사용자가 피드백을 제출할 때 생성되는 피드백 활동 보고서를 다운로드하지 않습니다.
- 사용하는 비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오.

1. 다음 스크립트를 포함하는 syncAprReports.bat라는 बै치(.BAT) 파일을 생성한 후 C:\automate_scripts 등의 편리한 위치에 저장합니다.

```
@echo off
title APR
setlocal DisableDelayedExpansion

REM To hardcode the values in the script replace %1, %2, %3, and %4, with
the actual values.
REM Example:
REM set apr_dir="C:\Oracle\apr"
REM set username="serviceAdmin"
REM set password="Ex@mp!e!"
REM set url="https://test-example.stg-pbcs.us1.oraclecloud.com"
set apr_dir=%1
set username=%2
set password=%3
set url=%4
setlocal EnableDelayedExpansion
set epmautomate_dir=%cd%
set lastfile=
set argC=0
for %%x in (%*) do Set /A argC+=1
if %argC% neq 0 (
    if %argC% neq 3 (
        if %argC% neq 4 (
            goto :usage
        )
    )
)
goto :login
:usage
echo.
echo Invalid syntax. Please check the parameters.
echo Syntax:
echo 1) syncAprReports.bat APR_FolderPath_on_client username password url
echo      or
echo 2) set the parameters in the file and use below syntax
echo      syncAprReports.bat
goto :end

:login
setlocal DisableDelayedExpansion
for /f "delims=" %%i in ('epmautomate login %username% %password% %url%')
do set result=%%i
if "Login successful" neq "%result%" (
```

```

        echo Login Failed
        goto :end
    )

    if not exist %apr_dir% (
        echo.
        echo apr folder does not exist
        GOTO :end
    )
    cd /D %apr_dir%
    for /f "delims=" %%D in ('dir /a:d /b /o:-n') do (
    REM AFTER: for /f "delims=" %%D in ('dir /a-d /b /s /o:-n') do (
        set "lastFile=%%~nD"
        goto :next
    )

    :next
    setlocal EnableDelayedExpansion
    echo.
    echo Most Recent APR on client is %lastFile%

    set "output_cnt=0"
    cd /D %epmautomate_dir%
    for /F "delims=" %%f in ('epmautomate listfiles') do (

        cd /D !apr_dir!
        set "line=%%f"
        for /f "tokens=* delims=" %%a in ("!line!") do set line=%%a
        if "!line:~0,3!" equ "apr" (

            if "!line:~4,8!" neq "Feedback" (

                set isValidFile=false
                if "!line:~5!" equ ".html" set isValidFile=true
                if "!line:~5!" equ ".json" set isValidFile=true

                if "!isValidFile!" equ "true" (

                    if "%lastFile%" lss "!line:~4,19!" (

                        if "!line:~4,19!" neq "!dirname!" (

                            set apr_dir=!apr_dir:!="
                            set /a output_cnt+=1
                            set "output[!output_cnt!]=!apr_dir!\!
line:~4,19!"

                            set "dirname=!line:~4,19!"

                            REM start downloading
                            mkdir "!dirname!"
                            cd /D !dirname!
                            echo downloading !line!
                            set "downloadDir=!apr_dir!\!dirname!"

                            cd /D %epmautomate_dir%
                            for /f "delims=" %%i in ('epmautomate

```

```

downloadfile "!line!"') do set result1=%%i
                        move "!line:~24!" "!downloadDir!" > nul
                        echo !result1!
                        REM end downloading

                    ) else (

                        REM start downloading
                        cd /D !dirname!
                        echo downloading !line!
                        set apr_dir=!apr_dir:"=!
                        set "downloadDir=!apr_dir!\!dirname!"
                        cd /D %epmautomate_dir%
                        for /f "delims=" %%i in ('epmautomate
downloadfile "!line!"') do set result1=%%i
                        move "!line:~24!" "!downloadDir!" > nul
                        echo !result1!
                        REM end downloading

                    )
                ) else (

                    REM TO-DO

                )
            )
        )
    )
)

echo.
echo %output_cnt% APR's downloaded
for /L %%n in (1 1 !output_cnt!) DO echo !output[%%n]!
goto :end

:end
cd /D %epmautomate_dir%
endlocal

```

2. syncAprReports.bat를 수정하여 다음 표의 매개변수 값을 설정합니다. 이러한 값은 환경에 액세스하여 활동 보고서를 다운로드하는 데 사용됩니다.

표 3-3 syncAprReports.bat에 포함할 매개변수 값

매개변수	예상 값
set apr_dir=%1	활동 보고서를 다운로드할 기존 디렉토리를 지정합니다. 예: set apr_dir="C:\Oracle\apr"
set username=%2	환경에 로그인하여 활동 보고서를 다운로드하는 데 사용되는 Oracle Fusion Cloud Enterprise Performance Management 사용자 이름입니다. 예: set username="ServiceAdmin"

표 3-3 (계속) syncAprReports.bat에 포함할 매개변수 값

매개변수	예상 값
set password=%3	username 변수로 지정된 사용자의 암호화된 비밀번호를 저장하는 파일의 이름 및 위치입니다. 사용자의 일반 텍스트 비밀번호를 지정할 수도 있습니다 (권장되지 않음). 암호화된 비밀번호 파일을 생성하는 방법에 대한 자세한 내용은 encrypt 명령을 참조하십시오. 예: set password="C:\mySecuredir\password.epw" set password="Ex@mpl1"
set url=%4	환경의 URL입니다. 예: set url="https://test-example.stg-pbcs.us1.oraclecloud.com"

- Windows 스케줄러를 사용하여 syncAprReports.bat를 스케줄합니다. 자세한 단계는 [스크립트 실행 자동화](#)를 참조하십시오.

환경에서 액세스 로그 다운로드

이 섹션의 스크립트를 사용하여 환경의 액세스 로그를 로컬 컴퓨터로 다운로드하는 프로세스를 자동화할 수 있습니다.

Windows 스케줄러를 통해 syncAccessLog.bat를 스케줄링하여 로그 파일 다운로드를 자동화할 수 있습니다. 애플리케이션 관리를 사용하여 액세스 로그를 다운로드하는 절차는 [관리/사용 시작 가이드](#)에서 활동 보고서 및 액세스 로그 보기 및 다운로드를 참조하십시오.

다음 스크립트는 환경을 확인하고 로컬 컴퓨터의 다운로드 디렉토리에서 사용할 수 있는 로그 파일보다 최신 버전인 로그 파일만 다운로드합니다. Windows 스크립트입니다. Linux/UNIX 환경용으로 유사한 쉘 스크립트를 생성할 수 있습니다.

- 다음 스크립트를 포함하는 syncAccessLog.bat라는 배치(.BAT) 파일을 생성한 후 C:\automate_scripts 등의 편리한 위치에 저장합니다.

① 주

비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오.

```
@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. syncAccessLog.bat를 수정하여 다음 표의 매개변수 값을 설정합니다. 이러한 값은 환경에 액세스하여 액세스 로그를 다운로드하는 데 사용됩니다.

표 3-4 syncAccessLog.bat에 포함할 변수 값

변수	예상 값
set apr_dir=%1	액세스 로그를 다운로드할 디렉토리를 지정합니다. 예: set apr_dir="C:\Oracle\apr"
set username=%2	환경에 로그인하여 액세스 로그를 다운로드하는 데 사용되는 Oracle Fusion Cloud Enterprise Performance Management 사용자 이름입니다. 예: set username="ServiceAdmin"
set password=%3	username 변수로 지정된 사용자의 암호화된 비밀번호를 저장하는 파일의 이름 및 위치입니다. 사용자의 일반 텍스트 비밀번호를 지정할 수도 있습니다 (권장되지 않음). 암호화된 비밀번호 파일을 생성하는 방법에 대한 자세한 내용은 encrypt 명령을 참조하십시오. 예: set password="C:\mySecuredir\password.epw" set password="P@ssword1"
set url=%4	환경의 URL입니다. 예: set url="https://test-cloudpln.pbcs.us1.oraclecloud.com"

3. Windows 스케줄러를 사용하여 syncAccessLog.bat를 스케줄링합니다. 자세한 단계는 [스크립트 실행 자동화](#)를 참조하십시오.

환경 복제 자동화

이 섹션의 스크립트를 사용하여 환경 복제를 자동화할 수 있습니다.

아래와 유사한 스크립트가 포함된 배치(.bat) 또는 셸(.sh) 파일을 생성하여 환경을 복제합니다. 아래 샘플 스크립트에서는 다음 활동을 처리합니다:

- 소스 환경에 로그인합니다.
- 소스 환경에서 사용할 수 있는 Artifact Snapshot(소스 환경의 최종 일별 유지관리 중 생성된 스냅샷) 또는 다른 스냅샷으로 타겟 환경을 소스의 복제본으로 변환합니다.
- 선택적으로, 소스 환경과 일치하도록 사용자 및 사전 정의된 애플리케이션 역할 지정을 생성합니다.
- 선택적으로, 일별 유지관리 시작 시간을 소스 환경의 해당 시간과 일치하도록 변경합니다.
- 사인아웃

복제 프로세스에 대한 자세한 내용은 *마이그레이션 관리*에서 "[Cloud EPM 환경 복제](#)"를 참조하십시오.

Windows 태스크 스케줄러를 사용하여 스크립트 스케줄을 예약하는 방법에 대한 자세한 내용은 [스크립트 실행 자동화](#)를 참조하십시오.

Windows

1. 다음 스크립트를 포함하는 cloneEnvironment.bat라는 배치(.BAT) 파일을 생성한 후 C:\automate_scripts 등의 편리한 위치에 저장합니다.

```
@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%

```

- cloneEnvironment.bat를 수정하여 다음 매개변수 값을 설정합니다.

표 3-5 cloneEnvironment.bat에서 설정할 매개변수

매개변수	설명
usersandpredefinedroles	사용자 및 사전 정의된 애플리케이션 역할 지정을 복제하려면 이 매개변수 값을 true로 설정합니다. 사용자 및 역할 지정을 복제하려면 스크립트를 실행하는 사용자가 타겟 환경에서 서비스 관리자 및 ID 도메인 관리자 역할이 있어야 합니다.
snapshotname	복제에 사용할 소스 환경의 스냅샷 이름입니다.

표 3-5 (계속) cloneEnvironment.bat에서 설정할 매개변수

매개변수	설명
dailymaintenancestarttime	복제된 환경의 일별 유지관리 시작 시간을 소스 환경의 해당 시간으로 변경하려면 이 매개변수 값을 true로 설정합니다. 복제된 환경의 현재 일별 유지관리 시작 시간을 유지하려면 이 값을 false로 설정합니다.

3. 다음 스크립트가 포함된 PowerShell 스크립트 cloneEnvironment.ps1을 생성하여 cloneEnvironment.bat를 저장한 디렉토리(예: C:\automate_scripts)에 저장합니다.

```
# 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. 다음 명령을 사용하여 cloneEnvironment.bat를 실행합니다.

```
cloneEnvironment.bat "SOURCE USERNAME" "SOURCE PASSWORD FILE" "SOURCE URL" "TARGET USERNAME" "TARGET PASSWORD FILE" "TARGET URL"
```

예를 들어 다음과 같습니다.

```
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. 다음 스크립트가 포함된 셸 스크립트 cloneEnvironment.sh를 생성하여 편리한 위치에 저장합니다.

```
#!/bin/bash

# Update the following parameters
# -----
epmautomatescript=/home/user1/epmautomate/bin/epmautomate.sh
javahome=/home/user1/jdk_17.0.12/
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
    
```

- cloneEnvironment.sh를 수정하여 다음 매개변수 값을 설정합니다.

표 3-6 cloneEnvironment.sh에서 설정할 매개변수

매개변수	설명
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다.
javahome	JAVA_HOME 위치입니다.
usersandpredefinedroles	사용자 및 사전 정의된 애플리케이션 역할 지정을 복제하려면 이 매개변수 값을 true로 설정합니다. 사용자 및 역할 지정을 복제하려면 스크립트를 실행하는 사용자가 타겟 환경에서 서비스 관리자 및 ID 도메인 관리자 역할이 있어야 합니다.
snapshotname	복제에 사용할 소스 환경의 스냅샷 이름입니다.
dailymaintenancestarttime	복제된 환경의 일별 유지관리 시작 시간을 소스 환경의 해당 시간으로 변경하려면 이 매개변수 값을 true로 설정합니다. 복제된 환경의 현재 일별 유지관리 시작 시간을 유지하려면 이 값을 false로 설정합니다.

- cloneEnvironment.sh를 실행합니다.

```

./cloneEnvironment.sh "SOURCE USERNAME" "SOURCE PASSWORD FILE" "SOURCE
URL" "TARGET USERNAME" "TARGET PASSWORD FILE" "TARGET URL"
    
```

예를 들어 다음과 같습니다.

```
./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.
```

기본 환경에서 일별 유지관리가 완료된 후 매일 기본 환경에서 대기 환경으로 복제

대기 환경을 기본 환경과 같은 최신 상태로 유지하려면 기본 환경에서 일별 유지관리가 완료된 직후 이 스크립트를 사용하여 Oracle Fusion Cloud Enterprise Performance Management 기본 환경을 대기 환경으로 복제합니다.

이러한 사용자정의 스크립트는 해당 날짜에 스케줄링된 일별 유지관리가 완료되었는지 확인한 후 환경을 복제합니다.

Windows 스크립트

다음 PowerShell 스크립트를 복사하여 `dailyclone.ps1`을 생성합니다.

```
# Clone Environment script
#
# Update the following parameters
# -----
$users_and_predefined_roles="false"
$daily_maintenance_start_time="true"
$data_management="true"
$app_audit="true"
$job_console="true"
$stored_snapshots_and_files="false"
# -----

$source_username=$args[0]
$source_password=$args[1]
$source_url=$args[2]
$target_username=$args[3]
$target_password=$args[4]
$target_url=$args[5]

if ($($args.count) -ne 6) {
    echo "Usage: ./dailyclone.ps1 <SOURCE USERNAME> <SOURCE PASSWORD> <SOURCE
URL> <TARGET USERNAME> <TARGET PASSWORD> <TARGET 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 deleteArtifactSnapshotIfExists {
    if (artifactSnapshotExists) {
        $command_del=$(epmautomate.bat deletefile "Artifact Snapshot")
    }
}

function artifactSnapshotExists {
    $filelist=$(epmautomate.bat listfiles)
    if ("$filelist".contains("Artifact Snapshot")) {
        return $true
    } else {
        return $false
    }
}

function cloneEnvironment {
    echo "Checking to see if daily maintenance processing has completed ..."
    while ($true) {
        if (artifactSnapshotExists) {
            echo "Daily maintenance processing has completed ..."
            break
        } else {
            echo "Sleeping for 30 seconds before the next check to see if
daily maintenance processing has completed ..."
            Start-Sleep -Seconds 30
        }
    }

    echo "Encrypting target password ..."
    epmautomate.bat encrypt "${target_password}" "oracleKey"
    "target_password.epw"

    echo "Cloning environment ..."
    epmautomate.bat cloneEnvironment "${target_username}"
    "target_password.epw" "${target_url}" "SnapshotName=Artifact Snapshot"
}

```

```

"UsersAndPreDefinedRoles=${users_and_predefined_roles}" "DataManagement=${
{data_management}" "appAudit=${app_audit}" "jobConsole=${job_console}"
"storedSnapshotsAndFiles=${stored_snapshots_and_files}"
"dailyMaintenanceStartTime=${daily_maintenance_start_time}"
}

echo "Beginning clone environment script."
echo "Logging into server ..."
epmautomate.bat login ${source_username} ${source_password} ${source_url}
$amwtime=getDailyMaintenanceStartTime
goToSleep ($amwtime)
deleteArtifactSnapshotIfExists
cloneEnvironment
echo "Logging out of server ..."
epmautomate.bat logout
echo "Clone environment script processing has completed."

```

Linux/UNIX 스크립트

다음 스크립트를 복사하여 `dailyclone.sh`를 생성합니다.

```

#!/bin/bash

# Update the following parameters
# -----
epmautomatescript="LOCATION_EPM_AUTOMATE_EXECUTABLE"
javahome="LOCATION_JAVA_HOME"
users_and_predefined_roles="false"
data_management="true"
app_audit="true"
job_console="true"
stored_snapshots_and_files="false"
daily_maintenance_start_time="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: ./dailyclone.sh SOURCE_USERNAME SOURCE_PASSWORD SOURCE_URL
TARGET_USERNAME TARGET_PASSWORD TARGET_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
}

deleteArtifactSnapshotIfExists() {
    found=1
    filelist=$((${epmautomatescript} listfiles)
    if [[ ${filelist} == *"Artifact Snapshot"* ]]
    then
        command_del=$((${epmautomatescript} deletefile "Artifact Snapshot")
    fi
}

artifactSnapshotExists() {
    found=1

    filelist=$((${epmautomatescript} listfiles)
    if [[ ${filelist} == *"Artifact Snapshot"* ]]
    then
        found=0
    else
        found=1
    fi

    echo ${found}
}

cloneEnvironment() {
    local found=1

    while true
    do
        found=$(artifactSnapshotExists)
        if [[ ${found} -eq 0 ]]
        then
            echo "Daily maintenance processing has completed ..."
            break
        else

```

```

        echo "Sleeping for 30 seconds before the next check to see if
daily maintenance processing has completed ..."
        sleep 30
    fi
done

    echo "Encrypting target password ..."
    ${epmautomatescript} encrypt "${target_password}" "oracleKey"
"target_password.epw"

    echo "Cloning environment ..."
    ${epmautomatescript} cloneEnvironment "${target_username}"
"target_password.epw" "${target_url}" "SnapshotName=Artifact Snapshot"
"UsersAndPreDefinedRoles=${users_and_predefined_roles}" "DataManagement=${
{data_management}" "appAudit=${app_audit}" "jobConsole=${job_console}"
"storedSnapshotsAndFiles=${stored_snapshots_and_files}"
"dailyMaintenanceStartTime=${daily_maintenance_start_time}"
}

echo "Beginning clone environment script."
echo "Logging into server ..."
${epmautomatescript} login ${source_username} ${source_password} ${source_url}
getDailyMaintenanceStartTime
goToSleep
deleteArtifactSnapshotIfExists
cloneEnvironment
echo "Logging out of server ..."
${epmautomatescript} logout
echo "Clone environment script processing has completed."

```

Groovy 스크립트

다음 코드를 복사하여 dailyclone Groovy 스크립트를 생성합니다.

```

// Clone Environment script

// Update the following parameters
String source_username="USERNAME"
String source_password="PASSWORD!"
String source_url="URL OF THE SOURCE ENVIRONMENT"
String target_username="USERNAME"
String target_password="PASSWORD"
String target_url="URL OF THE TARGET ENVIRONMENT"
String snapshot_name="Artifact Snapshot"

// -----

// Do not update the following parameters
String users_and_predefined_roles="false"
String data_management="true"
String app_audit="true"
String job_console="true"
String stored_snapshots_and_files="false"
String daily_maintenance_start_time="true"
// -----

```

```

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)
}

boolean artifactSnapshotExists(EpmAutomate automate,String snapshot_name) {
    LogMessage("Operation: listfiles")
    EpmAutomateStatus listfilesstatus = automate.execute('listfiles')
    String filelist=(listfilesstatus.getItemsList())
    LogOperationStatus(listfilesstatus)

    if (filelist.contains(snapshot_name)) {
        return true
    }
}

```

```

    } else {
        return false
    }
}

def deleteArtifactSnapshotIfExists(EpmAutomate automate,String snapshot_name){
    if (artifactSnapshotExists(automate,snapshot_name)) {
        LogMessage("Operation: deletefile " + snapshot_name)
        EpmAutomateStatus deletefilestatus =
automate.execute('deletefile',snapshot_name)
        LogOperationStatus(deletefilestatus)
    }
}

def waitForDailyMaintenanceToComplete(EpmAutomate automate,String
snapshot_name) {
    LogMessage("Checking to see if daily maintenance processing has
completed ...")
    while (true) {
        if (artifactSnapshotExists(automate,snapshot_name)) {
            LogMessage("Daily maintenance processing has completed ...")
            break
        } else {
            LogMessage("Sleeping for 30 seconds before the next check to see
if daily maintenance processing has completed ...")
            sleep(30000)
        }
    }
}

LogMessage("Clone environment script processing beginning ...")

EpmAutomate automate = getEpmAutomate()

LogMessage("Operation: login " + source_username + " " + source_password + "
" + source_url)
EpmAutomateStatus status =
automate.execute('login',source_username,source_password,source_url)
LogOperationStatus(status)

String amwtime = getDailyMaintenanceStartTime(automate)
goToSleep (amwtime)
deleteArtifactSnapshotIfExists(automate,snapshot_name)
waitForDailyMaintenanceToComplete(automate,snapshot_name)

LogMessage("Operation: encrypt " + target_password + " oracleKey
target_password.epw")
status =
automate.execute('encrypt',target_password,'oracleKey','target_password.epw')
LogOperationStatus(status)

LogMessage("Operation: cloneEnvironment " + target_username + "
target_password.epw " + target_url + " SnapshotName=" + snapshot_name + "
UsersAndPreDefinedRoles=" + users_and_predefined_roles + " DataManagement=" +
data_management + " appAudit=" + app_audit + " jobConsole=" + job_console + "
storedSnapshotsAndFiles=" + stored_snapshots_and_files + "

```

```

dailyMaintenanceStartTime=" + daily_maintenance_start_time)
EpmAutomateStatus cloneenvironmentstatus =
automate.execute('cloneEnvironment',target_username,"target_password.epw",target_url,"Snap
+ snapshot_name,"UsersAndPreDefinedRoles=" +
users_and_predefined_roles,"DataManagement=" + data_management,"appAudit=" +
app_audit,"jobConsole=" + job_console,"storedSnapshotsAndFiles=" +
stored_snapshots_and_files,"dailyMaintenanceStartTime=" +
daily_maintenance_start_time)
LogOperationStatus(cloneenvironmentstatus)

LogMessage("Operation: logout ")
status = automate.execute('logout')
LogOperationStatus(status)

LogMessage ("Clone environment script processing has completed.")

```

스크립트 실행

- 이전 섹션 중 하나에서 스크립트를 복사하여 `dailyclone.ps1`, `dailyclone.sh` 또는 `dailyclone Groovy` 스크립트를 생성합니다.

- 스크립트를 업데이트합니다.

`dailyclone.sh`에 대한 업데이트:

- `epmautomatescript`는 EPM Automate 실행파일의 위치로 업데이트합니다. 예:
`epmautomatescript="/home/utils/EPMAutomate/bin/epmautomate.sh"`
- `javahome`은 EPM Automate에서 사용되는 JDK가 설치된 디렉토리로 업데이트합니다. 예:
`"/home/user1/jdk_17.0.12"`

`dailyclone.ps1` 및 `dailyclone.sh`에 대한 업데이트:

- `users_and_predefined_roles`: 이 값을 `true`로 설정하면 사용자와 해당 사용자의 사전 정의된 역할 지정이 복제됩니다(액세스 제어 그룹은 항상 복제됨).
- `data_management`: Data Integration 레코드를 복제하지 않으려면 이 값을 `false`로 설정합니다. Data Integration 레코드는 소스 환경과 타겟 환경 둘 다 월별 업데이트 버전이 동일하거나 타겟 환경이 소스 환경보다 업데이트 버전이 하나 더 최신인 경우에만 복제할 수 있습니다. 예를 들어 22.01 Data Management 레코드는 다른 22.01 환경 또는 22.02 환경에만 복제할 수 있습니다. Narrative Reporting 및 Oracle Fusion Cloud Enterprise Data Management 환경에서는 무시됩니다.
- `app_audit`: Planning, FreeForm 및 Enterprise Profitability and Cost Management 애플리케이션에 대한 애플리케이션 감사 데이터를 복제하지 않으려면 이 값을 `false`로 설정합니다. Financial Consolidation and Close 및 Tax Reporting 감사 정보는 항상 복제됩니다.
- `job_console`: 작업 콘솔 데이터를 복제하지 않으려면 이 값을 `false`로 설정합니다.
- `stored_snapshots_and_files`: 소스 환경의 인박스 및 아웃박스에 있는 최상위 폴더의 콘텐츠를 복제하려면 이 값을 `true`로 설정합니다(하위폴더는 복제되지 않음).
- `daily_maintenance_start_time`: 복제된 타겟 환경의 유지관리 시작 시간을 소스 환경의 유지관리 시작 시간으로 재설정하지 않으려면 이 값을 `false`로 설정합니다.

`dailyclone Groovy` 스크립트에 대한 업데이트:

- `source_username`은 서비스 관리자의 사용자 이름입니다. 사용자 및 사전 정의된 역할을 복제하려면 ID 도메인 관리자 역할이 있어야 합니다.
- `source_password`는 `SOURCE_USERNAME`에서 확인된 사용자의 비밀번호입니다.

- `source_url`은 복제할 환경의 URL입니다.
 - `target_username`은 서비스 관리자의 사용자 이름입니다. 사용자 및 사전 정의된 역할을 복제하려면 ID 도메인 관리자 역할이 있어야 합니다.
 - `target_password`는 `TARGET_USERNAME`에서 확인된 사용자의 비밀번호입니다.
 - `target_url`은 타겟 환경의 URL입니다.
3. `dailyclone.ps1`, `dailyclone.sh` 또는 `dailyclone Groovy` 스크립트를 실행합니다.

Windows

- 명령 창에서 `dailyclone.ps1`이 저장된 폴더로 이동합니다.
- `./dailyclone.ps1 SOURCE_USERNAME SOURCE_PASSWORD SOURCE_URL TARGET_USERNAME TARGET_PASSWORD TARGET_URL` 명령을 실행합니다. 이러한 값에 대한 설명은 다음 테이블을 참조하십시오.

Table 3-7 매개변수 설명

매개변수	설명
<code>SOURCE_USERNAME</code>	서비스 관리자의 사용자 이름입니다. 또한 이 사용자는 사용자 및 사전 정의된 역할을 복제하려면 ID 도메인 관리자 역할이 있어야 합니다.
<code>SOURCE_PASSWORD</code>	<code>SOURCE_USERNAME</code> 에서 확인된 사용자의 비밀번호입니다.
<code>SOURCE_URL</code>	복제할 환경의 URL입니다.
<code>TARGET_USERNAME</code>	타겟 환경에서 서비스 관리자의 사용자 이름입니다. 또한 이 사용자는 사용자 및 사전 정의된 역할을 복제하려면 ID 도메인 관리자 역할이 있어야 합니다.
<code>TARGET_PASSWORD</code>	<code>TARGET_USERNAME</code> 에서 확인된 사용자의 비밀번호입니다.
<code>TARGET_URL</code>	타겟 환경의 URL입니다.

Linux/UNIX

- UNIX 또는 Linux 셸에서 `dailyclone.sh`가 저장된 디렉토리로 이동합니다.
- `./dailyclone.sh SOURCE_USERNAME SOURCE_PASSWORD SOURCE_URL TARGET_USERNAME TARGET_PASSWORD TARGET_URL` 명령을 실행합니다. 이러한 값에 대한 설명은 위 테이블을 참조하십시오.

Groovy

[EPM Automate를 설치하지 않고 명령 실행](#)를 참조하십시오.

환경에서 불필요한 파일 제거

이러한 스크립트를 사용하여 환경에서 불필요한 파일을 제거합니다.

이러한 스크립트는 다음 단계를 수행합니다.

- 환경에 로그인합니다.
- 환경의 파일 및 스냅샷을 나열합니다.
- `input.properties`에 지정된 파일을 삭제합니다.
- 로그아웃합니다.

Windows 샘플 스크립트

다음 스크립트를 복사하여 `removeUnnecessaryFiles.ps1`이라는 파일을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -raw)
$username="$($inputproperties.username)"
$passwordfile="$($inputproperties.passwordfile)"
$serviceURL="$($inputproperties.serviceURL)"
$file1="$($inputproperties.file1)"
$file2="$($inputproperties.file2)"

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate listfiles
epmautomate deletefile ${file1}
epmautomate deletefile ${file2}
epmautomate logout
```

Linux/UNIX 샘플 스크립트

다음 스크립트를 복사하여 `removeUnnecessaryFiles.sh`라는 파일을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "${serviceURL}"
${epmautomatescript} listfiles
${epmautomatescript} deletefile "${file1}"
${epmautomatescript} deletefile "${file2}"
${epmautomatescript} logout
```

input.properties 파일 생성

`removeUnnecessaryFiles` 스크립트를 실행하려면 `input.properties` 파일을 생성하고 해당 환경에 대한 정보로 업데이트합니다. `removeUnnecessaryFiles.ps1` 또는 `removeUnnecessaryFiles.sh`가 저장된 디렉토리에 파일을 저장합니다.

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
```

표 3-8 input.properties 매개변수

매개변수	설명
javahome	JAVA_HOME 위치입니다. Linux/UNIX에만 해당합니다.
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다. Linux/UNIX에만 해당합니다.
username	ID 도메인 관리자 역할도 가진 서비스 관리자의 사용자 이름입니다.
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
serviceURL	스냅샷을 생성하려는 소스 환경의 URL입니다.
file1 및 file2	환경에서 삭제할 파일 또는 스냅샷의 이름입니다. 파일이 아웃박스에 없으면 파일의 경로 및 이름을 지정합니다.

스크립트 실행

- 이전 섹션에서 스크립트를 복사하여 `removeUnnecessaryFiles.ps1` 또는 `removeUnnecessaryFiles.sh`를 생성합니다.
- `input.properties` 파일을 생성하고 `removeUnnecessaryFiles` 스크립트가 있는 디렉토리에 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. [input.properties 파일 생성](#)를 참조하십시오.
이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 **관리자 권한으로 실행** 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
- 스크립트를 실행합니다.
 - Windows PowerShell:** `removeUnnecessaryFiles.ps1`을 실행합니다.
 - Linux/UNIX:** `./removeUnnecessaryFiles.sh`를 실행합니다.

환경에서 파일 찾기 및 다운로드

이 섹션의 샘플 스크립트를 사용하면 텍스트 문자열을 와일드카드로 사용하여 Oracle Fusion Cloud Enterprise Performance Management 환경에서 하나 이상의 파일을 다운로드하는 프로세스를 자동화할 수 있습니다.

다음 스크립트를 사용하면 FILENAME 매개변수의 값으로 지정하는 문자열과 `listfiles` 명령을 사용하여 표시된 파일 이름이 일치하는지 확인한 후에 문자열과 일치하는 파일을 자동으로 다운로드할 수 있습니다.

적절한 검색 문자열을 FILENAME 매개변수에 지정하십시오. 예를 들어 FILENAME="Scheduler Output/epm"은 Scheduler Output/epm 문자열이 환경에 있는 `listfiles` 명령 출력의 파일 이름과 일치하는지 확인하여 다운로드할 파일을 확인합니다.

이 스크립트를 실행하기 위한 입력 매개변수는 `username`, `password` 또는 `password_file`, `service_url`입니다.

주

비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오.

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
```

감사를 위해 이전 Cloud EPM 환경 재생성

이 섹션의 스크립트를 사용하여 Oracle Fusion Cloud Enterprise Performance Management 환경에 대한 최신 스냅샷 라이브러리를 유지관리하는 셀프 서비스 솔루션을 생성합니다. 최신 스냅샷 라이브러리를 업그레이드하고 유지관리하기 위한 전용 환경이 필요합니다.

Cloud EPM에서는 하나의 월별 주기 동안만 스냅샷 호환성을 지원합니다. 유지관리 스냅샷을 테스트 환경에서 프로덕션 환경으로 마이그레이션하거나 그 반대로 마이그레이션할 수 있습니다. 일부 고객은 감사 요구사항을 충족하기 위해 최신 환경에서 여러 연도의 스냅샷을 복원하고 짧은 시간 안에 애플리케이션에 액세스해야 할 수도 있습니다.

매월 한 번 실행되어 사용가능한 스냅샷을 변환하고 최신 Cloud EPM 패치 레벨과 호환되게 만들도록 이 스크립트를 스케줄링해야 합니다. 스크립트를 매월 세번째 금요일 이후 실행하여 프로덕션 환경 내 모든 이슈가 해결되었는지 확인하는 것이 좋습니다.

① 주

이 스크립트를 사용하여 Narrative Reporting, Account Reconciliation, Oracle Fusion Cloud Enterprise Data Management 스냅샷을 업데이트할 수는 없습니다.

스크립트 작동 방법

업그레이드 스크립트는 EPM Automate를 사용하여 고객이 저장한 모든 스냅샷에 대해 다음 태스크를 완료합니다.

1. `input.properties` 파일의 정보를 사용하여 환경에 로그인합니다.
2. `recreate` 명령을 사용하여 환경을 개선합니다.
3. 스냅샷을 환경으로 импорт합니다.
4. 환경에서 일별 유지관리를 실행하여 스냅샷이 현재 Cloud EPM 패치 레벨과 호환되는 형식으로 변환되도록 합니다.
5. Artifact Snapshot(유지관리 스냅샷)을 폴더로 다운로드합니다. `snapshots/18.05`에서 스냅샷을 업로드하여 18.05 환경을 재생성한 경우 Artifact Snapshot이 `snapshots/18.06`으로 다운로드됩니다.
6. 이전 환경 재생성 결과를 전자메일 주소로 보냅니다(지정된 경우).

스크립트 실행

1. `input.properties` 파일을 생성하고 해당 환경에 대한 정보로 업데이트합니다. 로컬 디렉토리에 파일을 저장합니다. 본 설명에서는 이 디렉토리를 `parentsnapshotdirectory`라고 합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. 이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 **관리자 권한으로 실행** 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
2. `upgradeSnapshots.ps1`(Windows PowerShell) 또는 `upgradeSnapshots.sh`(Linux/UNIX) 스크립트를 생성하여 `input.properties`가 있는 `parentsnapshotdirectory`에 저장합니다.
3. `parentsnapshotdirectory` 내에 하위 디렉토리(예: `snapshots`)를 생성합니다.
4. 이전 단계에서 생성한 디렉토리(`snapshots`) 내에 현재 Cloud EPM 패치 레벨과 호환되도록 변환하려는 월별 스냅샷용 하위 디렉토리를 생성합니다. `YY.MM` 형식을 사용하여 디렉토리 이름을 지정합니다. 예를 들어 2018년 5월 스냅샷을 저장할 디렉토리의 이름으로 18.05를 지정합니다.
5. 스냅샷을 적절한 하위 디렉토리로 복사합니다. 예를 들어 2018년 5월 스냅샷을 `snapshots/18.05`로 복사합니다.
6. 스크립트를 실행합니다.
 - Linux/UNIX: `./upgradeSnapshots.sh`를 실행합니다.
 - Windows PowerShell: `upgradeSnapshots.ps1`을 실행합니다.

Windows

이 섹션의 스크립트를 복사하여 `input.properties` 및 `upgradeSnapshots.ps1` 스크립트를 생성합니다.

input.properties 생성

```
username=exampleAdmin
userpassword=examplePassword
```

```

serviceurl=exapleURL
proxyserverusername=proxyServerUserName
proxyserverpassword=proxyPassword
proxyserverdomain=proxyDoamin
parentsnapshotdirectory=C:/some_directory/snapshots
emailtoaddress=exampleAdmin@oracle.com

```

input.properties 업데이트

① 주

Windows 네트워크 환경에서 authentication at proxy server이 사용으로 설정되어 있지 않으면 proxyserverusername, proxyserverpassword 및 proxyserverdomain 등록정보를 input.properties 파일에서 제거하십시오.

표 3-9 input.properties 매개변수

매개변수	설명
username	서비스 관리자.의 사용자 이름
userpassword	서비스 관리자의 비밀번호입니다.
serviceurl	이 활동에 사용되는 환경의 URL입니다.
proxyserverusername	인터넷에 대한 액세스를 제어하는 프록시 서버를 통해 보안 세션을 인증할 사용자 이름입니다.
proxyserverpassword	프록시 서버를 통해 사용자를 인증할 비밀번호입니다.
proxyserverdomain	프록시 서버에 대해 정의된 도메인의 이름입니다.
parentsnapshotdirectory	처리할 스냅샷을 저장하는 디렉토리의 상위 디렉토리로 사용할 디렉토리의 절대 경로입니다. 슬래시(/)를 디렉토리 구분 문자로 사용합니다.
emailtoaddress	선택사항으로, 이전 환경 재생성 결과를 보낼 전자메일 주소입니다. 이 값이 지정된 경우에만 결과를 전자메일로 보냅니다. 예: john.doe@example.com

① 주

비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오.

upgradeSnapshots.ps1 생성

다음 샘플 스크립트를 사용하여 upgradeSnapshots.ps1을 생성하십시오.

```

# Script for recreating an old Cloud EPM 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)"
$password="$($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 $proxyserverpassword $proxy
    ProcessCommand "recreate -f"
    ProcessCommand "uploadfile $snapshotfile"
    ProcessCommand "importsnapshot $snapshotname"
    ProcessCommand "runDailyMaintenance skipNext=true -f"
    ProcessCommand "downloadfile 'Artifact Snapshot'"
    ProcessCommand "deletefile $snapshotname"
    ProcessCommand "logout"

    $nextdatedirexists=Test-Path $parentsnapshotdirectory/$nextdate
    if (!(($nextdatedirexists)) {
        mkdir $parentsnapshotdirectory/$nextdate 2>&1 | out-null
    }

    LogMessage "Renaming 'Artifact Snapshot.zip' to $snapshotname.zip and
    moving to $parentsnapshotdirectory/$nextdate"
    mv $workingdir/'Artifact Snapshot.zip' $workingdir/$snapshotname.zip
    >> $logfile 2>&1
    mv $workingdir/$snapshotname.zip $parentsnapshotdirectory/$nextdate
    >> $logfile 2>&1
}

function callSendMail
{
    $logfile=$logfile -replace "\\", "/"
    $elements=$logfile.split('/')
    $logfile=$elements[-1]

    if ($emailtoaddress -match "@") {
        epmautomate.bat login ${username} ${userpassword} ${serviceurl}
        epmautomate.bat uploadFile "$logfile"
        epmautomate.bat sendMail $emailtoaddress "Recreating An Old Cloud EPM
    Environment results" Body="The results of recreating an old Cloud EPM
    Environment are attached." Attachments=$logfile
        epmautomate.bat deleteFile "$logfile"
        epmautomate.bat logout
    }
}

#----- main body of processing
date

```

```

Init
LogAndEchoMessage "Starting upgrade snapshots processing"
$snapshotdirs=@(Get-ChildItem -Directory "$parentsnapshotdirectory" -name)
LogMessage "snapshot directories: $snapshotdirs"
$latestreleasedate=$snapshotdirs[-1]
LogMessage "latest release date: $latestreleasedate"
$latestreleasesnapshotdir="$parentsnapshotdirectory/$latestreleasedate"
LogMessage "latest release snapshot dir: $latestreleasesnapshotdir"
$nextdate=$(GetNextDate "$latestreleasedate")
$snapshotfiles=@(Get-ChildItem -File "$latestreleasesnapshotdir")
if ($snapshotfiles.length -eq 0) {
    LogAndEchoMessage "No snapshot files found in
directory $latestreleasesnapshotdir. Exiting script."
    exit
}
foreach ($snapshotfile in $snapshotfiles) {
    LogAndEchoMessage "Processing snapshotfile: $snapshotfile"
    ProcessSnapshot $latestreleasesnapshotdir/$snapshotfile $nextdate
}
LogAndEchoMessage "Upgrade snapshots processing completed"
date
callSendMail

```

Linux/UNIX

다음 스크립트를 복사하여 `upgradeSnapshots.sh` 및 `input.properties`를 생성합니다.

Linux/UNIX용 `input.properties` 생성

① 주

네트워크가 프록시 서버를 사용하여 인터넷에 액세스하도록 구성되어 있지 않으면 `proxyserverusername`, `proxyserverpassword` 및 `proxyserverdomain` 등록정보를 `input.properties` 파일에서 제거하십시오.

```

username=exampleAdmin
userpassword=examplePassword
serviceurl=exapleURL
proxyserverusername=
proxyserverpassword=
proxyserverdomain=
jdkdir=/home/user1/jdk_17.0.12
epmautomatescript=/home/exampleAdmin/epmautomate/bin/epmautomate.sh
parentsnapshotdirectory=/home/exampleAdmin/some_directory/snapshots
emailtoaddress=exampleAdmin@oracle.com

```

`input.properties` 업데이트

표 3-10 input.properties 매개변수

매개변수	설명
username	서비스 관리자의 사용자 이름
userpassword	서비스 관리자의 비밀번호입니다.
serviceurl	이 활동에 사용되는 환경의 URL입니다.
proxyserverusername	인터넷에 대한 액세스를 제어하는 프록시 서버를 통해 보안 세션을 인증할 사용자 이름입니다.
proxyserverpassword	프록시 서버를 통해 사용자를 인증할 비밀번호입니다.
proxyserverdomain	프록시 서버에 대해 정의된 도메인의 이름입니다.
jdkdir	JAVA_HOME 위치입니다.
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다.
parentsnapshotdirectory	처리할 스냅샷을 저장하는 디렉토리의 상위 디렉토리로 사용할 디렉토리의 절대 경로입니다.
emailtoaddress	선택사항으로, 이전 환경 재생성 결과를 보낼 전자메일 주소입니다.

① 주

비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오.

upgradeSnapshots.sh 생성

다음 샘플 스크립트를 사용하여 upgradeSnapshots.sh를 생성하십시오.

```
#!/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} | 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"`
}

```

```

        logout "$op" "$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 $emailtoaddress "Recreating An Old
Cloud EPM Environment results" Body="The results of recreating an old Cloud
EPM Environment are attached" Attachments=$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

```

데이터베이스 액세스 감사 및 준수 자동화

이 섹션의 PowerShell 및 Bash 셸 스크립트를 사용하여 EPM Automate 명령을 통해 수동 데이터베이스 액세스에 대한 감사 및 준수 데이터를 수집할 수 있습니다.

이 스크립트를 사용하여 다음 태스크를 완료할 수 있습니다.

- 현재 날짜의 활동 보고서 다운로드
- 보고서를 구문 분석하여 환경에 대해 수동 데이터베이스 액세스가 보고되었는지 확인
- 스크립트를 실행하는 디렉토리를 기준으로 ./reports/dataAccessAuditReport.txt 생성. 보고서에는 데이터베이스 액세스 시간 및 실행된 SQL 명령이 나열됩니다. 최신 정보를 맨위에 표시하는 누적 파일입니다. 제공되는 정보는 다음과 같습니다.
 - 보고서가 생성된 날짜 및 시간
 - 사용가능한 경우 데이터베이스 액세스 세부정보. 서비스 요청이 없는 데이터베이스 액세스와 서비스 요청이 있는 데이터베이스 액세스가 별도의 섹션에 나열됩니다. 활동 보고서에 수동 데이터베이스 액세스가 보고되지 않은 경우, No SQL statements executed가 보고서에 표시됩니다.
 - 선택적으로, 지정된 전자메일 주소로 보고서를 보냅니다.

데이터 액세스 감사 및 준수를 자동화하려면 다음을 수행합니다.

1. 다음 섹션의 스크립트 중 하나를 파일에 복사하고 파일 시스템에 저장합니다. 파일 이름을 parseActivityReport.ps1(Windows의 경우 [PowerShell 스크립트\(parseActivityReport.ps1\)](#) 참조) 또는 parseActivityReport.sh(Linux/UNIX의 경우 [Bash 셸 스크립트\(parseActivityReport.sh\)](#) 참조)로 지정합니다.
2. **Windows만 해당:** 다음 스크립트를 파일에 복사하여 parseActivityReport.bat라는 배치 파일을 생성합니다. parseActivityReport.ps1이 저장된 디렉토리에 파일을 저장합니다.

```

@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. parseActivityReport.bat(Windows) 또는 parseActivityReport.sh(Linux/UNIX 참조)를 수정하여 다음 테이블에 있는 매개변수 값을 설정합니다.

표 3-11 스크립트에 포함할 변수 값

변수	설명
epmuser	서비스 관리자의 사용자 이름 예: Windows: set epmuser="jDoe" Linux/UNIX: epmuser="jDoe"
epmpassword	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 위치. 암호화된 비밀번호 파일을 생성하는 방법에 대한 자세한 내용은 encrypt 명령을 참조하십시오. 비밀번호에 특수 문자가 포함된 경우 특수 문자 처리 를 참조하십시오. 예: Windows: set epmpassword = "Example" Linux/UNIX: epmpassword="Example"
epmurl	Oracle Fusion Cloud Enterprise Performance Management 환경의 URL입니다. 예: Windows: set epmurl="https://example.oraclecloud.com" Linux/UNIX: epmurl="https://example.oraclecloud.com"
report_email_to_address	선택사항으로, 보고서를 보낼 전자메일 주소입니다. 이 값이 지정된 경우에만 보고서를 전자메일로 보냅니다. 예: john.doe@example.com

4. **parseActivityReport.sh만 해당:** 다음 값이 시스템에 맞게 설정되었는지 확인합니다.
 - JAVA_HOME
 - epmautomatescript.sh 위치(epmautomatescript 지시어 값 업데이트)
5. 운영 체제에 제공된 스케줄러를 사용하여 parseActivityReport.bat(parseActivityReport.ps1) 또는 parseActivityReport.sh가 매일 한 번 실행되도록 스케줄링합니다. [스크립트 실행 자동화](#)를 참조하십시오.

PowerShell 스크립트(parseActivityReport.ps1)

```

# Parse Activity Report script

$epmuser=$args[0]
$epmpassword=$args[1]
$epmurl=$args[2]
$reportemailtoaddress=$args[3]

$logdir="./logs"
$logfile="$${logdir}/data_access.log"
$reportdir="./reports"
$reportfile="$${reportdir}/dataAccessAuditReport.txt"
$matchfile="$${reportdir}/matchfile.txt"
$nosrfile="$${reportdir}/data_access_nosr.csv"
$srfile="$${reportdir}/data_access_sr.csv"
$aprfilelist="$${reportdir}/aprfilelist.txt"
$activityreportfilelist="$${reportdir}/activityreportfiles.txt"
$activityreportregex='apr/[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}_[0-9]{2}_[0-9]{2}/[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}_[0-9]{2}_[0-9]{2}.html'

$global:activityreportfile=""

$NO_SQL_EXECUTED_STATEMENT="No SQL statements executed"
$SQL_WITH_SR_EXECUTED_STATEMENT="SQL statements executed with an SR"
$SQL_WITH_NO_SR_EXECUTED_STATEMENT="SQL statements executed without an SR"

function DownloadLatestActivityReport() {
    epmautomate.bat login ${epmuser} ${epmpassword} ${epmurl} >> ${logfile}
    epmautomate.bat listfiles > ${aprfilelist}
    foreach ($line in Get-Content $aprfilelist) {
        if ($line -match $activityreportregex){
            echo "$line" >> $activityreportfilelist
        }
    }
    $global:activityreportfile=Get-Content ${activityreportfilelist} -Tail 1
    $global:activityreportfile=$global:activityreportfile.trim()
    echo " "
    echo "Processing activity report file: $global:activityreportfile" | tee -
a ${logfile}
    epmautomate.bat downloadfile "$global:activityreportfile" >> ${logfile}
    epmautomate.bat logout >> ${logfile}
}

function deleteLine($file, $start, $end) {
    $i = 0
    $start--
    $end--
    (Get-Content $file) | where{
        ($i -lt $start -or $i -gt $end)
        $i++
    } > $file
    #(Get-Content $file)
}

```

```

function GenerateCsvs()
{
    $sqlregex='<DIV id="Database">.*?</DIV>'
    $activityreportfilename=Split-Path $global:activityreportfile -leaf

    echo "Creating CSV file: ${matchfile} from data in activityreportfile: ${
activityreportfilename}" >> ${logfile}
    # remove tab and newline characters
    $activityreportexists=Test-Path "$activityreportfilename"
    if ($activityreportexists) {
        (Get-Content "$activityreportfilename") -join ' ' | Set-Content
"$activityreportfilename"
        (Get-Content "$activityreportfilename") -replace "`t", "" | Set-
Content "$activityreportfilename"
    }

    # capture text matching regex
    $string=Get-Content $activityreportfilename
    $ans=$string -match $sqlregex

    if ($ans -eq "True") {
        $Matches.0 > $matchfile
        # remove HTML tags, etc.
        (Get-Content "$matchfile") -replace "<tr", "`n<tr" | Set-Content
"$matchfile"
        (Get-Content "$matchfile") -replace "<tr[^>]*>", "" | Set-Content
"$matchfile"
        (Get-Content "$matchfile") -replace "<th[^>]*>", "" | Set-Content
"$matchfile"
        (Get-Content "$matchfile") -replace "<td[^>]*>", "|" | Set-Content
"$matchfile"
        (Get-Content "$matchfile") -replace "<br>", "" | Set-Content
"$matchfile"
        (Get-Content "$matchfile") -replace "</td>", "" | Set-Content
"$matchfile"
        (Get-Content "$matchfile") -replace "</tr>", "" | Set-Content
"$matchfile"
        (Get-Content "$matchfile") -replace "\s*</table>\s*</DIV>", "" | Set-
Content "$matchfile"
        deleteLine $matchfile 1 2

        # create SR, NOSR CSV files
        Get-Content $matchfile | ForEach-Object {
            $elements=$_.split('|')
            $timeval=$elements[1].Trim()
            $srval=$elements[3].Trim()
            $sqlval=$elements[4].Trim()

            if (${srval} -eq "") {
                echo "${timeval}|${sqlval}" >> ${nosrfile}
            } else {
                if (${sqlval} -ne "") {
                    echo "${srval}|${timeval}|${sqlval}" >> ${srfile}
                }
            }
        }
    }
}

```

```

    } else { # no SQL statements in activity report
        echo "" >> ${reportfile}
        echo $(date) >> ${reportfile}
        echo "Processing activity report file: $global:activityreportfile"
    } >> ${reportfile}
    echo "${NO_SQL_EXECUTED_STATEMENT}" | tee -a ${reportfile}
    CleanUp
    EmailReportResults
    exit
}

function ReportResults() {
    echo $(date) >> ${reportfile}
    echo "Processing activity report file: $global:activityreportfile" >> ${reportfile}
    $srfileexists=Test-Path $srfile
    if ($srfileexists) {
        echo "" | tee -a ${reportfile}
        echo "${SQL_WITH_SR_EXECUTED_STATEMENT}" | tee -a ${reportfile}
        echo "SR#          Time          SQL Statement" | tee -a ${reportfile}
        echo "----          ----          -" | tee -a ${reportfile}

        # Loop through csv file and parse
        Get-Content $srfile | ForEach-Object {
            $elements=$_split('|')
            $srval=$elements[0]
            $timeval=$elements[1]
            $sqlval=$elements[2]
            echo "${srval}    ${timeval}    ${sqlval}" | tee -a ${reportfile}
        }
    }

    $nosrfileexists=Test-Path $nosrfile
    if ($nosrfileexists) {
        echo "" | tee -a ${reportfile}
        echo "${SQL_WITH_NO_SR_EXECUTED_STATEMENT}" | tee -a ${reportfile}
        echo "Time          SQL Statement" | tee -a ${reportfile}
        echo "-----          -" | tee -a ${reportfile}

        # Loop through csv file and parse
        Get-Content $nosrfile | ForEach-Object {
            $elements=$_split('|')
            $timeval=$elements[0]
            $sqlval=$elements[1]
            echo "${timeval}    ${sqlval}" | tee -a ${reportfile}
        }
    }
    EmailReportResults
}

function EmailReportResults
{
    $elements=$reportfile.split('/')
    $reportfilename=$elements[2]

```

```

if (${reportemailtoaddress} -match "@") {
    echo "Emailing Activity Report Results" | tee -a ${logfile}
    epmautomate.bat login ${epmuser} ${epmpassword} ${epmurl} >> ${logfile}
    epmautomate.bat uploadFile $reportfile >> ${logfile}
    epmautomate.bat sendMail $reportemailtoaddress "Database Access Audit
Report Results" Body="Database Access Audit Report Results are attached."
Attachments=$reportfilename >> ${logfile}
    epmautomate.bat deleteFile $reportfilename >> ${logfile}
    epmautomate.bat logout >> ${logfile}
}
}

function Init
{
    $logdirexists=Test-Path $logdir
    if (!(($logdirexists)) {
        mkdir $logdir 2>&1 | out-null
    }

    $reportdirexists=Test-Path $reportdir
    if (!(($reportdirexists)) {
        mkdir $reportdir 2>&1 | out-null
    }

    $logfileexists=Test-Path $logfile
    if ($logfileexists) {
        rm $logfile 2>&1 | out-null
    }

    $matchfileexists=Test-Path $matchfile
    if ($matchfileexists) {
        rm $matchfile 2>&1 | out-null
    }

    $nosrfileexists=Test-Path $nosrfile
    if ($nosrfileexists) {
        rm $nosrfile 2>&1 | out-null
    }

    $srfileexists=Test-Path $srfile
    if ($srfileexists) {
        rm $srfile 2>&1 | out-null
    }

    $aprfilelistexists=Test-Path $aprfilelist
    if ($aprfilelistexists) {
        rm $aprfilelist 2>&1 | out-null
    }

    $activityreportfilelistexists=Test-Path $activityreportfilelist
    if ($activityreportfilelistexists) {
        rm $activityreportfilelist 2>&1 | out-null
    }
}
}

```

```

function CleanUp
{
    $matchfileexists=Test-Path $matchfile
    if ($matchfileexists) {
        rm $matchfile 2>&1 | out-null
    }

    $aprfilelistexists=Test-Path $aprfilelist
    if ($aprfilelistexists) {
        rm $aprfilelist 2>&1 | out-null
    }

    $activityreportfilelistexists=Test-Path $activityreportfilelist
    if ($activityreportfilelistexists) {
        rm $activityreportfilelist 2>&1 | out-null
    }
}

Init
DownloadLatestActivityReport
GenerateCsvs
ReportResults
CleanUp

```

Bash 셸 스크립트(parseActivityReport.sh)

```

#!/bin/sh

export JAVA_HOME=/scratch/dteHome/autoWork/jdk_17.0.12
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}
        echo "---          ----          -" | tee -a ${reportfile}
        while read line
        do
            srval=$(echo "${line}" | cut -d'|' -f1)
            timeval=$(echo "${line}" | cut -d'|' -f2)
            sqlval=$(echo "${line}" | cut -d'|' -f3)
            echo "${srval}    ${timeval}    ${sqlval}" | tee -a ${reportfile}
        done < ${srfile}
    fi
}

```

```

if [[ -f ${nosrfile} ]]
then
    echo "" | tee -a ${reportfile}
    echo "${SQL_WITH_NO_SR_EXECUTED_STATEMENT}" | tee -a ${reportfile}
    echo "Time          SQL Statement" | tee -a ${reportfile}
    echo "----          ---" | tee -a ${reportfile}
    while read line
    do
        timeval=$(echo "${line}" | cut -d'|' -f1)
        sqlval=$(echo "${line}" | cut -d'|' -f2)
        echo "${timeval}    ${sqlval}" | tee -a ${reportfile}
    done < ${nosrfile}
fi

if [[ ! -f ${srfile} ]] && [[ ! -f ${nosrfile} ]]
then
    echo "" | tee -a ${reportfile}
    echo "${NO_SQL_EXECUTED_STATEMENT}" | tee -a ${reportfile}
fi

emailReportResults
}

downloadLatestActivityReport() {
    ${epmautomatescript} login ${epmuser} ${epmpwd} ${epmurl} >> ${logfile}
    ${epmautomatescript} listfiles > ${aprfilelist}
    activityreportfile=$(cat ${aprfilelist} | grep -P "$
{activityreportregex}" | tail -n 1 | sed -e 's/^ //' )
    echo " "
    echo "Processing activity report file: ${activityreportfile}" | tee -a $
{logfile}
    ${epmautomatescript} downloadfile "${activityreportfile}" >> ${logfile}
    ${epmautomatescript} logout >> ${logfile}
}

emailReportResults() {
    reportfilename=$(echo "${reportfile}" | cut -d'/' -f3)

    if [[ "${reportemailtoaddress}" == *@"* ]]
    then
        echo "Emailing Activity Report Results" | tee -a ${logfile}
        ${epmautomatescript} login ${epmuser} ${epmpwd} ${epmurl} >> $
{logfile}
        ${epmautomatescript} uploadFile "$reportfile" >> ${logfile}
        ${epmautomatescript} sendMail $reportemailtoaddress "Database Access
Audit Report Results" Body="Database Access Audit Report Results are
attached." Attachments=$reportfilename >> ${logfile}
        ${epmautomatescript} deleteFile "$reportfilename" >> ${logfile}
        ${epmautomatescript} logout >> ${logfile}
    fi
}

checkParams()
{
    if [ -z "$epmuser" ]

```

```
then
    echo "Username is missing."
    echo "Syntax: parseActivityReport.sh USERNAME PASSWORD URL"
    exit 2
fi

if [ -z "$epmpwd" ]
then
    echo "Password is missing."
    echo "Syntax: parseActivityReport.sh USERNAME PASSWORD URL"
    exit 2
fi

if [ -z "$epmurl" ]
then
    echo "URL is missing."
    echo "Syntax: parseActivityReport.sh USERNAME PASSWORD URL"
    exit 2
fi
}

init()
{
    checkParams

    if [ ! -d "${logdir}" ]
    then
        mkdir ${logdir}
    fi

    if [ ! -d "${reportdir}" ]
    then
        mkdir ${reportdir}
    fi

    if [ ! -f "${epmautomatescript}" ]
    then
        echo "Cannot locate EPMAutomate script: ${epmautomatescript}. Please
        check setting and run script again. Exiting." | tee -a ${logfile}
        exit
    fi

    if [ -f "${srfile}" ]
    then
        rm ${srfile}
    fi

    if [ -f "${nosrfile}" ]
    then
        rm ${nosrfile}
    fi

    if [ -f "${matchfile}" ]
    then
        rm ${matchfile}
    fi
}
```

```

        if [ -f "${aprfilelist}" ]
        then
            rm ${aprfilelist}
        fi
    }

cleanup()
{
    if [ -f "${matchfile}" ]
    then
        rm ${matchfile}
    fi

    if [ -f "${matchfile}.tmp" ]
    then
        rm ${matchfile}.tmp
    fi

    if [ -f "${aprfilelist}" ]
    then
        rm ${aprfilelist}
    fi
}

init
downloadLatestActivityReport
generateCsvs
reportResults
cleanup

```

사용자 및 사전 정의된 역할 지정 복제

이 섹션의 스크립트는 환경의 사용자 및 사전 정의된 역할 지정을 다른 환경으로 마이그레이션하는데 도움이 됩니다.

스크립트 정보

두 개의 고유한 스크립트(ID 도메인에 사용자를 복제하는 스크립트 및 사용자의 사전 정의된 역할 지정을 복제하는 스크립트)를 사용합니다. 스크립트 실행 순서는 다음과 같습니다.

- 사용자를 복제하는 스크립트(replicateusers)를 실행하고 타겟 ID 도메인에 모든 사용자가 생성되었는지 확인합니다. 이 스크립트를 실행하는 사용자에게 두 환경에서 모두 ID 도메인 관리자 및 서비스 관리자 역할이 있어야 합니다.
- 역할 지정을 복제하는 스크립트(replicatepredefinedroles)를 실행합니다.

① 주

- 비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오.
- 이 섹션의 스크립트는 사전 정의된 역할(서비스 관리자, 고급 사용자, 사용자, 조회자)에 대해서만 실행됩니다.

스크립트 실행

필수 스크립트 및 배치 파일을 생성하는 방법에 대한 자세한 내용은 다음 항목을 참조하십시오.

- [다른 ID 도메인에 ID 도메인의 사용자 복제](#)
- [환경 간에 사전 정의된 역할 지정 복제](#)

Windows 단계

1. replicateusers.bat, replicateusers.ps1, replicatepredefinedroles.bat, replicatepredefinedroles.ps1을 생성하여 쓰기 및 실행 권한이 있는 로컬 디렉토리에 저장합니다.
2. 필요한 경우 소스 및 타겟 환경과 인터넷 프록시 서버에 대한 정보로 배치 파일을 업데이트합니다.
3. replicateusers.bat를 실행합니다. 그러면 replicateusers.ps1이 실행됩니다. 복제된 사용자에게 지정할 기본 비밀번호를 다음과 같이 명령행 매개변수로 지정해야 합니다.
replicateusers.bat Pwd_for_users

비밀번호에 특수 문자가 포함된 경우 적절한 이스케이프 문자를 사용해야 합니다. [특수 문자 처리](#)를 참조하십시오.
4. replicatepredefinedroles.bat를 실행하여 소스 환경에 있는 것과 동일한 역할 지정을 생성합니다.

Linux/UNIX 단계

1. replicateusers.sh 및 replicatepredefinedroles.sh 스크립트를 생성하여 쓰기 및 실행 권한이 있는 로컬 디렉토리에 저장합니다.
2. 필요한 경우 소스 및 타겟 환경과 인터넷 프록시 서버에 대한 정보로 replicateusers.sh 및 replicatepredefinedroles.sh를 업데이트합니다.
3. replicateusers.sh를 실행합니다. 복제된 사용자에게 지정할 기본 비밀번호를 다음과 같이 명령행 매개변수로 지정해야 합니다.
./replicateusers.sh Pwd_for_users

비밀번호에 특수 문자가 포함된 경우 적절한 이스케이프 문자를 사용해야 합니다. [특수 문자 처리](#)를 참조하십시오.
4. replicatepredefinedroles.sh 스크립트를 실행하여 소스 환경에 있는 것과 동일한 역할 지정을 생성합니다.

다른 ID 도메인에 ID 도메인의 사용자 복제

이 섹션의 스크립트를 사용하여 ID 도메인의 사용자를 다른 ID 도메인에 복제할 수 있습니다. 이 스크립트를 실행하는 사용자에게 소스 및 타겟 환경에서 ID 도메인 관리자 및 서비스 관리자 역할이 있어야 합니다.

Windows

이 섹션의 스크립트를 복사하여 replicateusers.bat 및 replicateusers.ps1을 생성합니다.

1. 다음 스크립트를 복사하여 replicateusers.ps1을 생성합니다.

```
# 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 ${epmuserstarget} ${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. 다음 스크립트를 복사하여 replicateusers.bat를 생성합니다.

```

@ECHO OFF
SET thisdir=%~dp0
SET scriptpath=%thisdir%replicateusers.ps1
SET paramRequiredMessage=Syntax: replicateusers.bat "USER_PASSWORD"

REM USER DEFINED VARIABLES
REM -----
set epusersource="<EPM USER FOR SOURCE ENVIRONMENT>"
set epmpwdsourc="<EPM PASSWORD FOR SOURCE ENVIRONMENT>"
set epurlsource="<EPM URL FOR SOURCE ENVIRONMENT>"
set epidentitydomainsourc="<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 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%' -
epmusersource '%epmusersource%' -epmpwdsource '%epmpwdsource%' -
epmurlsource '%epmurlsource%' -epmidentitydomainsource
'%epmidentitydomainsource%' -epmusertarget '%epmusertarget%' -epmpwdtarget
'%epmpwdtarget%' -epmurltarget '%epmurltarget%' -epmidentitydomaintarget
'%epmidentitydomaintarget%' -proxyserverusername '%proxyserverusername%' -
proxyserverpassword '%proxyserverpassword%' -proxyserverdomain
'%proxyserverdomain%' -userpassword '%~1' -resetpassword '%resetpassword%'
-emailtoaddress '%emailtoaddress%'"

```

3. replicateusers.bat를 업데이트합니다. 지정해야 하는 값은 다음 테이블을 참조하십시오.

매개변수	설명
epmusersource	소스 환경에서 ID 도메인 관리자 및 서비스 관리자 역할을 가진 사용자의 사용자 이름입니다. 예: Windows: set epmusersource="jDoe" Linux/UNIX: epmusersource="jDoe"
epmpwdsource	사용자 비밀번호 또는 암호화된 비밀번호 파일의 절대 경로입니다. 예: Windows: set epmpwdsource="Example" Linux/UNIX: epmpwdsource="Example"
epmurlsource	복사할 사용자가 있는 환경의 URL입니다. 예: Windows: set epmurlsource="https://example.oraclecloud.com" Linux/UNIX: epmurlsource="https://example.oraclecloud.com"
epmidentitydomainsource	소스 환경에서 사용하는 ID 도메인의 이름입니다. 예: Windows: set epmidentitydomainsource="example_source_dom" Linux/UNIX: epmidentitydomainsource="example_source_dom"
epmusertarget	타겟 환경에서 ID 도메인 관리자 및 서비스 관리자 역할을 가진 사용자의 사용자 이름입니다. 예: Windows: set epmusertarget="John.Doe" Linux/UNIX: set epmusertarget="John.Doe"
epmpwdtarget	사용자 비밀번호 또는 암호화된 비밀번호 파일의 절대 경로입니다. 예: Windows: set epmpwdtarget="Example1" Linux/UNIX: epmpwdtarget="Example1"

매개변수	설명
epmurltarget	사용자를 생성할 환경의 URL입니다. 예: Windows: set epmurltarget="https://example.oraclecloud.com" Linux/UNIX: epmurltarget="https://example.oraclecloud.com"
epmidentitydomaintarget	타겟 환경에서 사용하는 ID 도메인의 이름입니다. 예: Windows: set epmidentitydomaintarget="example_source_dom" Linux/UNIX: epmidentitydomaintarget="example_target_dom"
proxyserverusername	인터넷에 대한 액세스를 제어하는 프록시 서버를 통해 보안 세션을 인증할 사용자 이름입니다. 사용되지 않는 경우 이 등록정보의 모든 발생을 삭제합니다. 예: Windows: set proxyserverusername="Example" Linux/UNIX: proxyserverusername="Example"
proxyserverpassword	프록시 서버를 통해 사용자를 인증할 비밀번호입니다. 사용되지 않는 경우 이 등록정보의 모든 발생을 삭제합니다. 예: Windows: set proxyserverpassword="examplePwd" Linux/UNIX: proxyserverpassword="examplePwd"
proxyserverdomain	프록시 서버에 대해 정의된 도메인의 이름입니다. 사용되지 않는 경우 이 등록정보의 모든 발생을 삭제합니다. 예: Windows: set proxyserverdomain="exampleDom" Linux/UNIX: proxyserverdomain="exampleDom"
emailtoaddress	선택사항으로, 역할 지정 보고서를 보낼 전자메일 주소입니다. 이 값이 지정된 경우에만 보고서를 전자메일로 보냅니다. 예: emailtoaddress=john.doe@example.com

Linux/UNIX

1. 다음 스크립트를 복사하여 replicateusers.sh를 생성합니다.

```
#!/bin/sh

userpassword="$1"

# 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>"
epmuserstarget="<<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>"
resetpassword="false"
emailtoaddress="<EMAIL TO ADDRESS>"
#-----

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. replicateusers.sh를 업데이트합니다. 지정해야 하는 값에 대한 자세한 내용은 위 테이블을 참조하십시오. 다음 등록정보의 값도 지정해야 합니다.
 - javahome: Java가 설치된 디렉터리의 절대 경로입니다.
 - epmautomatescript: epmautomatescript.sh 위치(예: epmautomatescript="/home/user1/epmate/bin/epmate.sh")입니다.

환경 간에 사전 정의된 역할 지정 복제

이 섹션의 스크립트를 사용하여 환경 간에 사전 정의된 역할 지정을 복제할 수 있습니다. 이 스크립트를 실행하는 사용자에게 두 환경에서 모두 서비스 관리자 역할이 있어야 합니다.

① 주

이 문서의 PDF 버전을 사용 중인 경우: 이 스크립트를 사용할 수 없게 만드는 줄 바꿈 및 바닥글 정보를 방지하려면 [이 항목의 HTML 버전](#)에서 스크립트를 복사합니다.

Windows

1. 다음 스크립트를 복사하여 replicatepredefineroles.ps1을 생성합니다.

```

# 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
{
    # epmate login Source App as an IDM Admin
    echo "Logging into source application at ${epmurlsource}"
    epmate 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 ${epmuser target} ${epmpwd target} ${epmurl target} $
{epmidentitydomain target} ${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. 다음 스크립트를 복사하여 replicatepredefineroles.bat를 생성합니다.

```

@ECHO OFF
SET thisdir=%~dp0

```

```

SET scriptpath=%thisdir%replicatepredefinedroles.ps1

REM USER DEFINED VARIABLES
REM -----
set epusersource="<EPM USER FOR SOURCE ENVIRONMENT>"
set epmpwdsourc="<EPM PASSWORD FOR SOURCE ENVIRONMENT>"
set epurlsource="<EPM URL FOR SOURCE ENVIRONMENT>"
set epidentitydomainsourc="<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%' -epmpwdsourc '%epmpwdsourc%' -
epurlsource '%epurlsource%' -epidentitydomainsourc
'%epidentitydomainsourc%' -epusertarget '%epusertarget%' -epmpwdtarget
'%epmpwdtarget%' -epurltarget '%epurltarget%' -epidentitydomaintarget
'%epidentitydomaintarget%' -proxyserverusername '%proxyserverusername%' -
proxyserverpassword '%proxyserverpassword%' -proxyserverdomain
'%proxyserverdomain%' -emailtoaddress '%emailtoaddress%' "

```

3. 필요에 따라 `replicatepredefineroles.bat`를 업데이트합니다. 이 파일의 등록정보에 설정해야 하는 값에 대한 자세한 내용은 다음 테이블을 참조하십시오.
replicatepredefineroles.bat 업데이트

매개변수	설명
<code>epusersource</code>	소스 환경에서 ID 도메인 관리자 및 서비스 관리자 역할을 가진 사용자의 사용자 이름입니다. 예: Windows: <code>set epusersource="jDoe"</code> Linux/UNIX: <code>epusersource="jDoe"</code>
<code>epmpwdsourc</code>	사용자 비밀번호 또는 암호화된 비밀번호 파일의 절대 경로입니다. 예: Windows: <code>set epmpwdsourc="Example"</code> Linux/UNIX: <code>epmpwdsourc="Example"</code>
<code>epurlsource</code>	복사할 사용자가 있는 환경의 URL입니다. 예: Windows: <code>set epurlsource="https://example.oraclecloud.com"</code> Linux/UNIX: <code>epurlsource="https://example.oraclecloud.com"</code>

매개변수	설명
epmidentitydomainsource	<p>소스 환경에서 사용하는 ID 도메인의 이름입니다.</p> <p>예:</p> <p>Windows: set epmidentitydomainsource="example_source_dom"</p> <p>Linux/UNIX: epmidentitydomainsource="example_source_dom"</p>
epmusertarget	<p>타겟 환경에서 ID 도메인 관리자 및 서비스 관리자 역할을 가진 사용자의 사용자 이름입니다.</p> <p>예:</p> <p>Windows: set epmusertarget="John.Doe"</p> <p>Linux/UNIX: set epmusertarget="John.Doe"</p>
epmpwdtarget	<p>사용자 비밀번호 또는 암호화된 비밀번호 파일의 절대 경로입니다.</p> <p>예:</p> <p>Windows: set epmpwdtarget="Example1"</p> <p>Linux/UNIX: epmpwdtarget="Example1"</p>
epmurltarget	<p>사용자를 생성할 환경의 URL입니다.</p> <p>예:</p> <p>Windows: set epmurltarget="https:// example.oraclecloud.com"</p> <p>Linux/UNIX: epmurltarget="https:// example.oraclecloud.com"</p>
epmidentitydomaintarget	<p>타겟 환경에서 사용하는 ID 도메인의 이름입니다.</p> <p>예:</p> <p>Windows: set epmidentitydomaintarget="example_target_dom"</p> <p>Linux/UNIX: epmidentitydomaintarget="example_target_dom"</p>
proxyserverusername	<p>인터넷에 대한 액세스를 제어하는 프록시 서버를 통해 보안 세션을 인증할 사용자 이름입니다. 사용되지 않는 경우 이 등록정보의 모든 발생을 삭제합니다.</p> <p>예:</p> <p>Windows: set proxyserverusername="Example"</p> <p>Linux/UNIX: proxyserverusername="Example"</p>
proxyserverpassword	<p>프록시 서버를 통해 사용자를 인증할 비밀번호입니다. 사용되지 않는 경우 이 등록정보의 모든 발생을 삭제합니다.</p> <p>예:</p> <p>Windows: set proxyserverpassword="examplePwd"</p> <p>Linux/UNIX: proxyserverpassword="examplePwd"</p>
proxyserverdomain	<p>프록시 서버에 대해 정의된 도메인의 이름입니다. 사용되지 않는 경우 이 등록정보의 모든 발생을 삭제합니다.</p> <p>예:</p> <p>Windows: set proxyserverdomain="exampleDom"</p> <p>Linux/UNIX: proxyserverdomain="exampleDom"</p>
emailtoaddress	<p>선택사항으로, 역할 지정 보고서를 보낼 전자메일 주소입니다. 이 값이 지정된 경우에만 보고서를 전자메일로 보냅니다.</p> <p>예: emailtoaddress=john.doe@example.com</p>

Linux/UNIX

1. 다음 스크립트를 복사하여 `replicatepredefineroles.sh`를 생성합니다.

```
#!/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>"
epmuserstarget="<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
    echo "Downloading role assignment report"
    ${epmautomatescript} downloadfile ${roleassignmentreport}
    ${epmautomatescript} deletefile ${roleassignmentreport}
    ${epmautomatescript} logout

    echo "Creating files to use with epmautomate assignRoles"
    while read line
    do
        user=$(echo "${line}" | cut -d',' -f1)
        rolename=$(echo "${line}" | cut -d',' -f5)
```

```

        if [[ "$rolename" == *"User" ]] && [[ "$rolename" != "*Power
User" ]]
        then
            count=$(echo "${rolename}" | wc -w);
            rolename="User"
            if [[ $count -le 2 ]]
            then
                echo "${user}" >> "role-${rolename}.csv"
            fi
        elif [[ "$rolename" == *"Viewer" ]]
        then
            count=$(echo "${rolename}" | wc -w);
            rolename="Viewer"
            if [[ $count -le 2 ]]
            then
                echo "${user}" >> "role-${rolename}.csv"
            fi
        elif [[ "$rolename" == *"Power User" ]]
        then
            count=$(echo "${rolename}" | wc -w);
            rolename="Power User"
            if [[ $count -le 3 ]]
            then
                echo "${user}" >> "role-${rolename}.csv"
            fi
        elif [[ "$rolename" == *"Service Administrator" ]]
        then
            count=$(echo "${rolename}" | wc -w);
            rolename="Service Administrator"
            if [[ $count -le 3 ]]
            then
                echo "${user}" >> "role-${rolename}.csv"
            fi
        elif [[ "$rolename" == "Planner" ]]
        then
            echo "${user}" >> "role-User.csv"
        fi
    done < ${roleassignmentreport}

    # write header line
    for f in role-*.csv
    do
        sed -i 'liUser Login' "$f"
    done

    # epmautomate login Target App as an IDM Admin
    echo "Logging into target application at ${epmurltarget}"
    ${epmautomatescript} login ${epmusertarget} ${epmpwdtarget} $
    {epmurltarget} ${epmidentitydomaintarget} ${proxyserverusername} $
    {proxyserverpassword} ${proxyserverdomain}

    for rolefile in role-*.csv
    do
        rolenamecsv=$(echo "$rolefile" | cut -d'-' -f2)
        rolename=$(echo "$rolenamecsv" | cut -d'.' -f1)
        ${epmautomatescript} deletedefile "${rolefile}" > /dev/null 2>&1
    done

```

```

        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. `replicatepredefineroles.sh`를 업데이트합니다. 지정해야 하는 값에 대한 자세한 내용은 위 테이블을 참조하십시오. 다음 등록정보의 값도 지정해야 합니다.
 - `javahome`: Java가 설치된 디렉터리의 절대 경로입니다.
 - `epmautomatescript`: `epmautomatescript.sh` 위치(예: `epmautomatescript="/home/user1/epmautomate/bin/epmautomate.sh"`)입니다.

분기별 Cloud EPM 업그레이드 주기 생성

이러한 스크립트를 사용하여 Cloud EPM 환경이 2주 테스트 주기에 따라 분기별로 업데이트되도록 업데이트를 건너뛰는 셀프 서비스 솔루션을 생성합니다. 이 경우 프로덕션 환경은 테스트 환경 2주 후에 업데이트됩니다.

필요한 경우 이 스크립트를 사용하여 두 달에 한 번 월별 업데이트를 건너뛸 수도 있습니다. 기본적으로 Cloud EPM은 월별 업데이트를 환경에 적용합니다. [skipUpdate](#) 명령을 사용하면 환경에 월별 업데이트를 적용하는 것을 건너뛰거나 현재 업데이트 건너뛰기 요청을 볼 수 있습니다. 이 섹션에 포함된 스크립트를 사용하여 `skipUpdate` 명령 수동 실행을 자동화할 수 있습니다. 이러한 스크립트는 업데이트 건너뛰기 프로세스를 자동화하여 업데이트가 분기별로 또는 두 달에 한 번 적용되도록 합니다.

Note

1. 업데이트는 최대 3개월까지만 연속으로 건너뛸 수 있습니다. 예를 들어, Cloud EPM 환경이 2월, 7월, 12월에만 업데이트되도록 하면 스크립트에서 오류가 발생합니다.
2. 중간에 있는 기간 중 발생한 모든 업데이트는 다음 업데이트 시 환경에 적용됩니다. 예를 들어 이 스크립트를 사용하여 분기별 업데이트가 2월, 5월, 8월, 11월에만 발생하도록 스케줄링한다고 가정해보겠습니다. 이 경우, 예를 들어, 5월 업데이트는 환경에 2월 업데이트 이후 릴리스된 모든 적용 가능한 Cloud EPM 월별 업데이트 및 패치를 적용합니다. 업데이트가 적용되는 경우 유지관리 프로세스에 보통보다 시간이 더 오래 걸릴 수 있습니다.
3. 이 스크립트는 하나의 분기에 대한 업데이트 주기만 설정합니다. 이 스크립트를 월별로 실행하여 전체 연도의 업데이트 주기가 구성되었는지 확인하십시오.

- [Windows 스크립트 및 지침](#)
- [UNIX/Linux 스크립트 및 지침](#)
- [Groovy 스크립트](#)

스크립트 실행

1. Windows 및 Linux/UNIX 스크립트를 실행하려면 다음을 수행합니다.
 - a. `input.properties` 파일을 생성하고 해당 환경에 대한 정보로 업데이트합니다. 로컬 디렉토리에 파일을 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. 이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 **관리자 권한으로 실행** 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
 - b. `skip_update.ps1`(Windows PowerShell) 또는 `skip_update.sh`(Linux/UNIX) Bash 스크립트를 생성하여 `input.properties`가 있는 디렉토리에 저장합니다.
 - c. 스크립트를 실행합니다.
 - Linux/UNIX: `./skip_update.sh`를 실행합니다.
 - Windows PowerShell: `skip_update.ps1`을 실행합니다.
2. Groovy 스크립트를 실행하려면 Cloud EPM 비즈니스 프로세스에서 Groovy 화면을 사용하거나 [runBusinessRule](#)을 사용하여 스크립트 실행을 자동화합니다. EPM Automate를 사용하여 Groovy 스크립트를 실행하는 방법에 대한 자세한 내용은 [EPM Automate를 설치하지 않고 명령 실행](#)을 참조하십시오.

Windows 스크립트 및 지침

이 섹션의 스크립트를 복사하여 `input.properties` 및 `skip_update.ps1`을 생성합니다.

1. 다음 스크립트를 복사하여 `input.properties`를 생성합니다.

```
username=exampleAdmin
password=examplePassword.epw
url=exampleURL
updatemonths=02,05,08,11
```

2. 매개변수 값을 지정하여 `input.properties`를 업데이트합니다.

Table 3-12 input.properties 매개변수

매개변수	설명
username	서비스 관리자의 사용자 이름
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
url	월별 이외의 업데이트 주기를 설정할 환경의 URL입니다.
updatemonths	Oracle Fusion Cloud Enterprise Performance Management 업데이트가 url 매개변수로 확인된 환경에 적용되어야 하는 월에 대한 심표로 구분된 목록입니다. updatemonths=02,05,08,11을 예로 들 수 있습니다. 월은 1월인 01에서 12월인 12까지 두 자리 숫자로 지정해야 합니다. 1월에서 9월 앞에는 0을 포함해야 합니다. 스크립트는 updatemonths 매개변수 값에 포함되지 않은 월에 skipUpdate 명령을 실행하려고 시도합니다. 예를 들어 updatemonths=02,05,08,11을 지정하는 경우 스크립트는 1월, 3월, 4월, 6월, 7월, 9월, 10월, 12월에 업데이트 건너뛰기 플래그를 설정하여 2월, 5월, 8월, 11월에만 업데이트가 수행되도록 합니다.

3. 다음 스크립트를 복사하여 skip_updates.ps1을 생성합니다.

```
# 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
            }
        }

        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 4) {
        echo "There are more than 3 months skipped from month ${thismonth}
to month ${nextmonthorig}. Please correct updatemonths in input.properties
so that there are not more than three months skipped between each update
month. Exiting."
        exit 1
    }
}

function validateUpdateMonths() {
    for ($i = 0; $i -le ($updatemonthsarr.length - 1); $i++) {
        $nextint = $i + 1
        $thisupdatemonth = $updatemonthsarr[$i]
        $thisupdatemonthint=[int]$thisupdatemonth
        $nextupdatemonth=$updatemonthsarr[$nextint]
        $nextupdatemonthint=[int]$nextupdatemonth

        if (${nextupdatemonth} -eq "") {
            $nextupdatemonth=$updatemonthsarr[0]
            $nextupdatemonthint=[int]$nextupdatemonth
        }

        compareUpdateMonths $thisupdatemonthint $nextupdatemonthint
    }
}

validateUpdateMonths
populateFromCurrentArrays
processSkipUpdates

```

UNIX/Linux 스크립트 및 지침

이 섹션의 스크립트를 복사하여 `input.properties` 및 `skip_update.sh`를 생성합니다.

1. 다음 스크립트를 복사하여 `input.properties`를 생성합니다.

```

javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
username=exampleAdmin
password=examplePassword.epw
url=exampleURL
updatemonths=02,05,08,11

```

2. 매개변수 값을 지정하여 `input.properties`를 업데이트합니다.

Table 3-13 `input.properties` 매개변수

매개변수	설명
<code>javahome</code>	JAVA_HOME 위치입니다.

Table 3-13 (Cont.) input.properties 매개변수

매개변수	설명
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다.
username	서비스 관리자.의 사용자 이름
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
url	월별 이외의 업데이트 주기를 설정할 환경의 URL입니다.
updatemonths	Oracle Fusion Cloud Enterprise Performance Management 업데이트가 url 매개변수로 확인된 환경에 적용되어야 하는 월에 대한 심표로 구분된 목록입니다. updatemonths=02,05,08,11을 예로 들 수 있습니다. 월은 두 자리 숫자로 지정해야 합니다. 1월에서 9월 앞에는 0이 포함됩니다. 스크립트는 updatemonths 매개변수 값에 포함되지 않은 월에 skipUpdate 명령을 실행하려고 시도합니다. 예를 들어 updatemonths=02,05,08,11을 지정하는 경우 스크립트는 1월, 3월, 4월, 6월, 7월, 9월, 10월, 12월에 업데이트 건너뛰기 플래그를 설정하여 2월, 5월, 8월, 11월에만 업데이트가 수행되도록 합니다.

3. 다음 스크립트를 복사하여 skip_updates.sh를 생성합니다.

```
#!/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 4 ]]
    then
        echo "There are more than 3 months skipped from month ${thismonth}
to month ${nextmonthorig}. Please correct updatemonths in input.properties
so that there are not more than three months skipped between each update
month. Exiting."
        exit 1
    fi
}

validateUpdateMonths() {
    for i in ${!updatemonthsarr[@]}
    do
        nextint=$((i+1))
        thisupdatemonth="${updatemonthsarr[$i]}"
        thisupdatemonthhint=${thisupdatemonth#0}
        nextupdatemonth="${updatemonthsarr[$nextint]}"
        nextupdatemonthhint=${nextupdatemonth#0}

        if [[ ${nextupdatemonth} == "" ]]
        then
            nextupdatemonth="${updatemonthsarr[0]}"
            nextupdatemonthhint=${nextupdatemonth#0}
        fi

        compareUpdateMonths ${thisupdatemonthhint} ${nextupdatemonthhint}
    done
}

validateUpdateMonths
populateFromCurrentArrays
processSkipUpdates

```

Groovy 스크립트

비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오. 또한, 환경에 적합하게 다음 매개변수 값을 바꾸십시오.

Table 3-14 변경할 매개변수

매개변수	설명
user	서비스 관리자.의 사용자 이름
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
url	월별 이외의 업데이트 주기를 설정할 환경의 URL입니다.
updatemonths	Oracle Fusion Cloud Enterprise Performance Management 업데이트가 url 매개변수로 확인된 환경에 적용되어야 하는 월에 대한 심표로 구분된 목록입니다. updatemonths=02,05,08,11을 예로 들 수 있습니다. 월은 1월인 01에서 12월인 12까지 두 자리 숫자로 지정해야 합니다. 1월에서 9월 앞에는 0을 포함해야 합니다. 스크립트는 updatemonths 매개변수 값에 포함되지 않은 월에 skipUpdate 명령을 실행하려고 시도합니다. 예를 들어 updatemonths=02,05,08,11을 지정하는 경우 스크립트는 1월, 3월, 4월, 6월, 7월, 9월, 10월, 12월에 업데이트 건너뛰기 플래그를 설정하여 2월, 5월, 8월, 11월에만 업데이트가 수행되도록 합니다.

```
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 > 4) {
        LogMessage('There are more than three months skipped from month ' +
thismonth + ' to month ' + nextmonthorig + '. Please correct updatemonths so
that there are not more than three 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')

```

6주 테스트 주기로 분기별 Cloud EPM 업그레이드 주기 생성

이 섹션의 스크립트를 사용하여 Oracle Fusion Cloud Enterprise Performance Management 환경이 6주 테스트 주기에 따라 분기별로 업데이트되도록 업데이트를 건너뛰는 셀프 서비스 솔루션을 생성합니다. 이 경우 프로덕션 환경은 테스트 환경 6주 후에 업데이트됩니다.

기본적으로 Cloud EPM은 월별 업데이트를 환경에 적용합니다. [skipUpdate](#) 명령을 사용하면 환경에 월별 업데이트를 적용하는 것을 건너뛰거나 현재 업데이트 건너뛰기 요청을 볼 수 있습니다. 이 섹션에 포함된 스크립트를 사용하여 skipUpdate 명령 수동 실행을 자동화할 수 있습니다. 이러한 스크립트는 업데이트 건너뛰기 프로세스를 자동화하여 업데이트가 6주 테스트 주기에 따라 분기별로 적용되도록 합니다.

Note

1. 업데이트는 최대 3개월까지만 연속으로 건너뛴 수 있습니다. Cloud EPM 환경이 2월, 11월에만 업데이트되도록 하면 이 스크립트에서 오류가 발생합니다.
2. 중간에 있는 기간 중 발생한 모든 업데이트는 다음 업데이트 시 환경에 적용됩니다. 예를 들어 이 스크립트를 사용하여 분기별 업데이트가 2월, 5월, 8월, 11월 업데이트에만 발생하도록 스케줄링한다고 가정해보겠습니다. 이 경우, 예를 들어, 5월 업데이트는 환경에 2월 업데이트 이후 릴리스된 모든 적용 가능한 Cloud EPM 월별 업데이트 및 패치를 적용합니다. 업데이트가 적용되는 경우 유지관리 프로세스에 보통보다 시간이 더 오래 걸릴 수 있습니다.
3. 이 스크립트는 하나의 분기에 대한 업데이트 주기만 설정합니다.
 샘플 시나리오: 테스트 환경 업데이트 주기는 2월(24.02 업데이트), 5월(24.05 업데이트), 8월(24.08 업데이트), 11월(24.11 업데이트)의 첫번째 금요일로 설정됩니다. 프로덕션 환경은 2월(24.02 업데이트)의 첫번째 금요일에 테스트 환경을 업데이트하는 데 사용된 버전으로 3월(24.02 업데이트)의 세번째 금요일에 업데이트됩니다. 프로덕션 환경에 대한 비슷한 업데이트는 6월(24.05 업데이트), 9월(24.08 업데이트), 12월(24.11 업데이트)의 세번째 주에 발생합니다. 이 시나리오에서 프로덕션 환경은 현재 업데이트가 아니라 현재 테스트 환경에 적용된 업데이트로 업데이트됩니다.

Windows 샘플 스크립트

다음 스크립트를 복사하여 skip_update.ps1을 생성합니다. 로컬 디렉토리에 파일을 저장합니다. 이 스크립트를 실행하는 방법에 관한 내용은 [스크립트 실행](#)을 참조하십시오.

```
# 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] $global:monthsarrfromcurrent[$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 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

```

Linux/UNIX 샘플 스크립트

다음 스크립트를 복사하여 skip_update.sh를 생성합니다. 로컬 디렉토리에 파일을 저장합니다. 이 스크립트를 실행하는 방법에 관한 내용은 [스크립트 실행](#)을 참조하십시오.

```

#!/bin/sh

. ./input.properties

echo "Starting skip_update.sh script."

export JAVA_HOME=${javahome}

updateversionsarr=( $(echo "${updateversions}" | sed 's/,/ /g') )
declare -a updatemonthsarr

year=$(date +%Y)
month=$(date +%m)
day=$(date +%d)
yearcurrent="${year:2:4}"
yearnext=$((yearcurrent+1))
yearprevious=$((yearcurrent-1))
dayofyear=""

populateUpdateMonthsArray() {
    if [[ "${podtype}" == "test" ]]
    then

```

```

        updatemonthsarr="{updateversionsarr[@]}"
    else
        for i in ${updateversionsarr[@]}
        do
            version=$((10#i))
            if [[ ${version} == "12" ]]
            then
                nextversion="01"
            else
                nextversion=$((version+1))
            fi
            printf -v nextversion "%02d" $nextversion
            updatemonthsarr+=("${nextversion}")
        done
    fi
}

getVersionPushDay() {
    local podtype="$1"
    local month="$2"
    local count=0

    for s in `seq 1 28`
    do
        fr=$(date +%a --date="${year}-${month}-${s}" )
        if [ "$fr" == "Fri" ];then
            friday[$count]=$s
            count=$((count+1))
        fi
    done

    if [[ "${podtype}" == "prod" ]]
    then
        echo ${friday[2]}
    else
        echo ${friday[0]}
    fi
}

getVersionPushDayOfYear() {
    local year="$1"
    local month="$2"
    local day="$3"
    local pushday=""

    pushday=$(date -d "${year}-${month}-${day}" +%j)
    pushday=$((10#pushday))

    echo "${pushday}"
}

skipUpdateAdd() {
    local version="$1"

    echo "Running: epmautomate skipUpdate add version=${version}"
    comment="\`adding skipUpdate\`"
}

```

```

    ${epmautomatescript} skipUpdate add version=${version} comment="adding
skipUpdate"
}

getVersionInstalled() {
    local version=""
    local response=$(curl -s -u "${username}:${password}" -H 'Content-Type:
application/json' "${url}/interop/rest/v2/maintenance/
getdailymaintenancestarttime")
    version=$(echo "${response}" | jq '.items[0].buildVersion' | cut -d'"' -
f2 | rev | cut -d'.' -f2- | rev)

    if [[ ${version} == "" ]]
    then
        echo "Unable to determine installed EPM Cloud version. Exiting."
        exit 1
    fi

    echo "${version}"
}

getVersionNext() {
    local version="$1"
    local versionyear=$(echo "${version}" | cut -d '.' -f1)
    local versionmonth=$(echo "${version}" | cut -d '.' -f2)
    local versionnext=""

    if [[ "${versionmonth}" == "12" ]]
    then
        versionyearnext=$((versionyear+1))
        versionnext="${versionyearnext}.01"
    else
        versionmonthnext=$((10#versionmonth+1))
        versionmonthnext=$(printf '%02d\n' "$versionmonthnext")
        versionnext="${versionyear}.${versionmonthnext}"
    fi

    echo "${versionnext}"
}

processSkipUpdates() {
    local day=$1
    local versionpushday=$2
    local version=""
    local daybeforeversionpushday=$((versionpushday-1))

    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}"
    fi
}

```

```

    ${epmautomatescript} login ${username} ${password} ${url}
ProxyServerUserName=${proxyserverusername} ProxyServerPassword=${
{proxyserverpassword} ProxyServerDomain=${proxyserverdomain}
    fi

    echo "Running: epmautomate skipUpdate remove"
    ${epmautomatescript} skipUpdate remove

    if [[ "${podtype}" == "test" ]]
    then
        if [[ ${updateversionsarr[@]} =~ ${month} ]] && [[ ${day} -eq ${
{versionpushday} ]]
        then
            return
        else
            version=$(getVersionInstalled)
            echo "Installed version: $version"
            versionnext=$(getVersionNext "${version}")
            versionafternext=$(getVersionNext "${versionnext}")
            skipUpdateAdd "${versionnext}"
            skipUpdateAdd "${versionafternext}"
        fi
    else # prod
        if [[ ${updatemonthsarr[@]} =~ ${month} ]] && [[ ${day} -eq ${
{daybeforeversionpushday} ]]
        then
            return
        else
            version=$(getVersionInstalled)
            echo "Installed version: $version"
            versionnext=$(getVersionNext "${version}")
            versionafternext=$(getVersionNext "${versionnext}")
            versionafternextnext=$(getVersionNext "${versionafternext}")
            skipUpdateAdd "${versionnext}"
            skipUpdateAdd "${versionafternext}"
            skipUpdateAdd "${versionafternextnext}"
        fi
    fi
    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]}"
        thisupdatemonthint=${thisupdatemonth#0}
        nextupdatemonth="${updateversionsarr[$nextint]}"
        nextupdatemonthint=${nextupdatemonth#0}

        if [[ ${nextupdatemonth} == "" ]]
        then
            nextupdatemonth="${updateversionsarr[0]}"
            nextupdatemonthint=${nextupdatemonth#0}
        fi

        compareUpdateMonths ${thisupdatemonthint} ${nextupdatemonthint}
    done
}

validateUpdateVersions
populateUpdateMonthsArray
versionpushday=$(getVersionPushDay ${podtype} ${month})
versionpushdayofyear=$(getVersionPushDayOfYear ${year} ${month} $
{versionpushday})
dayofyear=$(date -d "${year}-${month}-${day}" +%j)
dayofyear=$((10#$dayofyear))
processSkipUpdates "${dayofyear}" "${versionpushdayofyear}"

```

Groovy 스크립트

다음 스크립트를 복사하여 `skip_update.groovy`를 생성하고 업데이트합니다. 이 스크립트를 실행하는 방법에 관한 내용은 [스크립트 실행](#)을 참조하십시오. 이 Groovy 스크립트에서 다음 변수를 업데이트합니다.

- `username` - 월별 이외의 업데이트 주기를 설정하려는 환경에서 서비스 관리자의 사용자 이름입니다.
- `password` - 서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
- `url` - 월별 이외의 업데이트 주기를 설정할 환경의 URL입니다.
- `updateversions` - url 매개변수로 확인된 환경에 적용해야 하는 Cloud EPM 업데이트의 심포로 구분된 목록입니다. 예: `updateversions=02,05,08,11`. 버전은 두 자리 숫자로 지정해야 합니다. 업데이트 01 ~ 09 앞에는 0이 포함됩니다. 이 스크립트는 `updateversions` 매개변수 값에 포함되지 않은 업데이트에 대해 `skipUpdate` 명령을 실행하려고 시도합니다. 예를 들어 `updateversions=02,05,08,11`을 지정하면 이 스크립트는 01(1월), 03(3월), 04(4월), 06(6월), 07(7월), 09(9월), 10(10월), 12(12월) 업데이트에 대해

업데이트 건너뛰기 플래그를 설정하려고 시도합니다. 이 경우 Cloud EPM 업데이트 02(2월), 05(5월), 08(8월), 11(11월)이 환경에 적용됩니다.

- podtype - Cloud EPM 환경 유형입니다. 적합한 값은 test 및 prod입니다.
- proxyserverusername - 인터넷에 대한 액세스를 제어하는 프록시 서버를 사용하여 보안 세션을 인증할 사용자 이름입니다.
- proxyserverpassword - 프록시 서버를 사용하여 사용자를 인증할 비밀번호입니다.
- proxyserverdomain - 프록시 서버에 대해 정의된 도메인의 이름입니다.

① Note

프록시 서버를 사용하지 않는 경우에는 proxyserverusername, proxyserverpassword, proxyserverdomain 매개변수의 값을 지정하지 마십시오.

```
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')

```

input.properties 파일을 생성하여 skip_update Windows 및 Linux/UNIX 스크립트 실행

skip_update.ps1 또는 skip_update.sh를 실행하려면 input.properties 파일을 생성하고 해당 환경에 대한 정보로 업데이트합니다. 로컬 디렉토리에 파일을 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다.

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
updateversions=01,04,07,10
podtype=test

```

Table 3-15 input.properties 매개변수

매개변수	설명
javahome	JAVA_HOME 위치입니다. Linux/UNIX에만 해당합니다.
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다. Linux/UNIX에만 해당합니다.
username	서비스 관리자.의 사용자 이름
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
url	월별 이외의 업데이트 주기를 설정할 환경의 URL입니다.

Table 3-15 (Cont.) input.properties 매개변수

매개변수	설명
updateversions	updateversions - url 매개변수로 확인된 환경에 적용해야 하는 Cloud EPM 업데이트의 심표로 구분된 목록입니다. 예: updateversions=02,05,08,11. 버전은 두 자리 숫자로 지정해야 합니다. 업데이트 01 ~ 09 앞에는 0이 포함됩니다. 이 스크립트는 updateversions 매개변수 값에 포함되지 않은 업데이트에 대해 skipUpdate 명령을 실행하려고 시도합니다. 예를 들어 updateversions=02,05,08,11을 지정하면 이 스크립트는 01(1월), 03(3월), 04(4월), 06(6월), 07(7월), 09(9월), 10(10월), 12(12월) 업데이트에 대해 업데이트 건너뛰기 플래그를 설정하려고 시도합니다. 이 경우 Cloud EPM 업데이트 02(2월), 05(5월), 08(8월), 11(11월)이 환경에 적용됩니다.
podtype	Cloud EPM 환경 유형입니다. 적합한 값은 test 및 prod입니다.

스크립트 실행

1. Windows 및 Linux/UNIX에만 해당:

- 이전 섹션에서 스크립트를 복사하여 skip_update.ps1 또는 skip_update.sh를 생성합니다.
- input.properties 파일을 생성하고 skip_update 스크립트가 있는 디렉토리에 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. [input.properties 파일을 생성하여 skip_update Windows 및 Linux/UNIX 스크립트 실행](#)을 참조하십시오. 이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 **관리자 권한으로 실행** 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
- 스크립트를 실행합니다.
 - Windows PowerShell:** skip_update.ps1을 실행합니다.
 - Linux/UNIX:** ./skip_update.sh를 실행합니다.

2. Groovy:

- skip_update.groovy Groovy 스크립트를 생성하고 필요에 따라 업데이트합니다.
- Cloud EPM 비즈니스 프로세스에서 Groovy 화면을 사용하거나 [runBusinessRule](#)을 사용하여 스크립트 실행을 자동화합니다. EPM Automate를 사용하여 Groovy 스크립트를 실행하는 방법에 대한 자세한 내용은 [EPM Automate를 설치하지 않고 명령 실행](#)을 참조하십시오.

Planning, Consolidation, Tax Reporting, Enterprise Profitability and Cost Management 샘플 시나리오

이 섹션에서 사용가능한 스크립트는 Planning(Planning 모듈 포함), Financial Consolidation and Close, Tax Reporting, Enterprise Profitability and Cost Management 환경의 태스크를 자동화하는 데 도움이 됩니다.

참조:

- [집계 저장영역 큐브에서 다수의 셀 익스포트 자동화](#)
이 섹션의 PowerShell 또는 Bash 스크립트를 사용하여 ASO(집계 저장영역) 큐브에서 다수의 셀을 익스포트할 수 있습니다.

- [애플리케이션으로 메타데이터 импорт](#)
이러한 스크립트를 사용하여 파일에서 애플리케이션 메타데이터를 수동으로 импорт합니다.
- [데이터 импорт, 계산 스크립트 실행 및 블록 저장영역 데이터베이스에서 집계 저장영역 데이터베이스로 데이터 복사](#)
이러한 스크립트를 사용하여 파일에서 데이터를 импорт하고, 큐브를 새로고침하고, 비즈니스 규칙을 실행하여 큐브를 계산하고, 데이터를 ASO 큐브로 푸시합니다.
- [메타데이터와 데이터를 익스포트 및 다운로드](#)
이러한 스크립트를 사용하여 애플리케이션 메타데이터 및 데이터를 익스포트한 다음, 익스포트 파일을 로컬 디렉토리로 다운로드합니다.
- [애플리케이션 데이터 익스포트 및 다운로드](#)
이러한 스크립트를 사용하여 애플리케이션 데이터를 익스포트한 다음, 로컬 디렉토리로 다운로드합니다.
- [애플리케이션 감사 레코드 아카이브 자동화](#)
이 섹션의 Windows 및 Linux 스크립트를 사용하여 애플리케이션 감사 데이터를 로컬 컴퓨터로 익스포트하고 아카이브하는 프로세스를 자동화합니다.
- [데이터 파일을 환경으로 업로드 및 데이터 로드 규칙 실행](#)
이러한 스크립트를 사용하여 파일을 환경으로 업로드한 다음, 데이터 규칙을 실행하여 파일의 데이터를 애플리케이션으로 импорт합니다.
- [일일 데이터 통합 자동화](#)
이 시나리오에서는 샘플 스크립트를 사용하여 정기적인 데이터 통합을 자동화하는 방법을 살펴봅니다.

집계 저장영역 큐브에서 다수의 셀 익스포트 자동화

이 섹션의 PowerShell 또는 Bash 스크립트를 사용하여 ASO(집계 저장영역) 큐브에서 다수의 셀을 익스포트할 수 있습니다.

Oracle Essbase QUERYRESULTLIMIT를 통해 적용되는 한도로 인해 사용자 인터페이스에서 대량 데이터를 익스포트할 수 없습니다. 이 섹션에 제공된 PowerShell 스크립트는 익스포트 작업을 지정된 개수의 작업으로 분할하고 각 작업을 실행한 다음, 익스포트된 데이터를 다운로드하고 머릿글이 하나만 있도록 여러 익스포트 파일을 하나의 익스포트 파일로 연결합니다.

① 주

이러한 스크립트는 데이터 익스포트 유형의 기존 작업을 실행합니다. 작업 생성 방법에 대한 자세한 내용은 *Planning 관리*에서 "작업 관리"를 참조하십시오.

PowerShell 스크립트

```
$user = '<USERNAME>'
$pass = '<PASSWORD>'
$serverURL = '<URL>'
$applicationName = '<APPLICATIONNAME>'
$cubeName = '<CUBENAME>'
$splitDimension = '<DIMENSION_TO_SPLIT_THE_EXPORT>'
$stopLevelMemberForExport = '<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 members in export hierarchy
Write-Host "Retrieving Level 0 members for hierarchy"
getLevel0($member)
if ( $memberArray.Length -eq 0 ) {
    Write-Host "no level 0 members found in hierarchy, exiting ..."
    return 128
}
Write-Host $memberArray.Length " Level 0 members for export hierarchy
retrieved"

$request = $serverURL + '/HyperionPlanning/rest/v3/applications/'
+ $applicationName + '/jobs'

#splitting member list into the number of export files
$numberOfEntitiesPerFile =
[math]::truncate($memberArray.Length / $numberOfExportFiles)
for ($i = 1; $i -le $numberOfExportFiles; $i++) {
    $memberList = ""
    $firstMember = ($i - 1) * $numberOfEntitiesPerFile
    if ($i -lt $numberOfExportFiles) {
        $lastMember = $i * $numberOfEntitiesPerFile
    } else {
        $lastMember = $i * $numberOfEntitiesPerFile + $memberArray.Length
    }
    % $numberOfExportFiles
    for ($j = $firstMember; $j -lt $lastMember; $j++) {
        $memberList += $memberArray[$j]
        if ($j -lt $lastMember - 1) {$memberList += ","} #avoid adding a
comma (,) after the last member of each set
    }

    $jobDetails='
    {
    "jobType":"EXPORT_DATA","jobName":"' + $exportJobName + '",
    "parameters":{
        "exportFileName":"Export-' + $i + '.zip",
        "rowMembers":"' + $memberList + '",
        "columnMembers":"' + $columnMembers + '",
        "povMembers":"' + $povMembers + '"
    }
    }'
}

```

```

#start export job
try{
    $response = Invoke-RestMethod -Uri $request -Method Post -
Headers $headers -Body $jobDetails -ContentType "application/json"}
catch {
    Write-Host $_
    return
}

Write-Host "Started export job " $i " out of " $numberOfExportFiles

#checking job status, continue once jos is completed
$statusRequest = $serverURL + '/HyperionPlanning/rest/v3/applications/'
+ $applicationName + '/jobs/' + $response.jobId
$statusResponse = Invoke-RestMethod -Uri $statusRequest -Method Get -
Headers $headers -UseBasicParsing

while ( $statusResponse.descriptiveStatus -eq "Processing" ) {
    Write-Host $statusResponse.descriptiveStatus
    Start-Sleep -s 10
    $statusResponse = Invoke-RestMethod -Uri $statusRequest -Method Get -
Headers $headers -UseBasicParsing
}
Write-Host $statusResponse.descriptiveStatus

Write-Host "Downloading export file ..."
$downloadRequest = $serverURL + '/interop/rest/11.1.2.3.600/
applicationsnapshots/Export-' + $i + '.zip/contents'
$statusResponse = Invoke-RestMethod -Uri $downloadRequest -Method Get -
Headers $headers -OutFile "$exportFilePrefix-$i.zip"

Write-Host "Expanding archive ..."
Expand-Archive -Force -LiteralPath "$exportFilePrefix-$i.zip" -
DestinationPath "$exportFilePrefix-$i"
Remove-Item "$exportFilePrefix-$i.zip"

Get-ChildItem -Path "$exportFilePrefix-$i" -File -Name | ForEach-Object
{ $exportFileArray += "$exportFilePrefix-$i\" + $_ }
}

Write-Host "creating outputfile ..."
#write header to outputfile
Get-Content $exportFileArray[0] | Select-Object -First 1 | Out-File
"$exportFilePrefix.csv"

#write content to outputfile skipping header
ForEach ($exportFile in $exportFileArray) {
    Get-Content $exportFile | Select-Object -Skip 1 | Out-File -Append
"$exportFilePrefix.csv"
}

Compress-Archive -LiteralPath "$exportFilePrefix.csv" -DestinationPath
"$exportFilePrefix.zip"

Write-Host "cleaning up ..."

```

```
Remove-Item "$exportFilePrefix-*" -Recurse
Remove-Item "$exportFilePrefix.csv"
```

Bash 스크립트

```
#!/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"

```

ASO(집계 저장영역) 큐브에서 많은 수의 셀을 익스포트하려면 다음을 수행합니다.

1. PowerShell 또는 Bash 스크립트를 복사하여 파일 시스템에 저장합니다(예: ASOCellExport.ps1 또는 ASOCellExport.sh).
2. 스크립트 파일을 수정하고 매개변수 값을 설정합니다. 자세한 내용은 다음 테이블을 참조하십시오.

표 3-16 PowerShell 및 Bash 스크립트에 포함할 변수 값

변수	설명
user	<p><i>DOMAIN.USER</i> 형식으로 지정된 서비스 관리자의 도메인 및 사용자 이름입니다.</p> <p>예:</p> <p>Windows: \$user = 'exampleDomain.jDoe'</p> <p>Linux/UNIX: user = 'exampleDomain.jDoe'</p>
pass	<p>서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 위치. 암호화된 비밀번호 파일을 생성하는 방법에 대한 자세한 내용은 encrypt 명령을 참조하십시오.</p> <p>예:</p> <p>Windows: \$pass = 'Example'</p> <p>Linux/UNIX: pass = 'Example'</p>
serverURL	<p>Oracle Fusion Cloud Enterprise Performance Management 환경의 URL입니다.</p> <p>예:</p> <p>Windows: \$serverURL = 'https://example.oraclecloud.com'</p> <p>Linux/UNIX: serverURL = 'https://example.oraclecloud.com'</p>
applicationName	<p>Planning, Financial Consolidation and Close, Tax Reporting 또는 Enterprise Profitability and Cost Management 애플리케이션의 이름입니다.</p> <p>예:</p> <p>Windows: \$applicationName = 'Vision'</p> <p>Linux/UNIX: applicationName = 'Vision'</p>
cubeName	<p>애플리케이션에 있는 큐브의 이름.</p> <p>예:</p> <p>Windows: \$cubeName = 'VisASO'</p> <p>Linux/UNIX: cubeName = 'VisASO'</p>
splitDimension	<p>익스포트를 그룹으로 분할하는 데 해당 멤버가 사용되는 차원의 이름.</p> <p>예:</p> <p>Windows: \$splitDimension = 'Account'</p> <p>Linux/UNIX: splitDimension = 'Account'</p>
topLevelMemberForExport	<p>레벨 0 멤버 목록이 생성되는 차원 하위 계층의 멤버 이름.</p> <p>예:</p> <p>Windows: \$topLevelMemberForExport = 'Total Cash Flow'</p> <p>Linux/UNIX: topLevelMemberForExport = 'Total Cash Flow'</p>
exportJobName	<p>익스포트 데이터 유형의 기존 작업 이름. 이 작업에 지정된 설정은 스크립트에서 설정한 매개변수로 덮어씁니다.</p> <p>예:</p> <p>Windows: \$exportJobName = 'ASO Cell Export'</p> <p>Linux/UNIX: exportJobName = 'ASO Cell Export'</p>

표 3-16 (계속) PowerShell 및 Bash 스크립트에 포함할 변수 값

변수	설명
exportFilePrefix	익스포트 작업으로 생성된 파일을 고유하게 확인하기 위한 파일 이름 접두어. 예: Windows: \$exportFilePrefix = 'cashflow' Linux/UNIX: exportFilePrefix = 'cashflow'
columnMembers	익스포트에 포함할 멤버 열. 예: Windows: \$columnMembers = 'Period' Linux/UNIX: columnMembers = 'Period'
povMembers	익스포트에 포함할 POV. POV 멤버에는 다른 모든 차원이 포함되어야 하며 아래에 표시된 것처럼 함수를 포함할 수 있습니다. ILvl0Descendants(YearTotal), ILvl0Descendants(Year), ILvl0Descendants(Scenario), ILvl0Descendants(Version), ILvl0Descendants(P_TP), ILvl0Descendants(AltYear) 예: Windows: \$povMembers = 'YTD' Linux/UNIX: povMembers = 'YTD'
numberOfExportFiles	이 익스포트 작업에 대해 실행할 작업 수. 쿼리 제한으로 인해 익스포트가 계속 실패하는 경우 이 숫자를 늘리십시오. 예: Windows: \$numberOfExportFiles = 3 Linux/UNIX: numberOfExportFiles = 3

- Windows 스케줄러 또는 cron 작업을 사용하여 스크립트가 편리한 시간에 실행되도록 스케줄링합니다. 자세한 단계는 [스크립트 실행 자동화](#)를 참조하십시오.

애플리케이션으로 메타데이터 импорт

이러한 스크립트를 사용하여 파일에서 애플리케이션 메타데이터를 수동으로 импорт합니다.

이러한 스크립트는 다음 활동을 수행합니다.

- 환경에 로그인합니다.
- 메타데이터 파일을 업로드합니다.
- 작업을 사용하여 업로드된 파일에서 애플리케이션으로 메타데이터를 импорт합니다.
- 큐브를 새로 고칩니다.
- 로그아웃합니다.

Windows 샘플 스크립트

다음 스크립트를 복사하여 importMetadata.ps1을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -raw)
$username="$($inputproperties.username) "
```

```

$passwordfile="$(${inputproperties.passwordfile})"
$serviceURL="$(${inputproperties.serviceURL})"
$file1="$(${inputproperties.file1})"
$jobName="$(${inputproperties.jobName})"

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile ${file1}
epmautomate importmetadata ${jobName} ${file1}
epmautomate refreshcube
epmautomate logout

```

Linux/UNIX 샘플 스크립트

다음 스크립트를 복사하여 importMetadata.sh를 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```

#!/bin/bash
. ./input.properties
export JAVA_HOME=${javahome}
${epmautomatescript} login "${username}" "${passwordfile}" "${serviceURL}"
${epmautomatescript} uploadfile "${file1}"
${epmautomatescript} importmetadata "${jobName}" "${file1}"
${epmautomatescript} refreshcube
${epmautomatescript} logout

```

input.properties 파일 생성

다음 중 하나를 복사하고 해당 환경에 대한 정보로 업데이트하여 input.properties 파일을 생성합니다. importMetadata.ps1 또는 importMetadata.sh가 저장된 디렉토리에 파일을 저장합니다.

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

```

표 3-17 input.properties 매개변수

매개변수	설명
javahome	JAVA_HOME 위치입니다. Linux/UNIX에만 해당합니다.

표 3-17 (계속) input.properties 매개변수

매개변수	설명
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다. Linux/UNIX에만 해당합니다.
username	ID 도메인 관리자 역할도 가진 서비스 관리자의 사용자 이름입니다.
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
serviceURL	스냅샷을 생성하려는 소스 환경의 URL입니다.
File1	임포트할 메타데이터가 포함된 ZIP 파일의 이름입니다.
JobName	메타데이터를 임포트하는 데 사용할 작업입니다.

스크립트 실행

- 이전 섹션에서 스크립트를 복사하여 importMetadata.ps1 또는 importMetadata.sh를 생성합니다.
- input.properties 파일을 생성하고 importMetadata 스크립트가 있는 디렉토리에 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. [input.properties 파일 생성](#)을 참조하십시오.
이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 관리자 권한으로 실행 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
- 스크립트를 실행합니다.
 - Windows PowerShell:** importMetadata.ps1을 실행합니다.
 - Linux/UNIX:** ./importMetadata.sh를 실행합니다.

데이터 임포트, 계산 스크립트 실행 및 블록 저장영역 데이터베이스에서 집계 저장영역 데이터베이스로 데이터 복사

이러한 스크립트를 사용하여 파일에서 데이터를 임포트하고, 큐브를 새로고침하고, 비즈니스 규칙을 실행하여 큐브를 계산하고, 데이터를 ASO 큐브로 푸시합니다.

이러한 스크립트는 다음 작업을 수행합니다.

- 환경에 로그인합니다.
- data.csv 파일을 업로드합니다.
- 작업 loadingqldata를 사용하여 data.csv에서 애플리케이션으로 데이터를 임포트합니다.
- 큐브를 새로 고칩니다.
- 비즈니스 규칙을 실행하여 데이터를 변환합니다.
- 작업을 사용하여 데이터를 집계 저장영역 데이터베이스에 푸시합니다.
- 로그아웃합니다.

Windows 샘플 스크립트

다음 스크립트를 복사하여 importDataPlus.ps1을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
$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)"
$params1Key="$($inputproperties.params1Key)"
$params1Value="$($inputproperties.params1Value)"
$params2Key="$($inputproperties.params2Key)"
$params2Value="$($inputproperties.params2Value)"
$clearData="$($inputproperties.clearData)"

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile ${file1}
epmautomate importdata ${importDataJobName} ${file1}
epmautomate refreshcube
epmautomate runbusinessrule ${businessRuleName} ${params1Key}=${params1Value} ${params2Key}=${params2Value}
epmautomate runplantypemap ${planTypeMapName} clearData=${clearData}
epmautomate logout
```

Linux/UNIX 샘플 스크립트

다음 스크립트를 복사하여 importDataPlus.ps1을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
#!/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}" "${params1Key}=${params1Value}" "${params2Key}=${params2Value}"
${epmautomatescript} runplantypemap "${planTypeMapName}" clearData=${clearData}
${epmautomatescript} logout
```

input.properties 파일 생성

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
```

표 3-18 input.properties 매개변수

매개변수	설명
javahome	JAVA_HOME 위치입니다. Linux/UNIX에만 해당합니다.
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다. Linux/UNIX에만 해당합니다.
username	ID 도메인 관리자 역할도 가진 서비스 관리자의 사용자 이름입니다.
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
serviceURL	스냅샷을 생성하려는 소스 환경의 URL입니다.
File1	데이터를 애플리케이션에 로드할 소스 импорт 파일입니다.
iimportDataJobName	데이터를 импорт하는 데 사용할 작업의 이름입니다.
businessRuleName	им포트한 데이터에서 실행할 비즈니스 규칙입니다.
planTypeMapName	BSO 데이터베이스에서 ASO 데이터베이스로 또는 BSO 데이터베이스에서 다른 BSO 데이터베이스로 데이터를 복사하는 데 사용할 계획 유형 맵입니다.
param1Key	비즈니스 규칙을 실행하기 위한 런타임 프롬프트 1입니다.
param1Value	런타임 프롬프트 1의 값입니다.
param2Key	비즈니스 규칙을 실행하기 위한 런타임 프롬프트 2입니다.
param2Value	런타임 프롬프트 2의 값입니다.
clearData	수신 데이터베이스의 데이터를 삭제할지 여부를 나타냅니다. 데이터를 유지하려면 false를 지정합니다.

스크립트 실행

1. 이전 섹션에서 스크립트를 복사하여 `importDataPlus.ps1` 또는 `importDataPlus.sh`를 생성합니다.
2. `input.properties` 파일을 생성하고 `importDataPlus` 스크립트가 있는 디렉토리에 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. [input.properties 파일 생성](#)를 참조하십시오.
이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 **관리자 권한으로 실행** 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
3. 스크립트를 실행합니다.
 - **Windows PowerShell:** `importDataPlus.ps1`을 실행합니다.
 - **Linux/UNIX:** `./importDataPlus.sh`를 실행합니다.

메타데이터와 데이터를 익스포트 및 다운로드

이러한 스크립트를 사용하여 애플리케이션 메타데이터 및 데이터를 익스포트한 다음, 익스포트 파일을 로컬 디렉토리로 다운로드합니다.

이러한 스크립트는 다음 활동을 완료합니다.

- 환경에 로그인합니다.
- 지정된 작업을 사용하여 메타데이터를 zip 파일로 익스포트합니다.
- 지정된 작업을 사용하여 애플리케이션 데이터를 zip 파일로 익스포트합니다.
- 인박스/아웃박스의 콘텐츠를 나열합니다.
- 익스포트된 데이터 파일을 로컬 컴퓨터에 다운로드합니다.
- 사인아웃

Windows 샘플 스크립트

다음 스크립트를 복사하여 `exportDownloadMetadataAndData.ps1`을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
$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
```

Linux/UNIX 샘플 스크립트

다음 스크립트를 복사하여 `exportDownloadMetadataAndData.sh`를 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
#!/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
```

등록정보 파일 생성

다음 중 하나를 복사하고 해당 환경에 대한 정보로 업데이트하여 `input.properties` 파일을 생성합니다. `exportDownloadMetadataAndData.ps1` 또는 `exportDownloadMetadataAndData.sh`가 저장된 디렉토리에 파일을 저장합니다.

Windows

```
username=exampleAdmin
passwordfile=examplePassword.epw
serviceURL=exampleURL
exportFile1=FILE_NAME1.zip
exportFile2=FILE_NAME2.zip
exportMetaJobName=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
exportMetaJobName=METADATA_EXPORT_JOB_NAME
exportDataJobName=DATA_EXPORT_JOB_NAME
```

표 3-19 input.properties 매개변수

매개변수	설명
javahome	JAVA_HOME 위치입니다. Linux/UNIX에만 해당합니다.
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다. Linux/UNIX에만 해당합니다.
username	ID 도메인 관리자 역할도 가진 서비스 관리자의 사용자 이름입니다.

표 3-19 (계속) input.properties 매개변수

매개변수	설명
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
serviceURL	스냅샷을 생성하려는 소스 환경의 URL입니다.
exportFile1	메타데이터를 익스포트할 타겟 파일의 이름입니다.
exportFile2	데이터를 익스포트할 타겟 파일의 이름입니다.
exportDataJobName1	메타데이터를 익스포트하는 데 사용할 작업입니다.
exportDataJobName2	데이터를 익스포트하는 데 사용할 작업입니다.

스크립트 실행

- 이전 섹션에서 스크립트를 복사하여 `exportDownloadMetadataAndData.ps1` 또는 `exportDownloadMetadataAndData.sh`를 생성합니다.
- `input.properties` 파일을 생성하고 `exportDownloadMetadataAndData` 스크립트가 있는 디렉토리에 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. [등록정보 파일 생성](#)을 참조하십시오.
이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 **관리자 권한으로 실행** 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
- 스크립트를 실행합니다.
 - Windows PowerShell:** `exportDownloadMetadataAndData.ps1`을 실행합니다.
 - Linux/UNIX:** `./exportDownloadMetadataAndData.sh`를 실행합니다.

애플리케이션 데이터 익스포트 및 다운로드

이러한 스크립트를 사용하여 애플리케이션 데이터를 익스포트한 다음, 로컬 디렉토리로 다운로드합니다.

이러한 스크립트는 다음 작업을 수행합니다.

- 환경에 로그인합니다.
- 지정한 작업을 사용하여 두 개의 데이터 세트를 백업합니다.
- 익스포트된 데이터 파일을 다운로드합니다.
- 로그아웃합니다.

Windows 샘플 스크립트

다음 스크립트를 복사하여 `exportDownloadData.ps1`을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -raw)
$username="$($inputproperties.username)"
$passwordfile="$($inputproperties.passwordfile)"
$serviceURL="$($inputproperties.serviceURL)"
$exportFile1="$($inputproperties.exportFile1)"
$exportFile2="$($inputproperties.exportFile2)"
$exportDataJobName1="$($inputproperties.exportDataJobName1)"
$exportDataJobName2="$($inputproperties.exportDataJobName2)"
```

```

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate exportdata ${exportDataJobName1} ${exportFile1}
epmautomate exportdata ${exportDataJobName2} ${exportFile2}
epmautomate listfiles
epmautomate downloadfile ${exportFile1}
epmautomate downloadfile ${exportFile2}
epmautomate logout

```

Linux/UNIX 샘플 스크립트

다음 스크립트를 복사하여 `exportDownloadData.sh`를 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```

#!/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

```

input.properties 파일 생성

다음 중 하나를 복사하고 해당 환경에 대한 정보로 업데이트하여 `input.properties` 파일을 생성합니다. `exportDownloadData.ps1` 또는 `exportDownloadData.sh`가 저장된 디렉토리에 파일을 저장합니다.

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

```

표 3-20 input.properties 매개변수

매개변수	설명
javahome	JAVA_HOME 위치입니다. Linux/UNIX에만 해당합니다.
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다. Linux/UNIX에만 해당합니다.
username	ID 도메인 관리자 역할도 가진 서비스 관리자의 사용자 이름입니다.
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
serviceURL	스냅샷을 생성하려는 소스 환경의 URL입니다.
exportFile1 및 exportFile2	데이터를 익스포트할 타겟 파일의 이름입니다.
exportDataJobName1 및 exportDataJobName2	데이터를 익스포트하는 데 사용할 작업입니다.

스크립트 실행

- 이전 섹션에서 스크립트를 복사하여 exportDownloadData.ps1 또는 exportDownloadData.sh를 생성합니다.
- input.properties 파일을 생성하고 exportDownloadData 스크립트가 있는 디렉토리에 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. [테이블 1](#)를 참조하십시오. 이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 **관리자 권한으로 실행** 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
- 스크립트를 실행합니다.
 - Windows PowerShell:** exportDownloadData.ps1을 실행합니다.
 - Linux/UNIX:** ./exportDownloadData.sh를 실행합니다.

애플리케이션 감사 레코드 아카이브 자동화

이 섹션의 Windows 및 Linux 스크립트를 사용하여 애플리케이션 감사 데이터를 로컬 컴퓨터로 익스포트하고 아카이브하는 프로세스를 자동화합니다.

애플리케이션 감사 데이터는 365일 동안만 유지됩니다. 365일보다 오래된 과거 감사 데이터가 손실되지 않도록 하려면 이러한 스크립트를 사용자정의하고 180일마다 한 번씩 또는 데이터 보존 정책에서 요구하는 대로 스크립트를 실행합니다.

Note

이러한 스크립트는 로컬 저장영역에 데이터를 아카이브하도록 맞춤 설정되었습니다. 익스포트한 감사 데이터 파일을 네트워크 저장영역 또는 저장영역 클라우드(예: Oracle Object Storage)에 아카이브하도록 수정할 수 있습니다.

Table 3-21 매개변수 및 해당 값

매개변수	값
url	환경의 URL입니다. 예: <ul style="list-style-type: none"> Windows: set url=https://example-epmidm.epm.usphoenix-1.ocs.oraclecloud.com/epmcloud Linux: url=https://example-epmidm.epm.usphoenix-1.ocs.oraclecloud.com/epmcloud
user	감사 데이터를 다운로드하기 위해 환경에 로그인할 서비스 관리자의 사용자 이름입니다. 예: <ul style="list-style-type: none"> Windows: set user=ExampleAdmin Linux: user=ExampleAdmin
password	서비스 관리자의 비밀번호(권장되지 않음) 또는 암호화된 비밀번호 파일의 이름 및 위치입니다. 암호화된 비밀번호 파일을 생성하는 방법에 대한 자세한 내용은 encrypt 명령을 참조하십시오. 예: <ul style="list-style-type: none"> Windows: set password="C:\mySecuredir\password.epw" Linux: password="/home/user1/mySecuredir/password.epw"
AuditFileName	감사 데이터 파일의 이름입니다. 이 파일이 고유하도록 스크립트는 감사 데이터 익스포트 타임스탬프를 파일 이름에 추가합니다. 예: <ul style="list-style-type: none"> Windows: set AuditFileName=AuditData Linux: AuditFileName=AuditData
NumberOfBackups	저장영역에 유지할 백업 파일 수입니다. 기본값은 10이며 그 이후에는 필요에 따라 가장 오래된 백업이 대체됩니다. 예: <ul style="list-style-type: none"> Windows: set NumberOfBackups=20 Linux: NumberOfBackups=20
Linux 스크립트만 해당	
epmautomatescript	EPM Automate가 설치된 위치입니다. 예: /home/user1/epmautomate/bin/epmautomate.sh
javahome	JAVA_HOME 위치입니다. 예: /home/user1/jdk_17.0.12/

Windows 스크립트

배치 파일을 생성합니다. 예를 들어 AuditExport.bat에는 감사 데이터를 로컬 컴퓨터로 자동으로 익스포트하고 다운로드하는 다음과 유사한 스크립트가 포함됩니다.

```
@echo off
rem Sample script to download and maintain 10 audit data backups
rem Update the following parameters

SET url=https://example.oraclecloud.com
SET user=ServiceAdmin
SET password=Example.epw
```

```

SET AuditFileName="AuditBackup"
SET NumberOfBackups=10

rem EPM Automate commands
call epmautomate login %user% %password% %url%
    IF %ERRORLEVEL% NEQ 0 goto :ERROR
    call epmautomate exportAppAudit %AuditFileName% nDays=180
    IF %ERRORLEVEL% NEQ 0 goto :ERROR
    call epmautomate downloadfile %AuditFileName%.zip
    IF %ERRORLEVEL% NEQ 0 goto :ERROR
    call epmautomate logout
    IF %ERRORLEVEL% NEQ 0 goto :ERROR

rem Rename downloaded audit data backup, keep the last 10 backups
Set Timestamp=%date:~4,2%_%date:~7,2%_%date:~10,2%%
Set Second=%time:~0,2%%time:~3,2%
ren %AuditFileName%.zip %AuditFileName%_%Timestamp%_%Second%.zip
SET Count=0
FOR %%A IN (%AuditFileName%*.*) DO SET /A Count += 1
IF %Count% gtr %NumberOfBackups% FOR %%A IN (%AuditFileName%*.*) DO del "%%A"
&& GOTO EOF
:EOF

echo Scheduled Task Completed successfully
exit /b %errorlevel%
:ERROR
echo Failed with error #%errorlevel%.
exit /b %errorlevel%

```

Linux 스크립트

셸 스크립트를 생성합니다. 예를 들어 AuditExport.sh에는 감사 데이터를 로컬 컴퓨터로 자동으로 익스포트하고 다운로드하는 다음과 유사한 스크립트가 포함됩니다.

```

#!/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/jdk_17.0.12/

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 --

```

데이터 파일을 환경으로 업로드 및 데이터 로드 규칙 실행

이러한 스크립트를 사용하여 파일을 환경으로 업로드한 다음, 데이터 규칙을 실행하여 파일의 데이터를 애플리케이션으로 импорт합니다.

필수 조건

- Data Management의 정의는 다음과 같습니다.
 - VisionActual.이라는 데이터 로드 규칙 정의 데이터 규칙에서 입력 파일의 파일 경로를 지정하지 않는다고 가정합니다.
 - Mar-15에서 Jun-15 기간 정의
- 데이터가 포함된 올바른 형식의 데이터 파일(GLActual.dat)

데이터를 가져오고 데이터 로드 규칙을 실행하려면 다음 단계를 완료하는 명령을 실행합니다.

- 환경에 사인인합니다.
- Mar-15에서 Jun-15 기간의 데이터가 포함된 GLActual.dat 파일을 Data Management 폴더 inbox/Vision으로 업로드
- 데이터 로드 규칙 VisionActual, 시작 기간 Mar-15, 종료 기간 Jun-15, импорт 모드 REPLACE를 사용하여 GLActual.dat에서 Data Management로 데이터 импорт
- STORE_DATA 옵션으로 데이터를 익스포트하여 Data Management 스테이지 테이블의 데이터를 기존 애플리케이션 데이터와 병합

- 사인아웃

Windows 샘플 스크립트

다음 스크립트를 복사하여 runDataLoadRule.ps1을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -raw)
$username="$($inputproperties.username)"
$passwordfile="$($inputproperties.passwordfile)"
$serviceURL="$($inputproperties.serviceURL)"
$dataFile="$($inputproperties.dataFile)"
$dataRuleName="$($inputproperties.dataRuleName)"
$startPeriod="$($inputproperties.startPeriod)"
$endPeriod="$($inputproperties.endPeriod)"
$importMode="$($inputproperties.importMode)"
$exportMode="$($inputproperties.exportMode)"

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile ${datafile} ${dataFileUploadLocation}
epmautomate rundatarule ${dataRuleName} ${startPeriod} ${endPeriod} $
{importMode} ${exportMode} ${dataFileUploadLocation}/${dataFile}
epmautomate logout
```

Linux/UNIX 샘플 스크립트

다음 스크립트를 복사하여 runDataLoadRule.sh를 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
#!/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
```

input.properties 파일 생성

다음 중 하나를 복사하고 해당 환경에 대한 정보로 업데이트하여 input.properties 파일을 생성합니다. runDataLoadRule.ps1 또는 runDataLoadRule.sh가 저장된 디렉토리에 파일을 저장합니다.

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
```

표 3-22 input.properties 매개변수

매개변수	설명
javahome	JAVA_HOME 위치입니다. Linux/UNIX에만 해당합니다.
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다. Linux/UNIX에만 해당합니다.
username	ID 도메인 관리자 역할도 가진 서비스 관리자의 사용자 이름입니다.
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
serviceURL	스냅샷을 생성하려는 소스 환경의 URL입니다.
dataFile	데이터 규칙을 사용하여 임포트할 데이터가 포함된 파일입니다.
dataFileUploadLocation	데이터 파일을 업로드할 타겟 위치입니다.
dataRuleName	Data Integration에 정의된 데이터 로드 규칙의 이름입니다.
startPeriod	데이터를 로드할 첫번째 기간입니다. 이 기간 이름은 Data Integration 기간 매핑에 정의되어 있어야 합니다.
endPeriod	복수 기간 데이터 로드의 경우 데이터를 로드할 마지막 기간입니다. 단일 기간 로드의 경우 시작 기간과 동일한 기간을 사용합니다. 이 기간 이름은 Data Integration 기간 매핑에 정의되어 있어야 합니다.
importMode	Data Integration으로 데이터를 임포트하기 위한 모드입니다. APPEND, REPLACE 또는 RECALCULATE를 사용합니다. 스테이지 테이블로 데이터 임포트를 건너뛰려면 NONE을 사용합니다.
exportMode	애플리케이션으로 데이터를 익스포트하기 위한 모드입니다. Data Integration을 사용합니다. STORE_DATA, ADD_DATA, SUBTRACT_DATA 또는 REPLACE_DATA를 사용합니다. Data Integration에서 애플리케이션으로 데이터 익스포트를 건너뛰려면 NONE을 사용합니다.

주

Financial Consolidation and Close는 MERGE 및 NONE 모드만 지원합니다.

스크립트 실행

1. 이전 섹션에서 스크립트를 복사하여 `runDataLoadRule.ps1` 또는 `runDataLoadRule.sh`를 생성합니다.
2. `input.properties` 파일을 생성하고 `runDataLoadRule` 스크립트가 있는 디렉토리에 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. [input.properties 파일 생성](#)를 참조하십시오.
이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 **관리자 권한으로 실행** 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
3. 스크립트를 실행합니다.
 - **Windows PowerShell:** `runDataLoadRule.ps1`을 실행합니다.
 - **Linux/UNIX:** `./runDataLoadRule.sh`를 실행합니다.

일일 데이터 통합 자동화

이 시나리오에서는 샘플 스크립트를 사용하여 정기적인 데이터 통합을 자동화하는 방법을 살펴봅니다.

아래와 유사한 스크립트가 포함된 배치(.bat) 또는 셸(.sh) 파일을 생성하여 데이터 통합 관련 활동을 자동화합니다. 아래의 Windows용 샘플 스크립트에서는 다음 활동을 완료하여 일일 애플리케이션 데이터 통합을 자동화합니다.

- 환경에 사인인합니다.
- `DailyPlanData`가 있는 경우 삭제
- `DailyPlanData`를 서비스에 업로드
- `Plan1` 계획 유형에서 `Clear Plan Targets` 비즈니스 규칙 실행
- `LoadDailyPlan` 작업 이름을 사용하여 데이터 импорт
- `Balance Sheet - Plan` 비즈니스 규칙 실행
- `Allocate Plan Targets` 비즈니스 규칙 실행
- `DailyTarget.zip`이 있는 경우 삭제
- `ExportDailyTarget` 작업 이름을 사용하여 `DailyTarget.zip`에 데이터 익스포트
- `DailyTarget.zip`을 서버에 다운로드하고 타임스탬프 추가
- 환경에서 사인아웃합니다.

① 주

이 스크립트를 사용자의 용도에 맞게 고치는 경우 `SET url` 및 `SET user` 매개변수의 값을 수정합니다. 또한 요구사항에 맞게 `dataimportfilename`, `dataexportfilename`, `importdatajobname`, `exportdatajobname`, `br_clear`, `br_calculatebalancesheet` 및 `br_allocatetarget` 매개변수의 값을 수정할 수 있습니다.

Windows 태스크 스케줄러를 사용하여 스크립트 스케줄을 예약하는 방법에 대한 자세한 내용은 [스크립트 실행 자동화](#)를 참조하십시오.

```
@echo off

rem Sample Script to demonstrate daily data integration with
rem Cloud EPM 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%

```

Account Reconciliation의 샘플 시나리오

참조:

- [기간에 미리 형식 지정된 잔액 로드](#)
이러한 스크립트를 사용하여 매핑된 데이터를 업로드된 파일에서 Account Reconciliation 환경으로 임포트합니다.
- [백업 스냅샷 업로드 및 임포트](#)
이러한 스크립트를 사용하여 백업 스냅샷을 Account Reconciliation 환경으로 업로드 및 임포트합니다.
- [오래된 일치 트랜잭션 아카이브 및 아카이브된 트랜잭션 비우기](#)
이 섹션의 스크립트를 사용하여 지정된 기간의 일수보다 오래되거나 일수가 같은 일치 트랜잭션을 지원 및 조정 세부정보와 함께 아카이브한 다음, Account Reconciliation에서 아카이브된 트랜잭션을 비웁니다. 아카이브된 일치 트랜잭션은 ZIP 파일로 기록됩니다.

기간에 미리 형식 지정된 잔액 로드

이러한 스크립트를 사용하여 매핑된 데이터를 업로드된 파일에서 Account Reconciliation 환경으로 임포트합니다.

Windows 샘플 스크립트

다음 스크립트를 복사하여 runPreformattedBalances.ps1이라는 파일을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```

$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -raw)
$username="$($inputproperties.username)"
$passwordfile="$($inputproperties.passwordfile)"
$serviceURL="$($inputproperties.serviceURL)"
$dataFile="$($inputproperties.dataFile)"
$period="$($inputproperties.period)"
$balanceType="$($inputproperties.balanceType)"
$currencyBucket="$($inputproperties.currencyBucket)"

```

```

$elements=$dataFile.split('/')
$dataFileName=$elements[-1]

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile ${dataFile}
epmautomate importpremappedbalances ${period} ${dataFileName} ${balanceType} $
{currencyBucket}
epmautomate deletefile ${dataFileName}
epmautomate logout

```

Linux/UNIX 샘플 스크립트

다음 스크립트를 복사하여 runPreformattedBalances.sh라는 파일을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```

#!/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

```

샘플 input.properties 파일

runPreformattedBalances 스크립트를 실행하려면 input.properties 파일을 생성하고 해당 환경에 대한 정보로 업데이트합니다. runPreformattedBalances.sh 또는 runPreformattedBalances.ps1이 저장된 디렉토리에 파일을 저장합니다.

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
```

표 3-23 input.properties 매개변수

매개변수	설명
javahome	JAVA_HOME 위치입니다. Linux/UNIX에만 해당합니다.
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다. Linux/UNIX에만 해당합니다.
username	서비스 관리자.의 사용자 이름
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
serviceURL	미리 형식 지정된 잔액을 로드할 타겟 애플리케이션을 호스트하는 환경의 URL입니다.
dataFile	애플리케이션에 로드하려는 미리 형식 지정된 잔액(일반적으로 총계정원장에서 생성됨)이 포함된 CSV 파일입니다. 이 파일은 uploadFile 명령을 사용하여 해당 환경에 이미 업로드되었어야 합니다.
period	미리 형식 지정된 잔액을 업로드할 타겟 조정 기간입니다.
balanceType	dataFile에 포함된 미리 형식 지정된 잔액의 유형입니다.
currencyBucket	미리 형식 지정된 잔액에 대한 통화 버킷입니다.

스크립트 실행

- 이전 섹션에서 스크립트를 복사하여 runPreformattedBalances.ps1 또는 runPreformattedBalances.sh를 생성합니다.
- Windows 및 Linux/UNIX에만 해당:**
 - input.properties 파일을 생성하고 runPreformattedBalances 스크립트가 있는 디렉토리에 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. [테이블 1](#)를 참조하십시오.
이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 **관리자 권한으로 실행** 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
 - 스크립트를 실행합니다.
 - Windows PowerShell:** runPreformattedBalances.ps1을 실행합니다.
 - Linux/UNIX:** ./runPreformattedBalances.sh를 실행합니다.

백업 스냅샷 업로드 및 импорт

이러한 스크립트를 사용하여 백업 스냅샷을 Account Reconciliation 환경으로 업로드 및 импорт합니다.

Windows 샘플 스크립트

다음 스크립트를 복사하여 importBackupSnapshot.ps1이라는 파일을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -raw)
$username="$($inputproperties.username)"
$passwordfile="$($inputproperties.passwordfile)"
$serviceURL="$($inputproperties.serviceURL)"
```

```

$snapshotName="${inputproperties.snapshotName}"
$userPassword="${inputproperties.userPassword}"

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile ${snapshotName}.zip
epmautomate importsnapshot ${snapshotName} "userPassword=${userPassword}"
epmautomate deletefile ${snapshotName}.zip
epmautomate logout

```

Linux/UNIX 샘플 스크립트

다음 스크립트를 복사하여 importBackupSnapshot.sh라는 파일을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```

#!/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

```

샘플 input.properties 파일

importBackupSnapshot 스크립트를 실행하려면 input.properties 파일을 생성하고 해당 환경에 대한 정보로 업데이트합니다. importBackupSnapshot.sh 또는 importBackupSnapshot.ps1이 저장된 디렉토리에 파일을 저장합니다.

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

```

표 3-24 input.properties 매개변수

매개변수	설명
javahome	JAVA_HOME 위치입니다. Linux/UNIX에만 해당합니다.

표 3-24 (계속) input.properties 매개변수

매개변수	설명
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다. Linux/UNIX에만 해당합니다.
username	서비스 관리자의 사용자 이름
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
serviceURL	스냅샷을 임포트하려는 환경의 URL입니다.
snapshotName	아티팩트 및 데이터를 임포트할 소스 스냅샷의 이름입니다. 이 스냅샷은 uploadFile 명령을 사용하여 해당 환경에 이미 업로드되었어야 합니다.
userPassword	이 스냅샷 임포트의 결과로 ID 도메인에 생성된 모든 새 사용자에게 지정해야 하는 기본 비밀번호.

스크립트 실행

- 이전 섹션에서 스크립트를 복사하여 importBackupSnapshot.ps1 또는 importBackupSnapshot.sh를 생성합니다.
- input.properties 파일을 생성하고 runPreformattedBalances 스크립트가 있는 디렉토리에 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. [샘플 input.properties 파일](#)을 참조하십시오.
이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 관리자 권한으로 실행 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
- 스크립트를 실행합니다.
 - Windows PowerShell:** importBackupSnapshot.ps1을 실행합니다.
 - Linux/UNIX:** ./importBackupSnapshot.sh를 실행합니다.

오래된 일치 트랜잭션 아카이브 및 아카이브된 트랜잭션 비우기

이 섹션의 스크립트를 사용하여 지정된 기간의 일수보다 오래되거나 일수가 같은 일치 트랜잭션을 지원 및 조정 세부정보와 함께 아카이브한 다음, Account Reconciliation에서 아카이브된 트랜잭션을 비웁니다. 아카이브된 일치 트랜잭션은 ZIP 파일로 기록됩니다.

스크립트 작동 방법

- input.properties 파일의 정보를 사용하여 환경에 로그인합니다.
- 다음 archiveTmTransactions 명령을 실행하여 아카이브를 생성합니다. 생성되는 ZIP 파일 및 로그 파일은 기본 이름 Archive_Transactions_INTERCO_JOB_ID.zip 및 Archive_Transactions_INTERCO_JOB_ID.log를 사용합니다.
epmautomate archiveTmTransactions INTERCO 365 filterOperator=contains
filterValue=14001
input.properties 파일을 수정하여 명령 매개변수를 변경할 수 있습니다.
- 로그 파일 및 아카이브된 트랜잭션이 포함된 .ZIP 파일을 로컬 컴퓨터에 다운로드합니다. 이 스크립트는 일치 트랜잭션을 찾을 수 없는 경우 오류 메시지를 표시합니다.
- 아카이브된 트랜잭션이 포함된 .ZIP 파일을 Oracle Object Store로 복사합니다.
- purgeArchivedTmTransactions 명령을 실행하여(archiveTmTransactions 작업의 작업 ID와 함께) 아카이브된 일치 트랜잭션을 애플리케이션에서 삭제합니다.

스크립트 실행

1. `input.properties` 파일을 생성하고 해당 환경에 대한 정보로 업데이트합니다. 로컬 디렉토리에 파일을 저장합니다. 본 설명에서는 이 디렉토리를 `parentsnapshotdirectory`라고 합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. 이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 **관리자 권한으로 실행** 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
2. `transaction_match.ps1`(Windows PowerShell) 또는 `transaction_match.sh`(Linux/UNIX) 스크립트를 생성하고 `input.properties`가 있는 `parentsnapshotdirectory`에 저장합니다.
3. 스크립트를 실행합니다.
 - Linux/UNIX: `./transaction_match.sh`를 실행합니다.
 - Windows PowerShell: `transaction_match.ps1`을 실행합니다.

`input.properties` 스크립트 생성

다음 스크립트를 복사하고 업데이트하여 `input.properties`를 생성합니다.

```
javahome=JAVA_HOME
epmautomatescript=EPM_AUTOMATE_LOCATION
epmusername=exampleAdmin1
epmpassword=examplePassword1.epw
epmurl=exampleURL1
objectstorageusername=exampleAdmin2
objectstoragepassword=examplePassword2
objectstorageurl=exampleURL2
matchtype=INTERCO
age=365
filteroperator=contains
filtervalues=FilterValue=14001
proxyserverusername=myProxyserver
proxyserverpassword=myProxyserver_pwd
proxyserverdomain=myProxyDomain
```

① 주

Windows에만 해당: `input.properties` 파일에서 다음 등록정보를 제거합니다.

- `javahome=JAVA_HOME`
- `epmautomatescript=EPM_AUTOMATE_LOCATION`

Windows 네트워크 환경에 대해 `authentication at proxy server`가 사용으로 설정되지 않은 경우 `input.properties` 파일에서 이러한 등록정보를 제거합니다.

- `proxyserverusername`
- `proxyserverpassword`
- `proxyserverdomain`

표 3-25 input.properties 매개변수

매개변수	설명
javahome	<p>EPM Automate에서 사용하는 JDK가 설치된 디렉토리입니다. Windows 버전의 input.properties에서 이 항목을 삭제합니다.</p> <p>예: javahome=./home/JDK/bin</p>
epmautomatescript	<p>EPM Automate가 설치된 디렉토리입니다. Windows 버전의 input.properties에서 이 항목을 삭제합니다.</p> <p>예:epmautomatescript=./home/utills/EPMAutomate/bin</p>
epmusername	<p>일치 트랜잭션을 아카이브할 권한이 부여된 서비스 관리자, 고급 사용자, 사용자 또는 뷰어의 사용자 이름입니다.</p> <p>예: epmusername=ServiceAdmin</p>
epmuserpassword	<p>epmusername으로 확인된 사용자의 암호화된 비밀번호 파일입니다.</p> <p>예: epmpassword=myPwd.epw</p>
epmurl	<p>일치 트랜잭션을 아카이브할 환경의 URL입니다.</p> <p>예: epmurl=https://test-cloudpln.pbcs.us1.oraclecloud.com</p>
objectstorageusername	<p>Oracle Object Storage Cloud에 쓰는 데 필요한 액세스 권한을 가진 사용자의 ID입니다.</p> <p>페더레이션된 ID 제공자에서 생성된 사용자의 경우 사용자의 전체 이름을 지정합니다(예: exampleIdP/jdoe 또는 exampleIdP/john.doe@example.com, 설명: exampleIdP는 페더레이션된 ID 제공자의 이름). 다른 사용자의 경우 사용자 ID를 지정합니다.</p> <p>예: epmusername=myIdP/jdoe</p>
objectstoragepassword	<p>objectstorageusername에서 확인된 사용자에 연계된 Swift 비밀번호 또는 인증 토큰입니다. 이 비밀번호는 사용자가 Object Storage Console에 로그인하는 데 사용하는 비밀번호와 다릅니다. 인증 토큰은 Swift 클라이언트로 인증하는 경우와 같이 타사 API로 인증하는 데 사용하는 Oracle에서 생성된 토큰입니다. 이 토큰 생성에 대한 지침은 <i>Oracle Cloud Infrastructure 설명서</i>에서 인증 토큰을 생성하려면을 참조하십시오.</p> <p>예: objectstoragepassword=jDoe_PWD</p>
objectstorageurl	<p>선택적 객체 이름이 추가된 Oracle Object Storage Cloud 버킷의 URL입니다.</p> <p>예: objectstorageurl=https://swiftobjectstorage.region_identifier.oraclecloud.com/v1/namespace/MT_Archives/2023_archives</p>
matchtype	<p>일치 트랜잭션을 아카이브해야 하는 소스 일치 유형의 식별자(TextID)입니다.</p> <p>예: matchtype=cashrecon</p>
age	<p>트랜잭션이 일치한 이후 경과한 일수입니다. 이 값의 일수보다 오래되거나 일수가 같은 일치 트랜잭션은 아카이브됩니다.</p> <p>예: age=180</p>
filteroperator	<p>아카이브할 일치 트랜잭션이 포함된 계정을 확인하기 위한 필터 조건입니다. 다음 중 하나여야 합니다. equals, not_equals, starts_with, ends_with, contains 또는 not_contains. 이 값과 filterValue 값을 조합하여 일치 트랜잭션을 아카이브할 소스 계정을 확인합니다.</p> <p>예: filteroperator=not_equals</p>

표 3-25 (계속) input.properties 매개변수

매개변수	설명
filtervalues	아카이브할 트랜잭션을 확인하는 하나 이상의 필터 값입니다. equals 또는 not_equals를 filterOperator로 지정하면 공백으로 구분된 목록을 사용하여 여러 값을 지정할 수 있습니다. 여러 값이 지정되면 필터 연산자 및 필터 값 조합과 일치하는 계정의 트랜잭션이 아카이브용으로 선택됩니다. 예: filterValue=101-120 filterValue=140-202
proxyserverusername	인터넷에 대한 액세스를 제어하는 프록시 서버를 통해 보안 세션을 인증할 사용자 이름입니다. 예: proxyserverusername=myProxyserver
proxyserverpassword	프록시 서버를 통해 사용자를 인증할 비밀번호입니다. 예: proxyserverpassword=myProxyserver_pwd
proxyserverdomain	프록시 서버에 대해 정의된 도메인의 이름입니다. 예: proxyserverdomain=myProxyDomain

Windows 스크립트

다음 스크립트를 복사하여 transaction_match.ps1을 생성합니다. input.properties가 저장된 폴더에 파일을 저장합니다.

```
# Archive and Purge Transaction Matching PowerShell script

$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -raw)
$epmusername="$($inputproperties.epmusername)"
$epmpassword="$($inputproperties.epmpassword)"
$epmurl="$($inputproperties.epmurl)"
$objectstorageusername="$($inputproperties.objectstorageusername)"
$objectstoragepassword="$($inputproperties.objectstoragepassword)"
$objectstorageurl="$($inputproperties.objectstorageurl)"
$matchtype="$($inputproperties.matchtype)"
$age="$($inputproperties.age)"
$filteroperator="$($inputproperties.filteroperator)"
$filtervalues="$($inputproperties.filtervalues)"
$proxyserverusername="$($inputproperties.proxyserverusername)"
$proxyserverpassword="$($inputproperties.proxyserverpassword)"
$proxyserverdomain="$($inputproperties.proxyserverdomain)"

echo "Running processes to archive and purge transaction matching
transactions ..."
if (${proxyserverusername}) {
    echo "Running epmautomate login ${epmusername} ${epmpassword} ${epmurl} $
{proxyserverusername} ${proxyserverpassword} ${proxyserverdomain}"
    epmautomate login ${epmusername} ${epmpassword} ${epmurl} $
{proxyserverusername} ${proxyserverpassword} ${proxyserverdomain}
} else {
    echo "Running epmautomate login ${epmusername} ${epmpassword} ${epmurl}"
    epmautomate login ${epmusername} ${epmpassword} ${epmurl}
}
echo "Running epmautomate archiveTmTransactions \"${matchtype}\" ${age}
filterOperator=${filteroperator} ${filtervalues}"
epmautomate archiveTmTransactions "${matchtype}" "${age}" "filterOperator=$
{filteroperator}" "${filtervalues}" > ./outfile.txt.tmp
```

```

$jobIdLine=Select-String -Path "outfile.txt.tmp" -Pattern "Job
ID"; $jobIdLine=($jobIdLine -split ":-2); $jobid=($jobIdLine -split " ")
[1];
$logfile="Archive_Transactions_${matchtype}_${jobid}.log"
$transactionsfile="Archive_Transactions_${matchtype}_${jobid}.zip"
epmautomate listfiles > ./files.txt.tmp
$transactionslogfound=Select-String -Path "./files.txt.tmp" -Pattern "$
{logfile}"
$transactionsfilefound=Select-String -Path "./files.txt.tmp" -Pattern "$
{transactionsfile}"
rm ./outfile.txt.tmp
rm ./files.txt.tmp

if (${transactionslogfound}) {
    echo "Running epmautomate downloadfile \"${logfile}\""
    epmautomate downloadfile "${logfile}"
    if (${transactionsfilefound}) {
        echo "Running epmautomate downloadfile ${transactionsfile}"
        epmautomate downloadfile "${transactionsfile}"
        if ($?) {
            echo "Running epmautomate copyToObjectStorage $
{transactionsfile} ${objectstorageusername} ${objectstoragepassword} $
{objectstorageurl}"
            epmautomate copyToObjectStorage "${transactionsfile}" "$
{objectstorageusername}" "${objectstoragepassword}" "${objectstorageurl}"
            if ($?) {
                echo "Running epmautomate purgeArchivedTMTransactions JOBID=$
{jobid}"
                epmautomate purgeArchivedTMTransactions "JobID=${jobid}"
            }
        }
        else {
            echo "EPMAutomate copyToObjectStorage failed. Purging of
archived matched transactions will not be attempted."
        }
    }
    else {
        echo "Download of transactions file ${transactionsfile} failed.
Copy to object storage, and purging of archived matched transactions will not
be attempted."
    }
    else {
        echo "No matched transactions found. Archive file download, copy to
object storage, and purging of archived matched transactions will not be
attempted."
    }
}
else {
    echo "Matchtype ID \"${matchtype}\" not found. Archive file download,
copy to object storage, and purging of archived matched transactions will not
be attempted."
}

echo "Running epmautomate logout"

```

```
epmautomate logout
echo "Script processing completed."
```

Linux/UNIX 스크립트

다음 스크립트를 복사하여 transaction_match.sh를 생성합니다. input.properties가 저장된 폴더에 파일을 저장합니다.

```
#!/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
fi

echo "Running epmautomate logout"
${epmautomatescript} logout
echo "Script processing completed."

```

Profitability and Cost Management의 샘플 시나리오

이러한 시나리오에서는 몇 가지 일반적인 Profitability and Cost Management 태스크를 수행하기 위한 EPM Automate 명령 사용을 살펴봅니다.

참조:

- [애플리케이션으로 메타데이터 импорт](#)
이러한 스크립트를 사용하여 메타데이터 파일을 업로드하고 해당 파일의 차원 메타데이터를 Profitability and Cost Management 애플리케이션으로 импорт합니다.
- [데이터 импорт 및 프로그램 규칙 실행](#)
이러한 스크립트를 사용하여 데이터 파일을 업로드하고 업로드된 파일의 데이터를 Profitability and Cost Management 비즈니스 프로세스로 импорт합니다.

애플리케이션으로 메타데이터 импорт

이러한 스크립트를 사용하여 메타데이터 파일을 업로드하고 해당 파일의 차원 메타데이터를 Profitability and Cost Management 애플리케이션으로 импорт합니다.

이러한 스크립트는 다음 작업을 수행합니다.

- 환경에 로그인합니다.
- 메타데이터 파일을 업로드합니다.
- 업로드된 파일에서 애플리케이션으로 메타데이터를 импорт합니다.

- 애플리케이션을 사용으로 설정합니다.
- 로그아웃합니다.

Windows 스크립트

다음 스크립트를 복사하여 importMetadata.ps1을 생성합니다.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -raw)
$username="$($inputproperties.username)"
$passwordfile="$($inputproperties.passwordfile)"
$serviceURL="$($inputproperties.serviceURL)"
$applicationName="$($inputproperties.applicationName)"
$dataFileName="$($inputproperties.dataFileName)"
$dataFileNameDestination="$($inputproperties.dataFileNameDestination)"

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile "${dataFileName}" ${dataFileNameDestination}
epmautomate loadaddimdata ${applicationName} dataFileName=${dataFileName}
epmautomate enableapp ${applicationName}
epmautomate logout
```

Linux/UNIX 스크립트

다음 스크립트를 복사하여 importMetadata.sh를 생성합니다.

```
#!/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
```

input.properties 파일 생성

importMetadata 스크립트를 실행하려면 input.properties 파일을 생성하고 해당 환경에 대한 정보로 업데이트합니다. importMetadata.ps1 또는 importMetadata.sh가 저장된 디렉토리에 파일을 저장합니다.

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
```

표 3-26 input.properties 매개변수

매개변수	설명
javahome	JAVA_HOME 위치입니다. Linux/UNIX에만 해당합니다.
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epautomate.sh)입니다. Linux/UNIX에만 해당합니다.
username	ID 도메인 관리자 역할도 가진 서비스 관리자의 사용자 이름입니다.
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
serviceURL	스냅샷을 생성하려는 소스 환경의 URL입니다.
applicationName	데이터를 로드할 타겟 Profitability and Cost Management의 이름입니다.
dataFileName	임포트할 메타데이터가 포함된 파일의 이름입니다.
dataFileNameDestination	메타데이터 파일의 업로드 위치입니다.

스크립트 실행

- 이전 섹션에서 스크립트를 복사하여 importMetadata.ps1 또는 importMetadata.sh를 생성합니다.
- input.properties 파일을 생성하고 importMetadata 스크립트가 있는 디렉토리에 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. [input.properties 파일 생성](#)을 참조하십시오.
이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 관리자 권한으로 실행 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
- 스크립트를 실행합니다.
 - Windows PowerShell:** importMetadata.ps1을 실행합니다.
 - Linux/UNIX:** ./importMetadata.sh를 실행합니다.

데이터 임포트 및 프로그램 규칙 실행

이러한 스크립트를 사용하여 데이터 파일을 업로드하고 업로드된 파일의 데이터를 Profitability and Cost Management 비즈니스 프로세스로 임포트합니다.

이러한 스크립트는 다음 단계를 완료합니다.

- 환경에 사인인합니다.
- 로드할 데이터가 포함된 데이터 파일을 업로드합니다.
- 데이터 규칙이 포함된 규칙 파일을 업로드합니다.
- 데이터 파일의 데이터를 애플리케이션으로 로드하여 기존 값을 덮어씁니다.
- 규칙 파일의 모든 규칙을 실행합니다.
- 로그아웃합니다.

Windows 스크립트

다음 스크립트를 복사하여 importData.ps1이라는 파일을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
$inputproperties = ConvertFrom-StringData(Get-Content ./input.properties -raw)
$username="$($inputproperties.username)"
$passwordfile="$($inputproperties.passwordfile)"
$serviceURL="$($inputproperties.serviceURL)"
$applicationName="$($inputproperties.applicationName)"
$dataFileName="$($inputproperties.dataFileName)"
$rulesFileName="$($inputproperties.rulesFileName)"
$fileDestination="$($inputproperties.fileDestination)"
$clearAllDataFlag="$($inputproperties.clearAllDataFlag)"
$dataLoadValue="$($inputproperties.dataLoadValue)"

epmautomate login ${username} ${passwordfile} ${serviceURL}
epmautomate uploadfile "${dataFileName}" ${fileDestination}
epmautomate uploadfile "${rulesFileName}" ${fileDestination}
epmautomate loaddata ${applicationName} clearAllDataFlag=${clearAllDataFlag}
dataLoadValue=${dataLoadValue} rulesFileName=${rulesFileName}
dataFileName=${dataFileName}"
epmautomate logout
```

Linux/UNIX 스크립트

다음 스크립트를 복사하여 importData.sh라는 파일을 생성합니다. 로컬 디렉토리에 파일을 저장합니다.

```
#!/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
```

input.properties 파일 생성

importData 스크립트를 실행하려면 input.properties 파일을 생성하고 해당 환경에 대한 정보로 업데이트합니다. importData.ps1 또는 importData.sh가 저장된 디렉토리에 파일을 저장합니다.

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
```

표 3-27 input.properties 매개변수

매개변수	설명
javahome	JAVA_HOME 위치입니다. Linux/UNIX에만 해당합니다.
epmautomatescript	EPM Automate 실행 파일의 절대 경로(epmautomate.sh)입니다. Linux/UNIX에만 해당합니다.
username	ID 도메인 관리자 역할도 가진 서비스 관리자의 사용자 이름입니다.
password	서비스 관리자의 비밀번호 또는 암호화된 비밀번호 파일의 이름 및 위치입니다.
serviceURL	스냅샷을 생성하려는 소스 환경의 URL입니다.
applicationName	데이터를 로드할 타겟 Profitability and Cost Management의 이름입니다.
dataFileName	임포트할 데이터가 포함된 파일의 이름입니다.
rulesFileName	임포트된 데이터에 대해 실행할 규칙이 포함된 파일의 이름입니다.
fileDestination	데이터 및 규칙 파일을 업로드할 타겟 위치입니다.
clearAllDataFlag	애플리케이션 큐브에서 기존 데이터를 지울지 여부를 지정합니다. 기존 데이터를 지우지 않으려면 false로 설정합니다.
dataLoadValue	기존 데이터를 처리하는 방법을 지정합니다. 큐브에서 기존 데이터를 유지하려면 ADD_TO_EXISTING을 지정합니다.

스크립트 실행

- 이전 섹션에서 스크립트를 복사하여 importData.ps1 또는 importData.sh를 생성합니다.
- input.properties 파일을 생성하고 importData 스크립트가 있는 디렉토리에 저장합니다. 이 파일의 콘텐츠는 운영 체제에 따라 다릅니다. [input.properties 파일 생성](#)을 참조하십시오. 이 디렉토리에서 쓰기 권한이 있어야 합니다. Windows의 경우 관리자 권한으로 실행 옵션으로 PowerShell을 시작해야 스크립트를 실행할 수 있습니다.
- 스크립트를 실행합니다.
 - Windows PowerShell:** importData.ps1을 실행합니다.
 - Linux/UNIX:** ./importData.sh를 실행합니다.

Cloud EDM의 샘플 시나리오

이러한 샘플 시나리오에서는 EPM Automate 명령을 사용하여 Oracle Fusion Cloud Enterprise Data Management와 Oracle Fusion Cloud Enterprise Performance Management 간에 애플리케이션 차원을 동기화하는 작업에 대해 살펴봅니다.

참조:

- [Cloud EDM 차원 및 매핑을 Cloud EPM 애플리케이션과 동기화](#)
이 샘플 시나리오에서는 Oracle Fusion Cloud Enterprise Data Management 애플리케이션과 Oracle Fusion Cloud Enterprise Performance Management 애플리케이션 간 차원 동기화를 살펴봅니다.
- [Cloud EPM 차원을 Cloud EDM 애플리케이션과 동기화](#)
이 샘플 시나리오에서는 Oracle Fusion Cloud Enterprise Performance Management 애플리케이션과 Oracle Fusion Cloud Enterprise Data Management 애플리케이션 간의 차원 동기화에 대해 살펴봅니다.

Cloud EDM 차원 및 매핑을 Cloud EPM 애플리케이션과 동기화

이 샘플 시나리오에서는 Oracle Fusion Cloud Enterprise Data Management 애플리케이션과 Oracle Fusion Cloud Enterprise Performance Management 애플리케이션 간 차원 동기화를 살펴봅니다.

이 섹션의 스크립트를 사용하여 다음 태스크를 완료합니다.

- Cloud EDM 애플리케이션에서 차원 익스포트
- Cloud EDM 애플리케이션 차원에서 매핑 익스포트
- 익스포트 파일을 Cloud EPM 환경에 복사
- 차원 메타데이터 및 매핑을 Cloud EPM 애플리케이션으로 임포트

Cloud EDM 애플리케이션과 Cloud EPM 애플리케이션 간에 차원 및 매핑을 동기화하려면:

1. 다음 스크립트를 복사하여 스크립트 파일을 생성합니다.

```
rem Integration example to sync application dimensions between Cloud EDM
and Cloud EPM
rem Windows script for demonstration purposes only; do not use in
production environments

set EDMUSER=userid
set EDMSVR=https://hostname
set EDMPWDFILE=example_EDM
set EDMAPP=appname
set EDMDIM=dimname
set EDMLOC=location

set EPMUSER=userid
set EPMSVR=https://hostname
set EPIMPJOB=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 Cloud EPM 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. 스크립트 파일을 수정하고 필수 매개변수 값을 설정합니다. 매개변수 설명 및 예제는 [스크립트 실행 매개변수](#)를 참조하십시오.
3. 스크립트를 수동으로 실행하거나 필요에 따라 실행하도록 스케줄링합니다. [스크립트 실행 자동화](#)를 참조하십시오.

스크립트 실행 매개변수

이 섹션의 스크립트 파일에는 다음 테이블에 설명된 몇 가지 매개변수 값을 지정해야 합니다. 모든 매개변수가 모든 스크립트에서 사용되는 것은 아닙니다.

표 3-28 스크립트 파일의 매개변수 값

매개변수	설명
EDMUSER	Cloud EDM 서비스 관리자의 사용자 로그인 ID입니다. 예: EDMUSER=jdoe@example.com
EDMSVR	Cloud EDM 환경의 URL입니다. 예: EDMSVR=https:// example.oraclecloud.com
EDMPWDFILE	Cloud EDM 서비스 관리자의 EPW(암호화된 비밀번호 파일) 이름 및 위치입니다. 예: EDMPWDFILE=edm_jdoe.epw
EDMAPP	Cloud EDM 애플리케이션 차원의 이름입니다. 예: EDMAPP=USOperations
EDMDIM	익스포트하거나 임포트할 차원의 이름입니다. 예: EDMDIM=entity
EDMLOC	익스포트할 위치의 이름입니다. 예: EDMLOC=Loc1
EPMUSER	Cloud EPM 서비스 관리자의 로그인 이름입니다. 예: EPMUSER=john.doe@example.com

표 3-28 (계속) 스크립트 파일의 매개변수 값

매개변수	설명
EPMSVR	Cloud EPM 환경의 URL입니다. 예: EPMSVR=https://example.oraclecloud.com
EPMIMPJOB	Cloud EPM 환경에 있는 메타데이터 임포트 유형의 기존 임포트 작업 이름입니다. 예: EPMIMPJOB=imp_DIMMetadata
EPMEXPJOB	Cloud EPM 환경에 있는 메타데이터 익스포트 유형의 기존 작업 이름입니다. 예: EPMEXPJOB=Exp_DIMMetadata
PWDFILE	Cloud EPM 서비스 관리자의 암호화된 비밀번호 파일(EPW) 이름 및 위치입니다. encrypt 명령을 참조하십시오. 예: PWDFILE=pwd_jdoe.epw
DIMFILE	익스포트한 차원 데이터를 보관할 파일의 이름입니다. 예: DIMFILE=entity_file.CSV
MAPFILE	익스포트한 매핑 데이터를 보관할 파일의 이름입니다. 예: MAPFILE=map_file.CSV

Cloud EPM 차원을 Cloud EDM 애플리케이션과 동기화

이 샘플 시나리오에서는 Oracle Fusion Cloud Enterprise Performance Management 애플리케이션과 Oracle Fusion Cloud Enterprise Data Management 애플리케이션 간의 차원 동기화에 대해 살펴봅니다.

이 섹션의 스크립트를 사용하여 다음 태스크를 완료합니다.

- Cloud EPM 애플리케이션에서 메타데이터(차원) 익스포트
- 차원 데이터가 포함된 익스포트 파일을 Cloud EDM 환경에 복사
- 차원 메타데이터를 Cloud EDM 애플리케이션으로 임포트

Cloud EPM 애플리케이션과 Cloud EDM 애플리케이션 간에 차원을 동기화하려면:

1. 다음 스크립트를 복사하여 Windows 스크립트 파일을 생성합니다.

```
rem Integration example to sync an application dimension between Cloud EPM
and Cloud EDM
rem Windows script for demonstration purposes only; do not use in
production environments

set EDMUSER=userid
set EDMSVR=https://hostname
set EDMPWDFILE=example_EDM.epw
set EDMAPP=appname
set EDMDIM=dimname

set EPMUSER=userid
set EPMSVR=https://hostname
set PWDFILE=example_epm.epw
set EPMEXPJOB=exportjobname

rem Synchronizing EPM ---> EDM
```

```

rem Export Metadata from EPM
call epmautomate login %EPMUSER% %PWDFILE% %EPMSVR%
call epmautomate exportmetadata %EPMEXPJOB%
call epmautomate logout

rem Import Dimension to EDM
rem Log into the EDM environment
call epmautomate login %EDMUSER% %EDMPWDFILE% %EDMSVR%
rem Copy exported metadata file into the EDM environment
call epmautomate copyfilefrominstance %EPMEXPJOB%.zip %EPMUSER% %PWDFILE%
%EPMMSVR% %EPMEXPJOB%.zip
call epmautomate importdimension %EDMAPP% %EDMDIM% ReplaceNodes
%EPMEXPJOB%.zip
call epmautomate logout

```

스크립트 파일을 수정하고 필수 매개변수 값을 설정합니다. 매개변수 설명 및 예제는 [스크립트 실행 매개변수](#)를 참조하십시오.

2. 스크립트를 수동으로 실행하거나 필요에 따라 실행하도록 스케줄링합니다. [스크립트 실행 자동화](#)를 참조하십시오.

스크립트 실행 자동화

서비스 관리자는 Windows 태스크 스케줄러에서 스크립트를 스케줄링하거나 cron 작업을 사용하여 EPM Automate를 통해 활동을 자동화합니다.

Windows 태스크 스케줄러를 사용하여 EPM Automate 스크립트 실행을 스케줄링하려면 다음을 수행합니다.

1. 시작, 제어판, 관리 도구 순으로 누릅니다.
2. 태스크 스케줄러를 엽니다.
3. 작업, 기본 태스크 생성을 선택합니다.
4. 태스크 이름과 선택적 설명을 입력한 후 다음을 누릅니다.
5. 태스크 트리거에서 스크립트를 실행할 스케줄을 선택한 후 다음을 누릅니다.
6. 다음 화면에서 다른 스케줄 매개변수를 지정한 후 다음을 누릅니다.
7. 작업에서 프로그램 시작이 선택되었는지 확인합니다.
8. 프로그램 시작에서 다음 단계를 완료합니다.
 - a. 프로그램/스크립트에서 예약하려는 스크립트를 찾아 선택합니다.
 - b. 인수 추가(선택사항)에서 SET user 스크립트 매개변수로 확인되는 서비스 관리자의 비밀번호를 입력합니다.
 - c. 시작 위치(선택사항)에 EPM Automate가 설치된 위치를 입력합니다. 이 위치는 일반적으로 C:/Oracle/EPMAutomate/bin입니다.
 - d. 다음을 누릅니다.
9. 요약에서 완료를 누르면 이 태스크의 등록정보 대화상자 열기를 선택한 다음 완료를 누릅니다.
10. 일반에서 이러한 보안 옵션을 선택한 다음 확인을 누릅니다.
 - 사용자 로그인 여부에 상관없이 실행
 - 가장 높은 권한으로 실행

EPM Automate 활동 모니터

초기화된 작업의 상태를 확인하는 데 도움이 되도록 EPM Automate에서는 EPM Automate를 실행한 콘솔에 상태 코드를 표시합니다.

[종료 코드](#)를 참조하십시오.

작업 콘솔을 사용하여 EPM Automate에서 실행하는 작업을 모니터할 수 있습니다. 자세한 내용은 *Planning 관리*에서 작업 관리를 참조하십시오.

4

EPM Automate를 설치하지 않고 명령 실행

Groovy를 사용하면 Oracle Fusion Cloud Enterprise Performance Management에서 선택된 명령을 직접 실행할 수 있습니다. 서버측 명령을 실행하는 데 EPM Automate를 설치할 필요는 없습니다.

① Note

이 시나리오에서 Groovy 스크립트는 클라이언트 머신이 아닌 Cloud EPM에서 직접 실행하도록 작성됩니다.

이 장의 내용:

- [서버측 명령 실행을 지원하는 환경](#)
- [정보 소스](#)
- [지원되는 명령](#)
- [서버측 Groovy를 사용하여 EPM Automate를 실행하는 데 사용할 방법](#)
- [서버측 Groovy 스크립트를 사용하여 환경 복제](#)
- [서버측 Groovy 스크립트를 사용하여 활동 보고서를 전자메일로 보내기](#)

서버측 명령 실행을 지원하는 환경

서버측 EPM Automate 명령의 실행에 대한 Groovy 스크립트 지원은 다음 환경에서만 사용할 수 있습니다.

- EPM Enterprise Cloud Service 환경에 배포된 Planning 및 Planning 모듈 비즈니스 프로세스
- Enterprise Planning and Budgeting Cloud
- Planning and Budgeting Cloud(Plus One 옵션 포함)
- 무형식
- Enterprise Profitability and Cost Management
- EPM Enterprise Cloud Service 환경의 Financial Consolidation and Close
- EPM Enterprise Cloud Service 환경의 Tax Reporting
- Strategic Workforce Planning
- Sales Planning

EPM Automate 명령이 통합된 Groovy 스크립트는 앞의 Oracle Fusion Cloud Enterprise Performance Management 환경에서만 작성 및 실행될 수 있습니다. 하지만 이러한 환경에서 작성된 스크립트는 임의의 Cloud EPM 환경에서 EPM Automate 명령을 실행할 수 있습니다. 예를 들어 Planning EPM Enterprise Cloud Service 환경에서 스크립트를 생성하고 Groovy 스크립트를 지원하지 않는 Narrative Reporting 환경에서 명령을 실행할 수 있습니다.

정보 소스

자세한 내용은 *Calculation Manager를 사용하여 디자인*에서 다음 소스를 참조하십시오.

- Groovy 비즈니스 규칙 정보
- Groovy 비즈니스 규칙 생성

지원되는 명령

다음은 제외한 모든 EPM Automate 명령은 Groovy를 통해 실행할 수 있습니다.

- [downloadFile](#)
- [upgrade](#)
- [uploadFile](#)

다음 명령은 Groovy 스크립트 실행 환경에서 실행할 수 없습니다.

- [recreate](#)
- [replay](#)
- [resetService](#)
- [restructureCube](#)

① Note

- [encrypt](#) 명령을 실행하면 암호화된 비밀번호 파일이 서버에 생성되며 장기간 사용을 위해 보관됩니다. [listFiles](#) 명령을 실행하면 암호화된 비밀번호 파일이 나열되지 않습니다. [downloadFile](#) 명령을 사용하여 서버에서 해당 파일을 다운로드할 수 없습니다.
- [feedback](#) 명령이 실행되려면 첨부파일로 사용되는 모든 파일 및 스크린샷을 기본 업로드 위치에서 사용할 수 있어야 합니다. [기본 업로드 위치](#)를 참조하십시오. [uploadFile](#) 명령에서 위치를 지정하지 않은 경우 파일이 저장되는 위치입니다.

서버측 Groovy를 사용하여 EPM Automate를 실행하는 데 사용할 방법

- `getEPMAutomate ()` 이 정적 메소드는 EPM Automate 명령을 호출하는 데 사용할 수 있는 `EpmAutomate` 클래스 인스턴스를 제공합니다.
- `execute ()` 이 `EpmAutomate` 클래스 메소드는 EPM Automate 명령을 실행하는 데 사용됩니다. EPM Automate 명령 이름을 첫번째 매개변수로, 명령 옵션을 후속 매개변수로 전달하십시오. 이 메소드는 `EpmAutomateStatus` 클래스 인스턴스를 반환합니다.
- `getStatus ()` 이 `EPMAutomateStatus` 클래스 메소드는 명령에서 반환된 실행 상태를 반환합니다. 반환 값 0은 성공을 나타내며 0이 아닌 값은 명령 실패를 나타냅니다.
- `getOutput ()` 이 `EPMAutomateStatus` 클래스 메소드는 명령의 문자열 출력을 반환합니다. 예를 들어 이 메소드를 사용하여 `getApplicationAdminMode` 및

getDailyMaintenanceStartTime 명령의 출력을 반환할 수 있습니다. 명령의 반환 상태가 0이 아니면 이 메소드는 오류 메시지를 반환합니다.

- getItemsList () 이 EPMAutomateStatus 클래스 메소드는 명령의 목록 출력을 반환합니다. 예를 들어 이 메소드를 사용하여 getSubstVar, listBackups, listFiles 명령의 출력을 반환할 수 있습니다.

서버측 Groovy 스크립트를 사용하여 환경 복제

재해 복구를 위해 서버측 Groovy 스크립트에 EPM Automate 명령을 포함하여 환경을 복제할 수 있습니다. 이렇게 하면 온-프레미스 풋프린트 없이 재해 복구를 설정할 수 있습니다.

비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오. 또한, 환경에 적합하게 다음 매개변수 값을 바꾸십시오.

Table 4-1 변경할 매개변수

매개변수	설명
password	소스 환경에서 복제 작업을 수행하는 서비스 관리자 비밀번호입니다.
targetpassword	타겟 환경에서 복제 작업을 수행하는 서비스 관리자 비밀번호입니다.
username	소스 환경에서 서비스 관리자의 사용자 ID입니다.
targetusername	타겟 환경에서 서비스 관리자의 사용자 ID입니다. 이 사용자는 타겟 환경에서 ID 도메인 관리자 역할에도 지정되어야 합니다.
email_id	복제 프로세스에 대한 정보를 보낼 전자메일 주소입니다.

비밀번호 암호화 스크립트

```
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())
```

환경 복제 스크립트

이 스크립트는 이전 스크립트를 사용하여 생성된 암호화된 비밀번호 파일을 사용합니다.

```
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())
```

서버측 Groovy 스크립트를 사용하여 활동 보고서를 전자메일로 보내기

이 스크립트를 사용하여 활동 보고서를 목록의 수신자에게 전자메일로 보낼 수 있습니다. 그런 다음, 이 스크립트를 매일 실행하여 매일 활동 보고서를 가져오도록 스케줄링할 수 있습니다. 이 스크립트는 다음 기능을 수행합니다.

- Groovy 스크립트를 실행하는 서비스 관리자의 비밀번호를 암호화합니다.
- 암호화된 비밀번호를 사용하여 Oracle Fusion Cloud Enterprise Performance Management 환경에 로그인합니다.
- 환경에서 사용할 수 있는 활동 보고서를 목록의 수신자, 일반적으로 서비스 관리자에게 전자메일로 보냅니다.
- 환경에서 로그아웃합니다.

비밀번호에 특수 문자가 포함된 경우 [특수 문자 처리](#)를 참조하십시오. 또한, 환경에 적합하게 다음 매개변수 값을 바꾸십시오.

Table 4-2 변경할 매개변수

매개변수	설명
user	환경에 로그인할 서비스 관리자의 사용자 ID입니다.
password	서비스 관리자의 비밀번호입니다.
url	활동 보고서를 전자메일로 보낼 Cloud EPM 환경의 URL입니다.
emailaddresses	활동 보고서를 보낼 전자메일 주소의 세미콜론으로 구분된 목록입니다.

Groovy 규칙 사용에 대한 자세한 내용은 *Planning 관리*의 Groovy 규칙 사용을 참조하십시오.

```

/*RTPS: {user} {password} {url} {emailaddresses}*/

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

Cloud EPM 환경 복제

다음 단계는 예기치 못한 상황으로 인해 기본 지역을 사용할 수 없는 경우 서비스 가용성을 확보하기 위해 보조 Oracle Fusion Cloud Enterprise Performance Management 환경을 구성하는 작업에 관한 것입니다.

① 주

이 부록에 설명된 절차는 Narrative Reporting에 적용할 수 없습니다.

- [일별 아티팩트 복제 설정](#)
- [주문형 복제 설정](#)
- [보조 환경 구성](#)

일별 복제 설정

환경을 복제하려면 EPM Automate를 사용하여 일별 유지관리 중에 생성된 기본 환경의 아티팩트 스냅샷을 보조 환경으로 복사합니다.

Oracle은 각 환경에 대한 일상적인 유지관리를 매일 수행합니다. 이러한 서비스 유지관리 중에 오라클은 환경의 콘텐츠(ID 도메인의 사용자 및 역할 지정을 비롯한 기존 데이터 및 아티팩트)를 백업하여 유지관리 스냅샷을 생성합니다.

일별 서비스 복제를 설정하려면 다음을 수행합니다.

1. 다음 EPM Automate 명령을 포함하는 스크립트 파일을 생성합니다. 이 스크립트는 기본 환경의 애플리케이션 스냅샷을 보조 환경에 복제합니다.

① 주

사용자 이름, 비밀번호 파일, ID 도메인 이름 및 서비스 URL을 변경해야 합니다. 암호화된 비밀번호 파일 생성에 대한 자세한 내용은 [encrypt](#) 명령을 참조하십시오.

```
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
```

- 스케줄러(예: Windows 태스크 스케줄러)를 사용하여 유지 관리 기간 시작부터 2시간 동안 실행되도록 스크립트 파일 실행을 예약합니다.
- 기본 환경과 보조 환경 둘 다에서 동일한 유지관리 기간 시작 시간을 설정합니다. 자세한 내용은 *관리자용 시작 가이드*에서 서비스 유지 관리 시간 설정을 참조하십시오.

주문형 복제 설정

RPO를 줄이기 위해 기본 환경의 주문형 스냅샷을 생성한 다음 보조 환경에 복사할 수 있습니다.

예를 들어 일별 복제 사이에 6시간마다 실행되는 EPM Automate 스크립트를 생성 및 스케줄링하여 RPO를 24시간에서 6시간으로 줄일 수 있습니다.

① 주

주문형 스냅샷 생성 시 기본 환경은 몇 분 동안 읽기 전용 모드로 전환됩니다.

주문형 복제를 설정하려면 다음을 수행합니다.

- 다음 EPM Automate 명령을 포함하는 스크립트 파일을 생성합니다. 이 스크립트는 기본 환경의 애플리케이션 스냅샷을 보조 환경에 복제합니다.

① 주

사용자 이름, 비밀번호 파일, ID 도메인 이름 및 서비스 URL을 변경해야 합니다. 암호화된 비밀번호 파일 생성에 대한 자세한 내용은 [encrypt](#) 명령을 참조하십시오.

```
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
```

- 스케줄러(예: Windows 태스크 스케줄러)를 사용하여 원하는 RPO에 맞게 필요할 때마다 실행되도록 스크립트 파일 실행을 예약합니다.

보조 환경 구성

보조 환경을 구성하여 활성화합니다.

확장된 기간에 기본 환경을 사용할 수 없을 때 보조 환경을 활성화해야 하는 경우에만 이 절차를 완료합니다. 보조 환경을 구성하기 전에 *관리자용 시작 가이드*에서 다음 항목을 참조하십시오.

- 레거시 EPM Cloud 스냅샷의 마이그레이션 경로
- EPM Standard Cloud Service 및 EPM Enterprise Cloud Service 스냅샷의 마이그레이션 경로
보조 환경을 구성하려면 다음을 수행합니다.

1. EPM Automate 세션을 시작하고 해당 작업을 완료합니다.

- 서비스 관리자 및 ID 도메인 관리자 역할이 둘 다 지정된 계정을 사용하여 보조 환경에 사인인합니다. 적절한 사용자 이름, 비밀번호, 도메인 이름 및 서비스 URL을 지정해야 합니다.
- 보조 환경을 재생성합니다.
 - 기본 환경이 Planning, Tax Reporting, Financial Consolidation and Close 또는 Enterprise Profitability and Cost Management 환경인 경우 다음을 사용합니다.
`epmautomate recreate -f`
 - 기본 환경이 Planning, Tax Reporting, Financial Consolidation and Close 또는 Enterprise Profitability and Cost Management 환경이 아닌 경우 다음을 사용합니다.
`epmautomate recreate -f TempServiceType=PRIMARY_APPLICATION_TYPE` 설명:
`PRIMARY_APPLICATION_TYPE`은 ARCS, EDMCS, PCMCS 또는 EPRCS입니다.
- 스냅샷에서 애플리케이션 및 ID 도메인 아티팩트를 임포트합니다.
- 사인아웃

다음 명령을 실행하여 이전 작업을 완료할 수 있습니다. 다음 항목을 참조하십시오.

- [login](#) 명령
- [recreate](#) 명령
- [importSnapshot](#) 명령

```
epmautomate login serviceAdmin secondaryPassword.epw secondary_URL
epmautomate recreate -f
epmautomate importsnapshot "Artifact Snapshot" importUsers=true
epmautomate logout
```

2. 보조 환경에 사인인하고 모든 데이터를 사용할 수 있는지 확인합니다.
3. 모든 사용자에게 보조 환경의 URL을 보냅니다.

A

simulateConcurrentUsage 명령 실행 준비

[simulateConcurrentUsage](#) 명령은 환경에서 로드를 시뮬레이션하기 위해 다음 작업을 지원합니다.

- 양식 열기
- 양식 저장
- 비즈니스 규칙 실행
- 비즈니스 규칙 세트 실행
- 데이터 규칙 실행
- 임시 그리드 열기
- 관리 보고서 장부 실행
- 관리 보고 보고서 실행

관련 단계

1. requirement.csv 파일을 생성합니다. [requirement.csv 파일 생성](#)을 참조하십시오.
2. requirement.csv에 포함된 작업의 세부정보를 지정하는 입력 파일을 생성합니다. 참조:
 - [양식 입력 파일 열기](#)
 - [양식 입력 파일 저장](#)
 - [비즈니스 규칙 입력 파일 실행](#)
 - [비즈니스 규칙 세트 입력 파일 실행](#)
 - [데이터 규칙 입력 파일 실행](#)
 - [임시 그리드 입력 파일](#)
 - [장부 입력 파일 실행](#)
 - [보고서 입력 파일 실행](#)
3. 사용자 변수 멤버 매핑을 설정해야 하는 경우 사용자 변수 세부정보가 포함된 UserVarMemberMapping.csv를 생성합니다. [UserVarMemberMapping.csv 파일 생성](#)를 참조하십시오.
4. 필요에 따라 Smart View 옵션을 사용해야 하는 경우 Oracle Smart View for Office 옵션을 options.xml로 익스포트합니다. 자세한 내용은 [options.xml 파일 생성](#)을 참조하십시오.
5. 필요에 따라 기존 사용자를 사용하여 시뮬레이션을 실행하려면 users.csv 파일을 생성합니다. 자세한 내용은 [users.csv 파일 생성](#)을 참조하십시오.
6. 이전 단계에서 생성한 파일이 포함된 ZIP 파일을 생성하고 환경에 업로드합니다. [입력 ZIP 파일 생성 및 환경에 업로드](#)를 참조하십시오.
7. 업로드된 ZIP 파일을 사용하여 [simulateConcurrentUsage](#) 명령을 실행합니다.

requirement.csv 파일 생성

먼저 테스트할 사용 사례의 세부정보를 나열하는 requirement.csv 파일을 생성합니다. 이 CSV 파일의 각 라인은 수행할 작업 유형, 아티팩트 이름, 동시 사용자 수, 작업 세부정보를 지정하는 입력 파일, 작업 관련 추가 정보(있는 경우)를 확인합니다. 예를 들어 2개 양식을 열고, 2개 양식을 저장하고, 2개 비즈니스 규칙을 실행하려면 입력 CSV 파일에서 6개 라인을 지정합니다. requirement.csv의 첫번째 라인에서 헤더에는 다음 정보를 포함해야 합니다.

```
#Type of Operation,Artifact Name,Number of Users,Input File,Additional Info
```

파일의 각 후속 라인에는 단일 작업 및 해당 매개변수가 포함됩니다. 일부 작업에는 이러한 일부 매개변수 값이 필요하지 않을 수도 있습니다. 예상되는 파일 입력은 다음 테이블에서 설명합니다.

① Note

테이블에 별도로 표시된 경우가 아니면 모든 값이 필요합니다.

Table A-1 requirement.csv 형식

필드	설명
작업 유형	다음 중 하나여야 합니다. <ul style="list-style-type: none"> Open Form Save Form Run Business Rule 비즈니스 규칙 세트 실행 Run Data Rule Ad Hoc Grid Execute Report Execute Book
아티팩트 이름	이 값은 작업 유형에 따라 달라집니다. <ul style="list-style-type: none"> 양식 열기: 열려는 양식의 이름 및 위치. 양식 저장: 저장할 양식의 이름 및 위치. 비즈니스 규칙 실행: 비즈니스 규칙의 이름. 비즈니스 규칙 세트 실행: 비즈니스 규칙 세트의 이름. 데이터 규칙: 데이터 규칙의 이름. 임시 그리드: 해당 없음(공백 유지). 보고서 실행: 보고서의 이름 및 위치. 장부 실행: 장부의 이름 및 위치.
사용자 수	동시 사용을 시뮬레이션할 사용자 수.
입력 파일	POV 값, 런타임 프롬프트 또는 사용할 기타 사용 사례 특정 값을 포함하는 CSV 파일의 이름.
추가 정보	작업에 필요한 추가 매개변수. 임시 그리드에만 적용됩니다. 다른 사용 사례의 경우 비워 둡니다.

참고: 아티팩트 이름은 애플리케이션의 해당 이름과 일치해야 하고 대소문자가 동일해야 합니다.

파운드 기호(#)가 있는 라인을 시작하여 주석 처리합니다. 그러나 파일의 첫번째 라인은 헤더이며 #으로 시작해야 합니다.

requirement.csv 파일 예제:

```
# 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 Business Ruleset,RollupUSSales,5,runbusinessruleset_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
```

입력 파일 생성

requirement.csv에서 확인된 각 사용 사례에는 실행하는 데 필요한 모든 매개변수를 제공하는 일치하는 입력 파일이 있어야 합니다.

이상적으로, 입력 파일에는 requirement.csv에서 이 사용 사례에 대해 지정된 사용자 수당 하나의 입력이 포함되어야 합니다.

입력 파일의 입력 수가 requirement.csv의 해당 사용 사례에 대한 동시 사용자 수보다 적으면 EPM Automate는 입력 파일의 일부 입력을 반복하여 requirement.csv에서 확인된 사용자 수에 대해 작업이 실행될 때까지 사용 사례를 실행합니다.

예를 들어 비즈니스 규칙 실행 작업에 대한 requirement.csv의 사용 사례 입력이 다음과 같은 경우:

```
Run Business Rule, Copy Budget,10,br_input_file.csv,
```

br_input_file.csv에는 10개 입력이 포함되어야 합니다. br_input_file.csv에 6개 입력만 포함되어 있으면 EPM Automate는 처음 6명 사용자에게 대해 해당 입력을 사용합니다. 다음 4명 사용자의 경우 EPM Automate는 br_input_file.csv의 처음 4개 입력을 다시 사용합니다.

입력 파일의 입력 수가 사용 사례에 대해 지정된 사용자 수보다 많으면 EPM Automate는 입력 파일의 마지막 추가 입력을 무시합니다.

- [양식 입력 파일 열기](#)
- [양식 입력 파일 저장](#)
- [비즈니스 규칙 입력 파일 실행](#)
- [비즈니스 규칙 세트 입력 파일 실행](#)
- [데이터 규칙 입력 파일 실행](#)
- [임시 그리드 입력 파일](#)
- [보고서 입력 파일 실행](#)
- [장부 입력 파일 실행](#)

양식 입력 파일 열기

양식 열기를 지원하기 위해 requirement.csv에서 참조되는 이 파일에는 다음 형식의 POV 입력이 포함됩니다. pov=[DIM 1:MEMBER 1],[DIM 2:MEMBER 2],[DIM 3:MEMBER 3], 등.

이 설명에서 DIM 1, DIM 2 등은 차원 이름이고 MEMBER 1, MEMBER 2 등은 POV에 대한 차원 멤버 값입니다.

샘플 입력 파일:

```
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

requirement.csv에 지정한 양식에서 사용자 변수를 설정해야 하는 경우 UserVarMemberMapping.csv도 생성해야 합니다. [UserVarMemberMapping.csv 파일 생성](#)를 참조하십시오.

가이드라인

입력 파일의 pov 태그에는 선택할 수 있는 차원 및 사용자 변수만 포함되어야 합니다. 다음 예에서는 예측 및 단위를 선택할 수 없으므로 pov 태그에서 제외해야 합니다(예: pov=[Entity:Sales East Region],[Version View:Working],[ProductFamily:Computer Equipments];).

Sales East Region		Forecast	Units	Version View:Working	ProductFamily:Computer Equipments	Refresh							
N3													
	A	B	C	D	E	F	G	H	I	J	K	L	M
1		FY24					FY25						
2		Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
3	Product X	246	366	296	250	302	333	408	150	282	175	175	275
4	Sentinal Standard Notebook	393	99	386	377	411	454	825	300	306	650	247	333
5	Sentinal Custom Notebook	254	299	269	242	214	271	255	225	250	550	220	150
6	Envoy Standard Netbook	789	881	803	769	838	793	424	121	220	162	157	204
7	Envoy Custom Netbook	390	489	423	416	388	471	270	84	101	174	149	140
8	Other Computer	163	158	117	185	226	112	180	155	145	250	210	215
9	Tablet Computer	174	154	141	176	150	143	425	425	425	425	425	425
10	-Computer Equipments	2409	2446	2435	2415	2529	2577	2787	1460	1730	2387	1583	1742

양식 입력 파일 저장

requirement.csv에서 참조되는 이 파일은 양식 저장을 지원하며, POV 및 다음 형식의 셀 입력 값을 포함합니다. 비즈니스 규칙이 양식과 연계되어 있는 경우 이 파일에는 해당 비즈니스 규칙에 대한 런타임 프롬프트도 포함됩니다.

샘플 입력 파일:

```
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];rtp=[BUSINESS RULE NAME1[RTP1:VALUE1][RTP2:VALUE2]], [BUSINESS
RULE2[RTP3:VALUE3]]..
```

이 예제에서:

- DIM은 차원 이름을 나타내고 MEMBER는 차원 멤버 값을 나타냅니다.
- CELL COLUMN HEADER는 열 머리글의 이름을 확인하고 CELL ROW HEADER는 행 머리글의 이름을 확인합니다.
- BUSINESS RULE NAME은 비즈니스 규칙의 이름을 나타내고, RTP는 런타임 프롬프트 이름을 나타내고, VALUE는 해당 값을 나타냅니다. 양식에 연계된 비즈니스 규칙이 없거나 양식에 있는 기본 런타임 프롬프트 값을 사용하려는 경우에는 RTP가 필요하지 않습니다.

예를 들어 다음과 같습니다.

```
pov=[Version View:Working],[Sales Entity:International Sales];cells=[FY16->x-----x->Pct|P293:Maintenance->4120: Support|1];rtp=[Services Revenue - Forecast[Department:000][Scenario:Plan]],[Allocate Plan Targets[TargetVersion:Baseline]]
```

Note

requirement.csv에 지정한 양식에서 사용자 변수를 설정해야 하는 경우 UserVarMemberMapping.csv도 생성해야 합니다. [UserVarMemberMapping.csv 파일 생성](#)을 참조하십시오.

해당 양식과 연계된 스마트 푸시가 있으면 자동으로 실행됩니다.

가이드라인

1. 입력 파일의 pov 태그에는 선택할 수 있는 차원 및 사용자 변수만 포함되어야 합니다. 다음 예에서는 예측 및 단위를 선택할 수 없으므로 pov 태그에서 제외해야 합니다(예: pov=[Entity:Sales East Region],[Version View:Working],[ProductFamily:Computer Equipments];).

Sales East Region		Forecast	Units	Version View:Working	ProductFamily:Computer Equipments	Refresh							
N3													
	A	B	C	D	E	F	G	H	I	J	K	L	M
1		FY24					FY25						
2		Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
3	Product X	246	366	296	250	302	333	408	150	282	175	175	275
4	Sentinal Standard Notebook	393	99	386	377	411	454	825	300	306	650	247	333
5	Sentinal Custom Notebook	254	299	269	242	214	271	255	225	250	550	220	150
6	Envoy Standard Netbook	789	881	803	769	838	793	424	121	220	162	157	204
7	Envoy Custom Netbook	390	489	423	416	388	471	270	84	101	174	149	140
8	Other Computer	163	158	117	185	226	112	180	155	145	250	210	215
9	Tablet Computer	174	154	141	176	150	143	425	425	425	425	425	425
10	-Computer Equipments	2409	2446	2435	2415	2529	2577	2787	1460	1730	2387	1583	1742

2. Oracle Smart View for Office에서 양식을 열고 셀 열 머리글 및 행 머리글을 cells 태그에 복사합니다. Smart View에서 숨겨진 행 또는 열이 양식에 있는 경우 머리글을 복사하기 전에 숨김을 해제합니다. 셀에 도달하기 위해 숨겨진 행 및 열의 값을 포함하여 모든 열 머리글과 행 머리글을 기록해 둡니다. 위 예에서 타겟 셀이 D3 및 K8인 경우 cells 태그는 cells=[FY24->Sep|Product X|296],[FY25->Apr|Other Computer|250];이어야 합니다.

Note

셀 정의는 선택사항입니다. 셀 정의가 포함되지 않은 경우 이 명령은 지정된 POV에 양식을 저장합니다.

- 다음 상황에서는 rpt 태그가 필요하지 않습니다.
 - 양식과 연계된 비즈니스 규칙이 없습니다.
 - 양식에 있는 기본 런타임 프롬프트 값을 사용하려고 합니다.

비즈니스 규칙 입력 파일 실행

requirement.csv에서 참조되는 이 파일은 비즈니스 규칙 실행을 지원하며, `rtp=[PARAMETER:VALUE],[PARAMETER:VALUE]` 등의 형식으로 런타임 매개변수 값을 포함합니다. 이 형식에서 `PARAMETER`는 런타임 프롬프트 이름을 확인하고 `VALUE`는 해당 값을 확인합니다.

규칙을 실행하는 데 필요한 만큼 런타임 프롬프트를 추가해야 합니다. 런타임 프롬프트 및 해당 값을 제공하지 않으면 기본값이 사용됩니다. 이 명령은 규칙에 대해 정의된 것과 정확히 일치하지 않는 런타임 프롬프트는 무시합니다. 비즈니스 규칙에 대해 런타임 매개변수가 필요하지 않은 경우에는 `rtp=[]`를 포함합니다.

샘플 입력 파일:

```
rtp=[Period:Q1],[Entity:USA]
rtp=[Period:Q2],[Entity:USA]
rtp=[Period:Q3],[Entity:USA]
rtp=[Period:Q4],[Entity:USA]
```

비즈니스 규칙 세트 입력 파일 실행

requirement.csv에서 참조되는 이 파일은 비즈니스 규칙 세트 실행을 지원하며, `rtp=[PARAMETER:VALUE],[PARAMETER:VALUE]` 등의 형식으로 런타임 매개변수 값을 포함합니다. 이 형식에서 `PARAMETER:VALUE`는 런타임 매개변수 이름과 해당 값을 확인합니다.

`PARAMETER:VALUE` 쌍을 사용하여 규칙 세트에 필요한 개수만큼 런타임 프롬프트를 지정합니다. 런타임 매개변수의 값을 제공하지 않으면 기본값이 사용됩니다. 이 명령은 규칙 세트에 대해 정의된 것과 정확히 일치하지 않는 런타임 프롬프트는 무시합니다. 비즈니스 규칙 세트에 대해 런타임 매개변수 값이 필요하지 않은 경우에는 `rtp=[]`를 포함합니다.

샘플 입력 파일:

```
rpt=[Period:Q1],[Entity:USA]
rtp=[Period:Q2],[Entity:USA]
rtp=[Period:Q3],[Entity:USA]
rtp=[Period:Q4],[Entity:USA]
```

데이터 규칙 입력 파일 실행

데이터 규칙 실행을 지원하기 위해 requirement.csv에서 참조되는 이 파일은 시작 기간, 종료 기간, импорт 모드, 익스포트 모드 및 환경에서 사용가능한 선택적 импорт 파일 이름을 지정해야 합니다.

파일 이름을 지정하지 않으면 데이터 규칙에 지정된 파일 이름이 사용됩니다. 각 라인의 형식:
`startperiod=START PERIOD;endperiod=END PERIOD;importmode=IMPORT_MODE;exportmode=EXPORT_MODE;filename=FILE NAME`

샘플 입력 파일:

```
startperiod=Dec-15;endperiod=Dec-15;importmode=REPLACE;exportmode=STORE_DATA;filename=comma
startperiod=Dec-16;endperiod=Dec-16;importmode=REPLACE;exportmode=STORE_DATA;filename=comma
startperiod=Dec-17;endperiod=Dec-17;importmode=REPLACE;exportmode=STORE_DATA;filename=comma
startperiod=Dec-18;endperiod=Dec-18;importmode=REPLACE;exportmode=STORE_DATA;filename=comma
startperiod=Dec-19;endperiod=Dec-19;importmode=REPLACE;exportmode=STORE_DATA;filename=comma
```

임시 그리드 입력 파일

[simulateConcurrentUsage](#) 명령은 기본 모드 및 표준 모드 임시 그리드를 모두 지원합니다. 기본 모드 그리드에서는 POV가 Oracle Smart View for Office 플러그인 툴바의 일부로 표시됩니다. 표준 모드에서는 POV가 스프레드시트 자체에 표시되며 스프레드시트의 첫번째 행을 사용합니다.

requirement.csv에서 참조되는 임시 그리드 입력 파일은 임시 그리드를 여는 기능을 지원하며, 열려는 그리드를 지정해야 합니다. 파일의 각 라인은 다음 형식이어야 합니다.

```
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,..]
```

샘플 입력 파일:

```
fileName=dropdown.xlsx#sheet4;pov=[HSP_View:BaseData],[Scenario:Forecast],
[Product:No Product],[Entity:Sales Mid-Atlantic];rows = [ Account ]; cols=
[Year, Period, Version]
```

Note

입력 CSV 파일에서 fileName으로 확인된 파일의 지정된 시트에는 임시 그리드 정의가 포함되어야 합니다. 예를 들어 이전 샘플 입력 파일은 임시 그리드 정의의 스스로 dropdown.xlsx의 sheet4를 지정합니다. 이 Excel 파일은 requirement.csv 및 입력 CSV 파일과 함께 [simulateConcurrentUsage](#) 명령을 실행하는 데 사용되는 INPUT_FILE.zip에서 사용할 수 있어야 합니다.

보고서 입력 파일 실행

관리 보고서의 열기를 지원하기 위해 requirement.csv에서 참조되는 이 파일은 열려는 보고서를 지정해야 합니다. 파일의 각 라인은 다음 형식이어야 합니다.

```
format=[REPORT_FORMAT];globalPov=[DIM 1:MEMBER 1],[DIM 2:MEMBER
2],..;prompts=[PROMPT 1:VALUE 1],[PROMPT 2:VALUE 2],..
```

globalPov 및 prompts는 선택사항입니다.

Note

- 지원되는 보고서 형식은 pdf 및 embedded입니다.
- globalPov 차원 및 해당 멤버의 이름에 콜론(:) 또는 세미콜론(;)이 포함되면 기호 앞에 이스케이프 문자 \를 사용합니다. 예를 들어 차원 이름 Version:View는 Version\
\:View로 지정해야 합니다.

글로벌 POV [Version View:Working],[Sales Entity:International Sales] 및 프롬프트 [Actual;Budget],[Year:2018]을 사용하여 보고서에서 PDF를 생성하기 위한 샘플 입력 파일:

```
format=pdf;globalPov=[Version View:Working],[Sales Entity:International Sales];prompts=[Actual:Budget],[Year:2018]
```

장부 입력 파일 실행

보고서에서 장부 열기를 지원하기 위해 requirement.csv에서 참조되는 이 파일은 열리는 장부를 지정해야 합니다. 파일의 각 라인은 다음 형식이어야 합니다.

```
format=BOOK_FORMAT 또는
```

```
format=BOOK_FORMAT;globalPov=[DIM 1:MEMBER 1],[DIM 2:MEMBER 2],...
```

글로벌 POV [Version View:Working],[Sales Entity:International Sales]를 사용하여 장부에서 PDF를 생성하기 위한 샘플 입력 파일:

```
format=pdf;globalPov=[Version View:Working],[Sales Entity:International Sales]
```

Note

- 지원되는 장부 형식은 PDF 및 XLSX입니다.
- globalPov 차원 및 해당 멤버의 이름에 콜론(:) 또는 세미콜론(;)이 포함되면 기호 앞에 이스케이프 문자 \를 사용합니다. 예를 들어 차원 이름 Version:View는 Version\
\:View로 지정해야 합니다.

UserVarMemberMapping.csv 파일 생성

양식 열기 또는 양식 저장 사용 사례의 입력 파일에 지정한 양식에서 사용자 변수를 설정해야 하는 경우 이 파일이 필요합니다. 다른 사용 사례에는 이 파일이 필요하지 않습니다.

- 테스트 모드 0 및 1에서 이 파일은 생성된 테스트 사용자에게 대한 사용자 변수 멤버 매핑을 설정하는 데 사용됩니다.
- 테스트 모드 4에서 이 파일은 기존 사용자에게 대한 사용자 변수 멤버 매핑을 설정하는 데 사용됩니다.

이 파일의 첫번째 라인은 헤더 #Dimension,User Variable,Member입니다. 후속 라인에는 차원, 사용자 변수 및 차원 멤버의 매핑이 포함됩니다.

샘플 UserVarMemberMapping.csv 파일:

```
#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
```

options.xml 파일 생성

Oracle Smart View for Office의 행 숨김, 누락된 블록 숨김, 페이지 멤버 들어쓰기와 같은 옵션은 양식 열기, 양식 저장 및 임시 그리드 사용 사례를 시뮬레이션하는 동안 사용할 수 있습니다. 이러한 각 옵션은 환경에 서로 다른 레벨의 로드를 부과합니다.

Smart View 옵션을 사용하여 로드를 시뮬레이션하려면 입력 ZIP 파일에 options.xml 파일을 포함할 수 있습니다.

Smart View 옵션을 익스포트하여 options.xml을 생성합니다. options.xml을 생성하려면 **SmartView** 탭에서 옵션을 누른 후 **확인**, **옵션 익스포트** 순으로 누릅니다.

options.xml 사용은 필수가 아닙니다. 입력 ZIP 파일에 이 파일이 없으면 시뮬레이션에는 기본 옵션이 사용됩니다.

users.csv 파일 생성

[simulateConcurrentUsage](#) 명령의 모드 4를 사용하면 입력 ZIP 파일에 포함된 users.csv 파일에서 확인되는 기존 사용자를 사용하여 명령을 실행할 수 있습니다. 이 모드는 시뮬레이션할 사용자를 생성하지 않습니다.

users.csv 파일의 형식은 loginname,password입니다.

```
jdoe,jdoe_pwd
john.doe@example.com,john_doe_pwd
```

users.csv의 loginname 값은 ID 도메인에 정의된 로그인 이름과 일치해야 합니다. 또한, 확인된 사용자는 작업을 실행하는 데 필요한 사전 정의된 역할 및 애플리케이션 역할이 있어야 합니다.

입력 ZIP 파일 생성 및 환경에 업로드

필요한 경우, 7 Zip과 같은 툴을 사용하여 requirement.csv, 해당 사용 사례 입력 파일, 선택사항으로 UserVarMemberMapping.csv, users.csv, options.xml이 포함된 ZIP 파일을 생성합니다.

[uploadFile](#) 명령을 사용하여 시뮬레이션을 실행하려는 해당 환경의 인박스에 결과 ZIP 파일을 업로드합니다(예제 명령 구문 epmautomate uploadFile "C:/uploads/INPUT_FILE.zip" inbox).

샘플 동시 사용 시뮬레이션 보고서

동시 사용 시뮬레이션은 기본적으로 [simulateConcurrentUsage](#) 명령을 실행하는 사용자에게 전송됩니다. 전자메일 수신자를 지정하면 보고서는 해당 전자메일 수신자에게만 전자메일로 전송됩니다.

Simulate Concurrent Usage report

Operation #	Operation	Artifact Name	Users	Iterations	Min. Duration	Max. Duration	Avg. Duration	Return Status
1	Open Form	Set Services Revenue Forecast Assumptions	2	1	00:00:00.17	00:00:00.21	00:00:00.19	Passed
2	Save Form	Set Headcount and Salary Forecast Assumptions	1	1	00:00:00.45	00:00:00.45	00:00:00.45	Passed
3	Run Business Rule	Operating Expense Adj Plan	2	1	00:00:00.60	00:00:00.61	00:00:00.60	Passed
4	Run Data Rule	test	2	1	00:00:03.00	00:00:03.29	00:00:03.14	Passed
5	Ad Hoc Grid		1	1	00:00:00.13	00:00:00.13	00:00:00.13	Passed
6	Execute Book	Book1	1	1	00:00:10.21	00:00:10.21	00:00:10.21	Passed
7	Execute Report	Variance Explanations1	1	1	00:00:04.95	00:00:04.95	00:00:04.95	Passed
8	Save Form	Set Services Revenue Forecast Assumptions	1	1	00:00:05.56	00:00:05.56	00:00:05.56	Passed
9	Run Business Rule	Allocate Plan Targets	2	1	00:00:02.62	00:00:02.62	00:00:02.62	Passed
10	Run Business Ruleset	Revenue Plan	2	1	00:00:00.60	00:00:00.61	00:00:00.61	Passed

이 보고서는 다음을 확인합니다.

열	설명
작업 #	requirement.csv에서 사용 사례의 순번
작업	requirement.csv에 지정된 작업 유형
아티팩트 이름	requirement.csv에 지정된 아티팩트 이름
사용자	requirement.csv에 지정된 사용자 수
반복	iterations 매개변수로 지정된 사용 사례 실행 횟수
최소 기간	한 사용자가 이 사용 사례를 실행하는 데 소요된 최소 시간
최대 기간	한 사용자가 이 사용 사례를 실행하는 데 소요된 최대 시간
평균 기간	한 사용자가 이 사용 사례를 실행하는 데 소요된 평균 시간
반환 상태	사용 사례의 상태. Failed는 사용 사례 실행이 실패한 경우 표시됨

B

Replay 명령 실행 준비

replay 명령은 서비스가 지정된 로드 상태인 경우에도 사용자 환경이 허용되는 상태인지 확인하는 로드 상태의 환경 성능을 테스트하는 데 사용됩니다. 테스트 환경을 로드하기 전에 몇 가지 단계를 완료해야 합니다.

이 부록에서는 서비스 관리자가 EPM Automate replay 명령을 실행하기 전에 완료해야 하는 단계를 설명합니다.

- [Replay 명령 정보](#)
- [필수 조건](#)
- [HAR 파일 생성](#)
- [재생 파일 생성](#)
- [추적 파일 생성](#)
- [샘플 재생 세션](#)

Replay 명령 정보

replay 명령은 환경이 지정된 로드 상태일 때 사용자 환경이 허용되는지 확인하기 위해 환경에서 Oracle Smart View for Office, Oracle Fusion Cloud Enterprise Performance Management REST API 또는 EPM Automate 로드를 재생하여 과도한 로드 상태에서 성능을 테스트할 수 있도록 합니다.

예를 들어 과도한 로드 상태의 테스트 환경에서 사용자 환경을 테스트하여 테스트 환경에서 프로덕션 환경으로 애플리케이션을 마이그레이션한 후 환경이 원활하게 작동할지 확인할 수 있습니다.

필수 조건

재생 파일을 사용하여 명령을 실행하는 경우 서비스가 과중한 로드 상태에 있을 때 사용자 환경이 허용되는지 확인하기 위해 테스트를 수행할 수 있도록 EPM Automate는 재생 파일의 각 행을 병렬로 실행하여 서비스에 로드를 부과합니다.

- 환경에서 주요 처리를 수행해야 하는 양식을 확인합니다. 대량 데이터를 다루는 양식, 복잡한 계산을 포함하는 양식 등이 있습니다. 예를 들어 임시 및 정적 보고서 생성에 관련된 예측 프로세스를 제출하는 데 사용되는 양식은 서비스에 과도한 로드를 부과할 수 있습니다. 마찬가지로, 비즈니스 규칙 실행, 보고서 실행, 리소스가 많이 소요되는 REST API 및 EPM Automate 명령(예: runBusiness rule, runDataRule, exportData, exportMetadata, restructureCube) 실행 같은 활동은 환경에 과도한 로드가 발생하도록 할 수 있으며 로드 테스트 대상이 되도록 할 수 있습니다.
- 필요한 경우 Fiddler를 설치합니다. EPM Automate에는 Cloud EPM 환경과 Oracle Smart View for Office, Oracle Fusion Cloud Enterprise Performance Management REST API 또는 EPM Automate의 상호 작용에 대한 레코드가 포함된 HAR(HTTP 아카이브 형식) 1.1 파일이 있어야 합니다. 일반적으로 Fiddler를 사용하여 Cloud EPM과의 상호 작용에 대한 로그를 캡처하는 HAR 파일을 생성합니다.
- 이전에 확인한 주요 활동을 실행합니다. Smart View를 사용하여 양식 열기 및 저장, 비즈니스 규칙 실행, 보고서 생성과 같은 활동을 실행하며 Fiddler를 사용하여 활동 세부정보를 캡처하고

HAR 파일로 익스포트합니다. 마찬가지로, REST API 및 EPM Automate 명령을 실행하고 Fiddler에서 세부정보를 캡처하도록 합니다. 자세한 내용은 [HAR 파일 생성](#)을 참조하십시오.

- 인증서(사용자 이름 및 비밀번호)와 실행할 HAR 파일의 이름을 나열하는 재생 CSV 파일을 생성합니다. 파일의 각 행에는 여러 개의 동시 사용자 세션을 시뮬레이트하기 위해 고유 사용자의 사용자 이름과 비밀번호가 포함될 수 있습니다. 자세한 내용은 [재생 파일 생성](#)을 참조하십시오.

HAR 파일을 실행하도록 행에서 해당 인증서가 지정된 사용자가 HAR 파일을 생성하는 데 사용된 세션을 실행한 사용자일 필요는 없습니다. 그러나 이 사용자에게 환경에서 해당 활동을 실행할 수 있는 권한이 있어야 합니다.

Replay 명령을 실행하는 자세한 단계는 [샘플 재생 세션](#)을 참조하십시오.

HAR 파일 생성

HAR 파일은 Oracle Fusion Cloud Enterprise Performance Management와 Oracle Smart View for Office, REST API 또는 EPM Automate의 상호 작용을 추적하여 캡처합니다.

Fiddler는 모든 HTTP(S) traffic에 대한 정보를 캡처하므로 HAR 파일을 생성하는 동안 불필요한 추적을 Fiddler에 추가할 수 있는 작업을 자제하십시오.

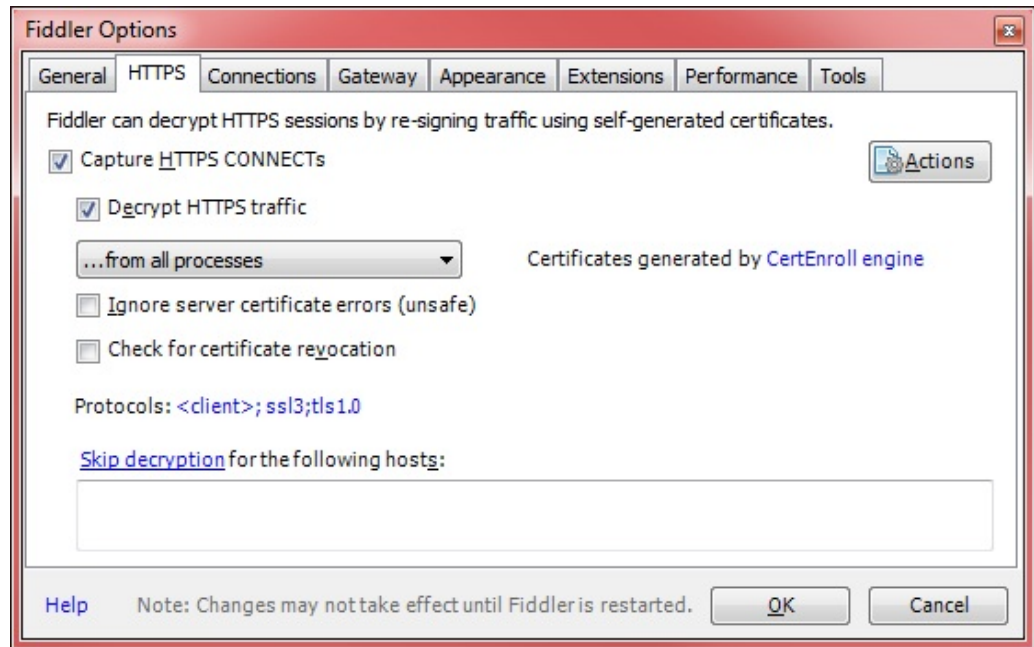
HAR 파일을 생성하려면 다음을 수행합니다.

1. Fiddler를 시작합니다.
2. Fiddler가 모든 프로세스의 HTTPS traffic을 해독하도록 구성되어 있는지 확인합니다.
 - a. **툴, 옵션, HTTPS** 순으로 선택합니다.
 - b. 선택되지 않은 경우 **Decrypt HTTPS traffic**을 선택합니다.

Fiddler는 HTTPS traffic을 가로채는 데 사용하는 루트 인증서에 대한 정보를 표시합니다. 일반적으로 이 인증서는 신뢰해도 안전합니다.

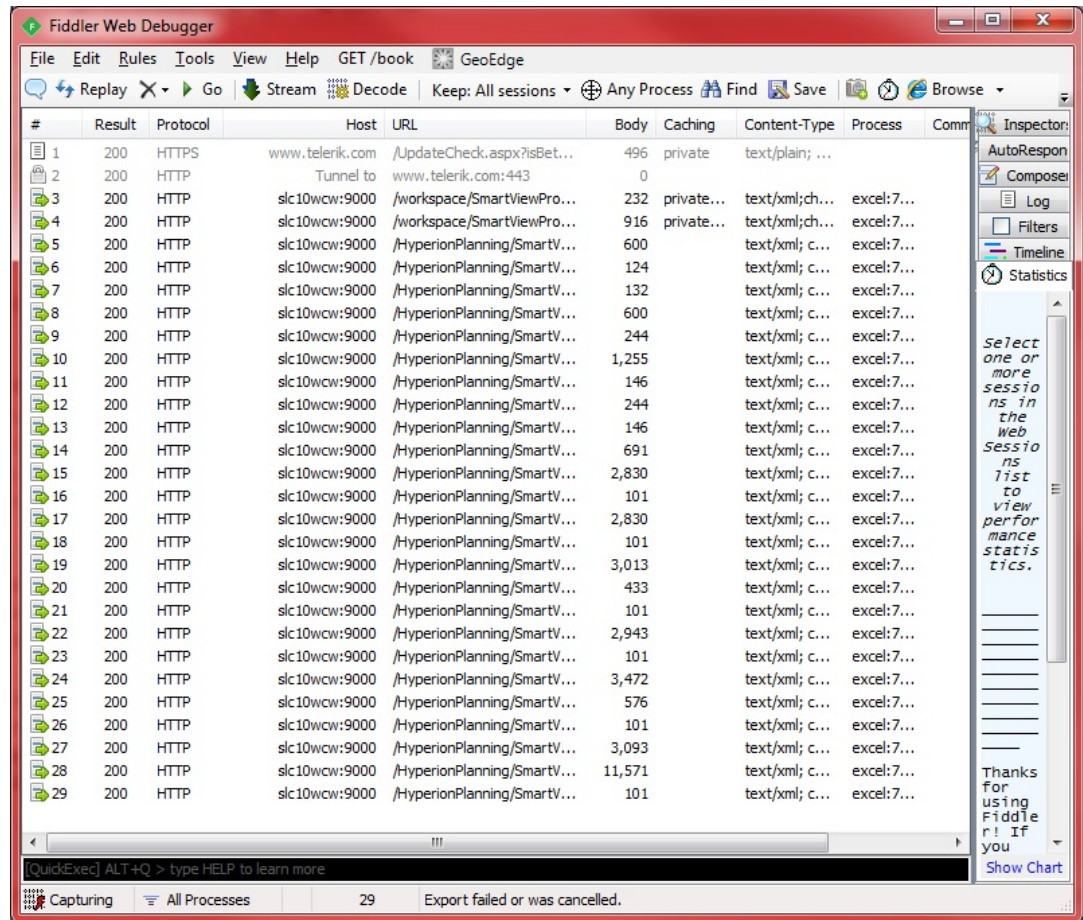


- c. 신뢰할 수 있는 CA 목록에 루트 인증서를 추가하려면 **예**를 누릅니다. 그렇지 않으면 **아니요**를 선택합니다.
- d. **선택사항:** 이전 단계에서 **No**를 선택한 경우 **Ignore server certificate errors**를 선택하여 HTTPS traffic 해독과 관련된 Fiddler 보안 경고를 제외할 수 있습니다.
- e. **확인**을 누릅니다.



3. Smart View를 시작하고 추적을 캡처하려는 환경에 액세스합니다.
4. Smart View, REST API 또는 EPM Automate를 사용하여 환경에 과도한 처리 로드를 부과하는 활동을 실행합니다. 예를 들어 Smart View에서 양식을 열어 Fiddler가 활동을 기록할 수 있도록 합니다.

Fiddler는 시작된 프로세스를 기록합니다.



5. Fiddler에서 다음 단계를 완료합니다.

- a. File, Export Sessions 순으로 선택한 다음 All Sessions 또는 Selected Sessions를 선택합니다. 다른 웹 사이트에 연결된 상태에서 Fiddler를 실행하는 경우 Selected Sessions를 선택하여 환경과 관련된 세션을 선택합니다.
- b. 익스포트 형식 선택에서 HTTPArchive v1.1을 익스포트 형식으로 선택합니다.
- c. 다음을 누릅니다.



- d. HTTPArchive v1.1로 익스포트에서 파일을 저장할 디렉토리를 선택하고 파일 이름을 지정합니다.
- e. 저장을 누릅니다.

재생 파일 생성

재생 파일은 EPM Automate replay 명령을 사용하여 시스템을 로드하기 위해 실행할 HAR 파일의 이름과 인증서(사용자 이름 및 비밀번호)를 나열하는 CSV 파일입니다.

지정한 사용자 이름과 비밀번호에 HAR 파일에 포함된 작업을 실행할 수 있는 권한이 있는지 확인합니다.

Replay 명령을 실행하면 EPM Automate는 재생 파일의 각 행을 병렬로 실행하여 서비스에 로드를 부과합니다. 예를 들어 재생 파일에 10개의 행이 있는 경우 서비스가 지정된 로드 상태에 있을 때 사용자 환경이 허용되는지 확인하기 위해 테스트를 수행할 수 있도록 EPM Automate에서 10개의 세션을 재생합니다. HAR 파일에 포함된 각 작업이 순차적으로 실행됩니다.

Replay 명령 실행에 대한 자세한 내용은 [replay](#) 를 참조하십시오.

재생 파일을 생성하려면 다음을 수행합니다.

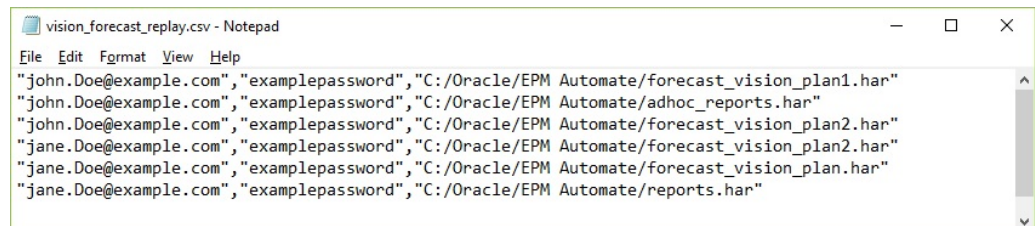
1. Microsoft Office Excel을 열고 새 워크시트를 시작합니다.
2. 행 1의 열 A, B, C에 각각 사용자 이름, 비밀번호 및 HAR 파일 위치를 입력합니다.
추가 행을 생성하려면 이 단계를 반복합니다.

① 주

HAR 파일 위치의 절대 경로를 지정해야 합니다. 파일 경로의 디렉토리 구분 문자로 슬래시(/)를 사용하고, 백슬래시(\)는 사용하지 마십시오.

3. 파일을 저장합니다.
4. 다른 이름으로 저장에서 다음 단계를 완료합니다.
 - a. 재생 파일을 저장할 디렉토리를 선택합니다.
 - b. 파일 이름에 이름을 지정하고, 파일 형식에서 CSV(쉼표로 분리) (*.csv)를 선택합니다.
 - c. 저장을 누릅니다.

샘플 재생 파일은 다음과 같을 수 있습니다.



```
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"
```

추적 파일 생성

Replay 명령을 실행하는 동안 이슈 해결을 위해 Oracle 지원 센터와 공유할 추적 파일을 생성할 수 있습니다. Oracle 지원 센터는 추적 파일을 사용하여 환경에서 Oracle Smart View for Office 활동을 어떻게 처리했는지 파악합니다.

Replay 명령에 선택적 `trace=true` 매개변수를 사용하여 XML 형식의 추적 파일을 생성합니다. 이 매개변수를 사용하는 경우 HAR 파일의 각 활동에 대해 EPM Automate는 활동에 대한 Smart View 응답을 포함하는 추적 파일을 생성합니다.

추적 파일 이름은 `trace-N.xml`로 지정됩니다(예: `trace-1.xml`). 여기서 N은 1부터 시작하는 카운터입니다. 재생 파일에 이름이 같은 HAR 파일을 여러 개 지정하면 EPM Automate에서 추적 파일을 한 폴더에 통합합니다.

HAR 파일과 관련된 추적 파일은 EPM Automate를 실행한 디렉토리 내의 폴더에 저장됩니다. EPM Automate는 재생 파일에 나열된 각 HAR 파일마다 하나의 폴더를 생성합니다. EPM Automate는 현재 서버 시스템 시간과 HAR 파일 이름의 조합을 `YYYY_MM_DD_HH_MM_SS_HAR_FILE_NAME` 형식으로 사용하여 폴더 이름을 지정합니다. 예를 들어 HAR 파일 이름이 `forecast1.har`이면 폴더 이름은 `2016_06_08_10_21_42_forecast1`일 수 있습니다.

샘플 재생 세션

여러 HAR 파일을 사용하여 `replay` 명령을 실행하는 방법에 대해 설명합니다.

이 섹션에서는 다음과 같이 가정합니다.

- 다음 HAR 파일을 생성했습니다. 각 HAR 파일에 동일한 작업 세트가 포함될 수 있습니다. 자세한 내용은 [HAR 파일 생성](#)을 참조하십시오.
 - `C:\Oracle\EPM Automate\forecast_vision_plan1.har`
 - `C:\Oracle\EPM Automate\forecast_vision_plan2.har`
 - `C:\Oracle\EPM Automate\forecast_plan2.har`
- 다음 콘텐츠를 사용하여 재생 파일 `C:/Oracle/EPM Automate/vision_forecast_replay.csv`를 생성했습니다(자세한 내용은 [재생 파일 생성](#) 참조).

① 주

재생 파일에서 파일 경로의 디렉토리 구분 문자로 슬래시(/)를 사용하고, 백슬래시(\)는 사용하지 마십시오.

```
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
```

`Replay` 명령을 실행하려면 다음을 수행합니다.

1. 명령 프롬프트 창에서 EPM Automate가 설치되어 있는 디렉토리(예: `C:\Oracle\EPM Automate\bin`)로 이동합니다.
2. 서비스 관리자로 환경에 로그인하고 다음과 같이 `replay` 명령을 실행합니다.

```
epmautomate login john.doe@example.com examplePassword https://test-cloud-
pln.pbc.us1.oraclecloud.com myIdentityDomain

epmautomate replay "c:/Oracle/EPM Automate/vision_forecast_replay.csv"
duration=12 lagTime=5.5 trace=true
```

EPM Automate는 콘솔에 재생 정보를 표시하고 지정된 기간(앞의 예에서는 12분) 후에 처리를 종료합니다. 또한 앞의 명령에는 `trace=true` 매개변수가 포함되어 있으므로 추적 폴더 및 파일을 생성합니다.

`C:\Oracle\EPM Automate\bin`에서 명령을 실행했기 때문에 EPM Automate는 다음 폴더에 추적 파일을 저장합니다. 이러한 폴더는 HAR 파일 이름을 따서 이름이 지정됩니다.

- 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. 환경에서 사인아웃합니다.

```
epmautomate logout
```

C

특수 문자 처리

Oracle Fusion Cloud Enterprise Performance Management 비밀번호, 프록시 비밀번호 및 명령 매개변수 값에는 특수 문자가 포함될 수 있습니다. EPM Automate에서 이러한 문자를 처리하려면 특수 처리가 필요합니다.

이 섹션의 예에서는 샘플 비밀번호를 사용하여 특수 문자 사용에 대해 설명합니다.

매개변수 및 값 쌍은 큰따옴표로 묶는 것이 좋습니다.

Windows

이러한 특수 문자는 해당 특수 문자 또는 특수 문자가 포함된 매개변수 값을 큰따옴표(")로 이스케이프해야 합니다.

① 주

이름에 &가 포함된 폴더(예: C:\Oracle\A&B)에서는 EPM Automate를 실행할 수 없습니다.

표 C-1 특수 문자 처리: Windows

문자	설명	이스케이프된 예
)	닫는 괄호	<ul style="list-style-type: none">Example") "pwd1 또는"Example)pwd1"
<	다음보다 작음	<ul style="list-style-type: none">Example"<"pwd1 또는"Example<pwd1"
>	다음보다 큼	<ul style="list-style-type: none">Example">"pwd1 또는"Example>pwd1"
&	앰퍼샌드	<ul style="list-style-type: none">Example"&"pwd1 또는"Example&pwd1"
	파이프	<ul style="list-style-type: none">Example" "pwd1 또는"Example pwd1"
"	따옴표	<ul style="list-style-type: none">Example""pwd1 또는"Example"pwd1"

Windows 배치 파일의 일반 텍스트 비밀번호에 느낌표 사용

EPM Automate에 사용되는 Windows 배치 파일의 일반 텍스트 비밀번호에 느낌표(!)를 사용하는 경우 다음과 같이 처리해야 합니다.

1. 느낌표 앞에 두 개의 캐럿(^)을 이스케이프 문자로 사용합니다. 예를 들어 비밀번호가 Welc0me!인 경우 Welc0me^^!로 인코딩합니다.
2. 다음 선언을 포함하여 파일의 시작 부분에서 DisableDelayedExpansion을 설정하도록 배치 파일을 업데이트합니다.
setlocal DisableDelayedExpansion

3. 스크립트에 `setlocal EnableExtensions EnableDelayedExpansion` 선언이 있는 경우 제거합니다.

UNIX/Linux

UNIX 및 Linux 운영 체제에서는 특수 문자를 백슬래시(\)로 이스케이프해야 합니다.

① 주

- ! (느낌표)를 이스케이프하려면 비밀번호를 작은따옴표로 묶거나 백슬래시(\)를 이스케이프 문자로 사용합니다.
- \, \$, ' 및 "를 이스케이프하려면 비밀번호를 큰따옴표로 묶거나 백슬래시(\)를 이스케이프 문자로 사용합니다.

표 C-2 특수 문자 처리: UNIX/Linux

문자	설명	이스케이프된 예
(여는 괄호	Example\(pwd1
)	닫는 괄호	Example\)pwd1
<	다음보다 작음	Example\ <pwd1< td=""> </pwd1<>
>	다음보다 큼	Example\>pwd1
`	아포스트로피	Example\'pwd1
!	느낌표	<ul style="list-style-type: none"> • 'Example!pwd1' 또는 • Example\!pwd1
#	해시	Example\#pwd1
&	앰퍼샌드	Example&pwd1
	파이프	Example\ pwd1
;	세미콜론	Example\;pwd1
.	기간	Example\.pwd1
"	따옴표	<ul style="list-style-type: none"> • Example\"pwd1 또는 • "Example\"pwd1"
'	작은따옴표	<ul style="list-style-type: none"> • Example\'pwd1 또는 • "Example\'pwd1"
\$	달러 기호	<ul style="list-style-type: none"> • Example\\$pwd1 또는 • "Example\\$pwd1"
\	백슬래시	<ul style="list-style-type: none"> • Example\\pwd1 또는 • "Example\\pwd1"

UNIX 또는 Linux 스크립트의 일반 텍스트 비밀번호에 느낌표 사용

UNIX/Linux 스크립트에서는 셸 변수에 저장된 EPM Automate 비밀번호에 특수 문자가 포함된 경우 세 개의 백슬래시를 이스케이프 시퀀스로 사용하고 해당 문자열을 큰따옴표로 묶습니다. 예를 들어 셸 변수 `password`에 포함된 비밀번호 `lzi[AC0(e*7Qd)jE`는 다음과 같이 스크립트가 지정되어야 합니다.

```
password="lzi[AC0\\(e*7Qd\\)jE"
```

D

개별 Cloud EPM 서비스 관련 명령

- [Account Reconciliation 명령](#)
- [Financial Consolidation and Close 명령](#)
- [Narrative Reporting 명령](#)
- [Cloud EDM 명령](#)
- [Planning, Planning 모듈, 무형식, Strategic Workforce Planning, Sales Planning 명령](#)
- [Profitability and Cost Management 명령](#)
- [Enterprise Profitability and Cost Management 명령](#)
- [Tax Reporting 명령](#)

Account Reconciliation 명령

Account Reconciliation용 EPM Automate 명령

addUsers	groupAssignmentAuditReport	replay
addUsersToGroup	help	resetService
addUsersToTeam	importARApplicationProperties	restoreBackup
addUserToGroups	importBackgroundImage	roleAssignmentAuditReport
archiveTmTransactions	importLogImage	roleAssignmentReport
assignRole	importBalances	runAutoAlert
cloneEnvironment	importDataManagement	runAutomatch
copyFileFromInstance	importMapping	runBatch
copyFromObjectStorage	importPreMappedBalances	runComplianceReport
copyFromSFTP	importPreMappedTransactions	runDailyMaintenance
copySnapshotFromInstance	importProfiles	runDataRule
copyToObjectStorage	importRates	runDMReport
copyToSFTP	importRCAAttributeValues	runIntegration
createGroups	importReconciliationAttributes	runMatchingReport
createReconciliations	importSnapshot	runPipeline
deleteFile	importTMAAttributeValues	sendMail
deleteGroups	importTmPremappedTransactions	setApplicationAdminMode
downloadFile	invalidLoginReport	setDailyMaintenanceStartTime
encrypt	listBackups	setDemoDates
exportAccessControl	listFiles	setEncryptionKey
exportARApplicationProperties	login	setIdleSessionTimeout
exportBackgroundImage	logout	setIPAllowlist(사용되지 않음)
exportDataManagement	provisionReport	setRestrictedDataAccess
exportLogImage	purgeArchivedTmTransactions	setManualDataAccess
exportMapping	purgeTmTransactions	setPeriodStatus
exportSnapshot	recreate	setVirusScanOnFileUploads
feedback	refreshCube	skipUpdate
getApplicationAdminMode	removeUserFromGroups	unassignRole
getDailyMaintenanceStartTime	removeUsers	updateUsers
getIdleSessionTimeout	removeUsersFromGroup	upgrade
getIPAllowlist(사용되지 않음)	removeUsersFromTeam	uploadFile
getRestrictedDataAccess	renameSnapshot	userAuditReport(사용되지 않음)
getVirusScanOnFileUploads		userGroupReport
		userLoginReport

Financial Consolidation and Close 명령

Financial Consolidation and Close용 EPM Automate 명령

addUsers	exportTaskManagerAccessControl	replay
addUsersToGroup	exportValidIntersections	resetService
addUsersToTeam	feedback	restoreBackup
addUserToGroups	getApplicationAdminMode	restructureCube
applicationAdminMode	getDailyMaintenanceStartTime	roleAssignmentAuditReport
assignRole	getEssbaseQryGovExecTime	roleAssignmentReport
clearDataByProfile	getIdleSessionTimeout	runBatch
cloneEnvironment	getIPAllowlist(사용되지 않음)	runBusinessRule
copyDataByProfile	getRestrictedDataAccess	runDailyMaintenance
copyFileFromInstance	getSubstVar	runDataRule
copyFromObjectStorage	getVirusScanOnFileUploads	runDMReport
copyFromSFTP	groupAssignmentAuditReport	runIntegration
copyOwnershipDataToNextYear	help	runRuleSet
copySnapshotFromInstance	importAppSecurity	runIntercompanyMatchingReport
copyToObjectStorage	importConsolidationJournals	runPipeline
copyToSFTP	importData	runSupplementalDataReport
createGroups	importDataManagement	runTaskManagerReport
deleteFile	importEJDimensionMembers	sendMail
deleteGroups	importJobConsole	setApplicationAdminMode
deployEJTemplates	importLibraryDocument	setDailyMaintenanceStartTime
deployFormTemplates	importMapping	setDemoDates
deployTaskManagerTemplate	importMetadata	setEJJournalStatus
downloadFile	importOwnershipData	setEncryptionKey
essbaseBlockAnalysisReport	importSDMDimensionMembers	setEssbaseQryGovExecTime
executeReportBurstingDefinition	importSnapshot	setIdleSessionTimeout
exportDataManagement	importSupplementalCollectionData	setIPAllowlist(사용되지 않음)
exportEssbaseData	importSupplementalData	setRestrictedDataAccess
encrypt	importValidIntersections	setVirusScanOnFileUploads
exportAppAudit	invalidLoginReport	setManualDataAccess
exportAppSecurity	listBackups	setSubstVars
exportConsolidationJournals	listFiles	simulateConcurrentUsage
exportData	login	skipUpdate
exportEJournals	logout	snapshotCompareReport
exportJobConsole	maskData	unassignRole
exportLibraryDocument	provisionReport	updateGuidedLearningSettings
exportMapping	recomputeOwnershipData	updateUsers
exportMetadata	recreate	upgrade
exportOwnershipData	refreshCube	uploadFile
exportSnapshot	removeUserFromGroups	userAuditReport(사용되지 않음)
exportTaskManagerAccessControl	removeUsers	userGroupReport
exportValidIntersections	removeUsersFromGroup	userLoginReport
	removeUsersFromTeam	

Financial Consolidation and Close용 EPM Automate 명령[exportSnapshot](#)[renameSnapshot](#)[validateConsolidationMetadata](#)

Narrative Reporting 명령

Narrative Reporting용 EPM Automate 명령[addUsers](#)[addUsersToGroup](#)[addUserToGroups](#)[assignRole](#)[cloneEnvironment](#)[copyFileFromInstance](#)[copyFromObjectStorage](#)[copyFromSFTP](#)[copyToObjectStorage](#)[copyToSFTP](#)[createGroups](#)[createNRSnapshot](#)[deleteFile](#)[deleteGroups](#)[downloadFile](#)[encrypt](#)[executeBurstDefinition](#)[exportLibraryArtifact](#)[feedback](#)[getDailyMaintenanceStartTime](#)[getIdleSessionTimeout](#)[getIPAllowlist\(사용되지 않음\)](#)[getRestrictedDataAccess](#)[getVirusScanOnFileUploads](#)[groupAssignmentAuditReport](#)[help](#)[importLibraryArtifact](#)[invalidLoginReport](#)[listBackups](#)[listFiles](#)[login](#)[logout](#)[provisionReport](#)[recreate](#)[refreshReportPackageDatasource](#)[removeUserFromGroups](#)[removeUsers](#)[removeUsersFromGroup](#)[replay](#)[resetService](#)[restoreBackup](#)[roleAssignmentAuditReport](#)[roleAssignmentReport](#)[runDailyMaintenance](#)[sendMail](#)[setDailyMaintenanceStartTime](#)[setEncryptionKey](#)[setIdleSessionTimeout](#)[setIPAllowlist\(사용되지 않음\)](#)[setManualDataAccess](#)[setRestrictedDataAccess](#)[setVirusScanOnFileUploads](#)[skipUpdate](#)[unassignRole](#)[updateUsers](#)[upgrade](#)[uploadFile](#)[userAuditReport\(사용되지 않음\)](#)[userGroupReport](#)[userLoginReport](#)

Cloud EDM 명령

Oracle Fusion Cloud Enterprise Data Management용 EPM Automate 명령

addUsers	getIdleSessionTimeout	replay
addUsersToGroup	getIPAllowlist(사용되지 않음)	resetService
addUserToGroups	getRestrictedDataAccess	restoreBackup
assignRole	getVirusScanOnFileUploads	roleAssignmentAuditReport
cloneEnvironment	groupAssignmentAuditReport	roleAssignmentReport
copyFileFromInstance	help	runDailyMaintenance
copyFromObjectStorage	importDimension	sendMail
copyFromSFTP	importSnapshot	setDailyMaintenanceStartTime
copySnapshotFromInstance	invalidLoginReport	setEncryptionKey
copyToObjectStorage	listBackups	setIdleSessionTimeout
copyToSFTP	listFiles	setIPAllowlist(사용되지 않음)
createGroups	loadDimensionViewpoint	setRestrictedDataAccess
deleteFile	loadViewpoint	setManualDataAccess
deleteGroups	login	setVirusScanOnFileUploads
downloadFile	logout	skipUpdate
encrypt	provisionReport	unassignRole
exportDimension	recreate	updateUsers
exportDimensionMapping	removeUserFromGroups	upgrade
exportSnapshot	removeUsers	uploadFile
extractDimension	removeUsersFromGroup	userAuditReport(사용되지 않음)
extractPackage	renameSnapshot	userGroupReport
feedback		userLoginReport
getDailyMaintenanceStartTime		

Planning, Planning 모듈, 무형식, Strategic Workforce Planning, Sales Planning 명령

Planning, Planning 모듈, 무형식, Strategic Workforce Planning, Sales Planning용 EPM Automate 명령

addUsers	feedback	restoreBackup
addUsersToGroup	getApplicationAdminMode	restructureCube
addUserToGroups	getDailyMaintenanceStartTime	roleAssignmentAuditReport
applicationAdminMode	getEssbaseQryGovExecTime	roleAssignmentReport
assignRole	getIdleSessionTimeout	runBatch
autoPredict * 각주 참조	getIPAllowlist(사용되지 않음)	runBusinessRule
clearCube	getRestrictedDataAccess	runDailyMaintenance
cloneEnvironment	getSubstVar	runDataRule
compactCube	getVirusScanOnFileUploads	runDMReport
copyFileFromInstance	groupAssignmentAuditReport	runIntegration
copyFromObjectStorage	help	runPipeline
copyFromSFTP	importAppAudit	runPlanTypeMap
copySnapshotFromInstance	importAppSecurity	runRuleSet
copyToObjectStorage	importCellLevelSecurity	sendMail
copyToSFTP	importData	setApplicationAdminMode
createGroups	importDataManagement	setDailyMaintenanceStartTime
deleteFile	importJobConsole	setDemoDates
deleteGroups	importLibraryDocument	setEncryptionKey
dismissIPMInsights **	importMapping	setEssbaseQryGovExecTime
downloadFile	importMetadata	setIdleSessionTimeout
enableQueryTracking	importSnapshot	setIPAllowlist(사용되지 않음)
encrypt	importValidIntersections	setManualDataAccess
essbaseBlockAnalysisReport	invalidLoginReport	setRestrictedDataAccess
executeAggregationProcess	listBackups	setSubstVars
executeReportBurstingDefinition	listFiles	setVirusScanOnFileUploads
exportAppAudit	login	simulateConcurrentUsage
exportAppSecurity	logout	skipUpdate
exportCellLevelSecurity	maskData	snapshotCompareReport
exportData	mergeDataSlices	sortMember
exportDataManagement	provisionReport	unassignRole
exportEssbaseData	recreate	updateGuidedLearningSettings
exportJobConsole	refreshCube	updateUsers
exportLibraryDocument	removeUserFromGroups	upgrade
exportMapping	removeUsers	uploadFile
exportMetadata	removeUsersFromGroup	userAuditReport(사용되지 않음)
exportSnapshot	renameSnapshot	userGroupReport
exportValidIntersections	replay	userLoginReport
	resetService	

Planning, Planning 모듈, 무형식, Strategic Workforce Planning, Sales Planning용 EPM Automate 명령

* 무형식, Strategic Workforce Planning, Sales Planning에는 이 명령이 지원되지 않습니다.

** 무형식에는 이 명령이 지원되지 않습니다.

Profitability and Cost Management 명령

Profitability and Cost Management용 EPM Automate 명령

addUsers	getDailyMaintenanceStartTime	renameSnapshot
addUsersToGroup	getEssbaseQryGovExecTime	replay
addUserToGroups	getIdleSessionTimeout	resetService
applyDataGrants	getIPAllowlist(사용되지 않음)	restoreBackup
assignRole	getRestrictedDataAccess	roleAssignmentAuditReport
clearPOV	getVirusScanOnFileUploads	roleAssignmentReport
cloneEnvironment	groupAssignmentAuditReport	runBatch
copyFileFromInstance	help	runCalc
copyFromObjectStorage	importDataManagement	runDailyMaintenance
copyFromSFTP	importMapping	runDataRule
copyPOV	importSnapshot	runDMReport
copySnapshotFromInstance	importTemplate	runIntegration
copyToObjectStorage	invalidLoginReport	sendMail
copyToSFTP	listBackups	setDailyMaintenanceStartTime
createGroups	listFiles	setEncryptionKey
deleteFile	loadData	setEssbaseQryGovExecTime
deleteGroups	loadDimData	setIdleSessionTimeout
deletePOV	login	setIPAllowlist(사용되지 않음)
deployCube	logout	setRestrictedDataAccess
downloadFile	mergeSlices	setManualDataAccess
enableApp	optimizeASOCube	skipUpdate
encrypt	programDocumentationReport	setVirusScanOnFileUploads
exportDataManagement	provisionReport	unassignRole
exportMapping	recreate	updateUsers
exportQueryResults	removeUserFromGroups	upgrade
exportSnapshot	removeUsers	uploadFile
exportTemplate	removeUsersFromGroup	userAuditReport(사용되지 않음)
feedback		userGroupReport
		userLoginReport

Enterprise Profitability and Cost Management 명령

Enterprise Profitability and Cost Management용 EPM Automate 명령

addUsers	exportMetadata	renameSnapshot
addUsersToGroup	exportSnapshot	replay
addUserToGroups	exportTaskManagerAccessControl	resetService
applicationAdminMode	exportValidIntersections	restoreBackup
assignRole	feedback	roleAssignmentAuditReport
calculateModel	getApplicationAdminMode	roleAssignmentReport
clearCube	getDailyMaintenanceStartTime	runBatch
compactCube	getEssbaseQryGovExecTime	runBusinessRule
copyDataByPointOfView	getIdleSessionTimeout	runDailyMaintenance
cloneEnvironment	getIPAllowlist(사용되지 않음)	runDataRule
copyFileFromInstance	getRestrictedDataAccess	runDMReport
copyFromObjectStorage	getSubstVar	runIntegration
copyFromSFTP	getVirusScanOnFileUploads	runPipeline
clearDataByPointOfView	groupAssignmentAuditReport	sendMail
copySnapshotFromInstance	help	runRuleSet
copyToObjectStorage	importAppAudit	runTaskManagerReport
copyToSFTP	importAppSecurity	setApplicationAdminMode
createGroups	importCellLevelSecurity	setDailyMaintenanceStartTime
deleteFile	importData	setDemoDates
deleteGroups	importDataManagement	setEncryptionKey
downloadFile	importJobConsole	setEssbaseQryGovExecTime
deletePointOfView	importLibraryDocument	setIdleSessionTimeout
deployTaskManagerTemplate	importMapping	setIPAllowlist(사용되지 않음)
enableQueryTracking	importMetadata	setManualDataAccess
encrypt	importSnapshot	setRestrictedDataAccess
executeAggregationProcess	importValidIntersections	setSubstVars
executeReportBurstingDefinition	invalidLoginReport	setVirusScanOnFileUploads
exportAppAudit	listBackups	skipUpdate
exportAppSecurity	listFiles	snapshotCompareReport
exportCellLevelSecurity	login	sortMember
exportData	logout	unassignRole
exportDataManagement	maskData	updateGuidedLearningSettings
exportEssbaseData	mergeDataSlices	updateUsers
exportJobConsole	provisionReport	upgrade
exportLibraryDocument	recreate	uploadFile
exportMapping	refreshCube	userAuditReport(사용되지 않음)
exportMetadata	removeUserFromGroups	userGroupReport
	removeUsers	userLoginReport
	removeUsersFromGroup	validateModel

Tax Reporting 명령

Tax Reporting용 EPM Automate 명령

addUsers	getApplicationAdminMode	replay
addUsersToGroup	getDailyMaintenanceStartTime	resetService
addUsersToTeam	getEssbaseQryGovExecTime	restoreBackup
addUserToGroups	getIdleSessionTimeout	restructureCube
applicationAdminMode	getIPAllowlist(사용되지 않음)	roleAssignmentAuditReport
assignRole	getRestrictedDataAccess	roleAssignmentReport
clearDataByProfile	getSubstVar	runBatch
copyDataByProfile	getVirusScanOnFileUploads	runBusinessRule
copyFileFromInstance	groupAssignmentAuditReport	runDailyMaintenance
copyFromObjectStorage	help	runDataRule
copyFromSFTP	importAppSecurity	runDMReport
copyOwnershipDataToNextYear	importCellLevelSecurity	runIntegration
copySnapshotFromInstance	importData	runPipeline
copyToObjectStorage	importDataManagement	runRuleSet
copyToSFTP	importJobConsole	runSupplementalDataReport
createGroups	importLibraryDocument	runTaskManagerReport
deleteFile	importMapping	sendMail
deleteGroups	importMetadata	setApplicationAdminMode
deployFormTemplates	importOwnershipData	setDailyMaintenanceStartTime
deployTaskManagerTemplate	importSDMDimensionMembers	setDemoDates
downloadFile	importSnapshot	setEncryptionKey
encrypt	importSupplementalCollectionData	setEssbaseQryGovExecTime
essbaseBlockAnalysisReport	importSupplementalData	setIdleSessionTimeout
executeReportBurstingDefinition	importValidIntersections	setIPAllowlist(사용되지 않음)
exportAppAudit	invalidLoginReport	setManualDataAccess
exportCellLevelSecurity	listBackups	setRestrictedDataAccess
exportData	listFiles	setSubstVars
exportDataManagement	login	setVirusScanOnFileUploads
exportEssbaseData	logout	simulateConcurrentUsage
exportJobConsole	maskData	skipUpdate
exportLibraryDocument	provisionReport	snapshotCompareReport
exportMapping	recomputeOwnershipData	unassignRole
exportMetadata	recreate	updateGuidedLearningSettings
exportOwnershipData	refreshCube	upgrade
exportSnapshot	removeUserFromGroups	updateUsers
exportTaskManagerAccessControl	removeUsers	uploadFile
exportValidIntersections	removeUsersFromGroup	userAuditReport(사용되지 않음)
feedback	removeUsersFromTeam	userGroupReport
	renameSnapshot	userLoginReport