

# Oracle® Hyperion Financial Management 관리자 가이드



릴리스 11.2.13

F26659-14

2023년 6월

ORACLE®

F26659-14

Copyright © 2000, 2023, 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, and MySQL 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 Financial Management 정보

---

Financial Management 기능	1-1
EPM Workspace	1-1
Financial Management 차원	1-2
시나리오 차원	1-2
연도 차원	1-2
기간 차원	1-2
엔티티 차원	1-3
값 차원	1-3
계정 차원	1-3
내부거래 차원	1-3
뷰 차원	1-3
사용자정의 차원	1-3
사용자 정의 요소	1-4

## 2 애플리케이션 관리

---

애플리케이션 관리	2-1
애플리케이션 생성	2-2
새 애플리케이션 생성	2-2
빈도 정의	2-4
기간 편집	2-5
동위 멤버 및 1차 하위 구성요소 기간 추가	2-6
기간 삭제	2-6
기능 정의	2-7
애플리케이션 설정 지정	2-7
애플리케이션 모듈 사용 또는 사용 안함	2-8

사용자정의 차원 생성	2-9
애플리케이션 프로파일 저장	2-10
프로파일 파일에서 애플리케이션 생성	2-11
애플리케이션에서 복사	2-11
애플리케이션 수정	2-11
애플리케이션 등록	2-12
애플리케이션 열기	2-13
애플리케이션 닫기	2-13
애플리케이션 서버 클러스터 변경	2-13
애플리케이션 목록 보기	2-13
애플리케이션 삭제	2-14
애플리케이션 요소 로드	2-14
애플리케이션 요소 추출	2-15
Financial Management 애플리케이션 복제	2-16
애플리케이션 임포트	2-16
Oracle 데이터베이스에 대한 데이터베이스 링크 생성	2-18
SQL Server 데이터베이스의 링크된 서버 생성	2-18
시스템 메시지 작업	2-20
시스템 메시지 보기	2-20
시스템 메시지 삭제	2-20
시스템 메시지 세부정보 문자열	2-21
애플리케이션 액세스 관리	2-21
사용자 로그아웃	2-22
연결 비활성화 및 활성화	2-22
애플리케이션 연결 상태 보기	2-23
시스템 사용자 관리	2-23
사용자 보기	2-23
선택한 사용자 또는 모든 사용자 로그아웃	2-24
서버 및 애플리케이션 관리	2-25
관리 모드 활성화 및 비활성화	2-25
서버 동기화	2-25
태스크 감사	2-25
데이터 감사	2-28
실행 중인 태스크 모니터	2-29
실행 중인 태스크 보기	2-30
실행 중인 태스크 중지	2-31
실행 중인 태스크 새로고침	2-31
부적합한 레코드 스캔 및 지우기	2-32
HFM 인사이트를 사용하여 시스템 상태 모니터	2-32
HFM 인사이트 액세스	2-32
HFM 인사이트 기본 대시보드	2-33

HFM 인사이드에서 애플리케이션 필터링	2-33
HFM 인사이드의 애플리케이션 세부정보	2-34

### 3 애플리케이션 보안 관리

---

애플리케이션 보안 고려 사항	3-1
Financial Management에서 Shared Services Console 실행	3-2
보안 클래스를 지정할 사용자 및 그룹 선택	3-2
보안 클래스 설정	3-3
보안 클래스 생성	3-3
보안 클래스 삭제	3-4
보안 클래스 선택	3-4
보안 클래스에 대한 사용자 액세스 권한 지정	3-5
전자메일 알림 설정	3-6
보안 보고서 실행	3-7
애플리케이션 보안 로드	3-7
보안 정보 지우기 및 로드	3-9
보안 정보를 지우기 전에 수행할 작업	3-9
보안 정보를 지운 후 수행할 작업	3-10
애플리케이션 보안 추출	3-11

### 4 메타데이터 관리

---

계정 정의	4-1
계정 유형 동작	4-5
동적 계정 정의	4-7
사용자정의 멤버 정의	4-7
엔티티 멤버 정의	4-9
시나리오 멤버 정의	4-11
애플리케이션 설정 정의	4-13
기간별 조직	4-16
통합 메소드 정의	4-17
통합 메소드 사용	4-18
수동으로 통합 메소드 지정	4-19
POWN 또는 POWNMIN 통합 메소드 사용	4-19
통화 정의	4-21
셀 텍스트 레이블 정의	4-22
시스템 생성 계정	4-25
통합 계정	4-25
소유권 계정	4-26
시스템 생성 계정 편집	4-26

내부거래 파트너 설정	4-27
시스템 생성 ICP 멤버 편집	4-28
시스템 생성 값 멤버 편집	4-28
보안을 기반으로 하는 메타데이터 필터링	4-29
APP 형식의 메타데이터 파일 생성	4-29
파일 형식	4-31
버전	4-31
애플리케이션 설정	4-31
통화	4-33
멤버	4-33
계정	4-34
시나리오	4-34
엔티티	4-35
사용자정의	4-35
값	4-36
내부거래 파트너	4-37
통합 메소드	4-38
계층	4-38
계정 계층	4-39
시나리오 계층	4-41
엔티티 계층	4-41
사용자정의 계층	4-41
차원이 메타데이터 파일에 없음	4-42
메타데이터 참조 무결성	4-42
메타데이터 참조 무결성 검사	4-43
메타데이터 로그 파일 참조 무결성 오류	4-43
메타데이터 병합 유틸리티 사용	4-43
메타데이터 로드	4-45
메타데이터 로드 변경사항 보기	4-47
메타데이터 추출	4-49

## 5    멤버 목록 관리

---

멤버 목록 파일 생성	5-1
EnumMemberLists	5-2
EnumMembersInList	5-3
동적 멤버 목록	5-6
동적 POV 멤버 목록	5-7
멤버 목록 로드	5-10
멤버 목록 추출	5-11

## 6 분개 관리

---

분개 파일 생성	6-1
파일 형식 섹션	6-2
버전 섹션	6-2
분개 그룹 섹션	6-2
표준 섹션	6-3
반복 섹션	6-3
머리글 섹션	6-3
분개 로드	6-5
분개 추출	6-7

## 7 데이터 양식 관리

---

양식 디자이너에서 데이터 양식 생성	7-1
POV 설정	7-2
양식 세부정보 지정	7-3
데이터 양식에 대해 주문형 규칙 지정	7-3
양식 행 및 열 옵션 지정	7-4
양식 머리글 지정	7-4
데이터 양식 스크립트 사용	7-5
AddMember	7-8
BackgroundPOV	7-9
Blank	7-11
Cn	7-11
Calc1	7-12
CalcByRow	7-13
Cell_Link	7-14
CellText	7-15
CustomHeader	7-15
CustomHeaderStyle	7-16
DynamicPOV	7-17
FormInputBoxLength	7-17
FormNumDecimals	7-17
FormRowHeight	7-18
FormScale	7-18
HeaderOption	7-19
HideInPov	7-20
Instructions	7-20

LineItemDetailSinglePeriod	7-21
Link	7-21
MaxCells	7-21
MaxColsForSparseRetrievalMethod	7-22
NoSuppress	7-22
NumDecimals	7-23
OnDemandRules	7-23
Override	7-23
POVOrder	7-25
PrintNumDataColsPerPage	7-25
PrintNumRowsPerPage	7-26
PrintRepeatHeadersonAllPages	7-26
Rn	7-26
ReadOnly	7-28
ReportDescription	7-28
ReportLabel	7-29
ReportSecurityClass	7-29
ReportType	7-30
RowHeaderPct	7-30
SCalc	7-30
Scale	7-32
SelectablePOVList	7-32
ShowDescriptions	7-33
ShowLabels	7-34
String	7-34
Style	7-34
SuppressColHeaderRepeats	7-36
SuppressInvalidCols	7-37
SuppressInvalidRows	7-37
SuppressNoDataCols	7-38
SuppressNoDataRows	7-38
SuppressRowHeaderRepeats	7-38
SuppressZeroCols	7-39
SuppressZeroRows	7-39
상대 기간 사용	7-39
충돌하는 속성의 우선 순위	7-40
데이터 양식 편집	7-41
데이터 양식 로드	7-41
데이터 양식 추출	7-42
데이터 양식 삭제	7-43



## 8 데이터를 데이터베이스로 추출

---

DSN(데이터 소스 이름) 구성	8-1
스타 스키마	8-2
스타 스키마 형식	8-3
접두어 테이블	8-4
데이터 생성 및 스타 스키마로 익스포트	8-4
스타 스키마 업데이트	8-6
스타 스키마 삭제	8-7
스타 스키마 템플릿 생성	8-8
스타 스키마 템플릿 삭제	8-8

## 9 보고서 정의

---

분개 보고서 스크립트 정의	9-1
내부거래 매칭 보고서 스크립트 정의	9-2
내부거래 매칭 보고서의 멤버 목록 선택	9-3
내부거래 매칭 보고서의 계정 선택	9-3
내부거래 매칭 보고서의 소수 자릿수 지정	9-3
내부거래 매칭 보고서의 스타일시트 선택	9-3
내부거래 매칭 보고서의 통화 지정	9-3
내부거래 매칭 보고서의 제외 옵션	9-4
내부거래 매칭 보고서 스크립트 키워드	9-4
ReportType	9-5
ReportDescription	9-5
StyleSheet	9-5
POV	9-5
엔티티	9-6
Partner	9-6
AccountEntity 및 AccountPartner	9-7
SuppressIfMatch	9-9
SuppressReversedTransactions	9-10
SuppressDetails	9-10
MatchingTolerance	9-11
SuppressCustoms	9-11
ScaleFactor	9-11
소수 자릿수	9-12
DisplayLabels	9-12
DisplayDescriptions	9-13
DisplayPlugElimAccts	9-13
GroupByCustom	9-13

## 10 규칙 관리

규칙 유형	10-1
규칙 고려 사항	10-4
계산 명령	10-4
강제 계산	10-5
계산 명령이 포함된 계산 규칙	10-5
통합 명령이 포함된 계산 규칙	10-6
현재 차원 멤버	10-9
계정 표현식	10-10
함수를 통해 자동으로 데이터 지우기	10-11
오류 메시지	10-11
통합 중 규칙 실행	10-12
기본 환산	10-13
Financial Management 객체	10-13
규칙에서 VBScript 사용	10-14
VBScript 연산자	10-14
VBScript 문	10-15
VBScript 키워드	10-16
VBScript 함수	10-16
날짜 및 시간 함수	10-16
형식 함수	10-17
변환 함수	10-17
수리 함수	10-17
배열 함수	10-17
문자열 함수	10-18
기타 함수	10-18
VBScript 객체	10-19
일반적으로 사용되는 규칙	10-19
데이터 재사용	10-20
금액을 계산하여 계정 설정	10-20
조건부 규칙	10-21
차원 멤버 테스트	10-21
둘 이상의 멤버 테스트	10-22
각 멤버에 대해 서로 다른 작업 수행	10-22
모든 계정의 기초 잔액 설정	10-23
규칙 파일 생성	10-24
규칙 로드	10-26
규칙 추출	10-27

규칙 프로파일링	10-28
규칙 파일에 프로파일 정보 추가	10-28
규칙 프로파일링 시작 및 중지	10-29
규칙 프로파일 결과 보기	10-29
규칙 뷰	10-30
엔티티 뷰	10-30
규칙 프로파일 삭제	10-31

## 11 규칙 함수

---

함수 개요	11-1
ABSExp	11-9
AccountType	11-9
AccountTypeID	11-11
AddEntityToList	11-12
AddEntityToListUsingIDs	11-13
AddMemberToList	11-13
AddMemberToListUsingIDs	11-13
Alloc	11-14
AllowAdjFromChildren	11-15
AllowAdjs	11-16
ApplicationName	11-17
CalculateExchangeRate	11-18
CalculateRate	11-18
CalcStatus	11-19
CellTextUnitItem	11-21
Clear	11-23
Con	11-24
Consol1, Consol2, Consol3	11-25
ConsolidateYTD	11-26
ContainsCellText	11-26
ContainsCellTextWithLabel	11-28
통화	11-29
CustomTop	11-29
DataUnitItem	11-30
소수 자릿수	11-31
DefaultFreq	11-32
DefaultParent	11-33
DefaultTranslate	11-34
DefaultView	11-35
DefCurrency	11-36

Down	11-37
Dynamic	11-38
Exp	11-39
차원 교차점 고려 사항	11-41
기간 및 연도 키워드	11-41
수치 계산	11-42
Exp 안에 다른 함수 배치	11-43
동시에 여러 계정 설정	11-43
표현식 및 차원 교차 고려 사항	11-43
모든 교차점 적합	11-44
부적합한 교차점	11-45
오른쪽에 하나의 적합한 멤버 지정	11-45
GetCell	11-46
GetCellNoData	11-47
GetCellRealData	11-48
GetCellText	11-49
GetCellTextWithLabel	11-50
GetCellType	11-51
GetCustomLabelArray	11-52
GetItem	11-52
GetItemIDs2	11-53
GetItemIDs2ExtDim	11-54
GetNumItems	11-55
GetNumLID	11-56
GetRate	11-57
GetSubmissionGroup	11-57
GetSubmissionPhase	11-58
Holding	11-58
ICPTopMember	11-59
ICPWeight	11-60
IDFromMember	11-61
ImpactStatus	11-62
Input	11-63
IsAlmostEqual	11-64
IsBase	11-65
IsCalculated	11-67
IsChild	11-68
IsConsolidated	11-70
IsDescendant	11-71
IsFirst	11-73
IsICP	11-74

IsLast	11-75
IsTransCur	11-76
IsTransCurAdj	11-76
IsValidDest	11-77
IsZero	11-78
List	11-80
Member	11-81
MemberFromID	11-82
Method	11-84
NoInput	11-84
NoRound	11-85
NumBase	11-86
Number	11-88
NumChild	11-89
NumCustom	11-91
NumDescendant	11-91
NumPerInGen	11-93
NumPeriods	11-94
OpenCellTextUnit	11-95
OpenDataUnit	11-96
OpenDataUnitSorted	11-96
Owned	11-97
Owner	11-97
PCon	11-98
PEPU	11-98
PeriodNumber	11-99
PlugAcct	11-100
POwn	11-101
PVAForBalance	11-102
PVAForFlow	11-103
RateForBalance	11-103
RateForFlow	11-104
ReviewStatus	11-105
ReviewStatusUsingPhaseID	11-106
Round	11-107
Scale	11-108
SecurityAsPartner	11-109
SecurityClass	11-110
SetCellTextWithLabel	11-112
SetData	11-112
SetDataWithPOV	11-113

SubmissionGroup	11-114
SupportsProcessManagement	11-115
SupportsTran	11-116
SwitchSign	11-116
SwitchType	11-118
Trans	11-119
TransPeriodic	11-120
UD1...3	11-121
ValidationAccount	11-123
ValidationAccountEx	11-123
XBRLTags	11-124

## 12 사용자정의 함수

---

관리 보고 함수	12-2
평균	12-2
Cumulative	12-5
Difference	12-8
DSO - 매출 회전 일수	12-11
Opening	12-14
Rate	12-17
비즈니스 규칙 함수	12-24
Custom_Alloc	12-24
Increase_Decrease	12-28
Pro_Rata_Ratio	12-30
Spread	12-32
Units_Rates	12-35

## 13 Calculation Manager를 사용하여 규칙 생성

---

Calculation Manager 보안 역할	13-1
Calculation Manager의 애플리케이션 작업	13-1
규칙을 Calculation Manager로 마이그레이션	13-2
함수 선택기의 VB 함수 지원	13-2
Financial Management에 대한 특수 VB 스크립트 함수	13-3

## 14 내부거래 트랜잭션 관리

---

내부거래 트랜잭션 설정	14-1
내부거래 기간 열기	14-1
일치 허용오차 설정	14-2

계시 전에 일치/검증을 수행하시겠습니까? 옵션 설정	14-3
내부거래 기간 닫기	14-3
원인 코드 관리	14-4
원인 코드 추가	14-4
원인 코드 편집	14-5
원인 코드 삭제	14-5
내부거래 트랜잭션 모니터	14-5
엔티티 잠금 및 잠금 해제	14-7
내부거래 트랜잭션 요약 보기	14-8

## 15 프로세스 관리 제출 단계 관리

---

제출 단계 정의	15-1
제출 그룹 설정	15-2
제출 그룹 및 단계 예	15-3
단계에 제출 그룹 지정	15-4
지정되지 않은 제출 그룹 보기	15-5

## 16 전자메일 알림 관리

---

프로세스 관리 알림 설정	16-1
내부거래 알림 설정	16-2

### A 구성 설정

---

사용가능 구성 설정	A-1
구성 설정 변경	A-7
값 대체	A-8
설정 테이블 표시 변경	A-8
설정 검색	A-9
유효 설정 보기	A-9
설정 익스포트	A-9
설정 삭제	A-10

### B 성능 최적화

---

성능 개요	B-1
공통 EPM 설치 디렉토리 참조	B-1
Financial Management 레코드 및 서브큐브	B-2
Financial Management에 대한 조정 권장 사항	B-3
성능 문제 진단	B-3

모니터링 툴 사용	B-3
RDA(원격 진단 에이전트) 사용	B-5
참조 애플리케이션 사용	B-5
운영 체제 매개변수 조정	B-6
Windows 매개변수 조정	B-6
웹 서버 조정	B-7
HFM 웹 조정	B-7
HFM 웹 시간 초과 매개변수	B-7
HFM 웹 조정 매개변수	B-8
Financial Management 애플리케이션 조정	B-10
일반적으로 조정되는 Financial Management 설정	B-10
MaxNumDataRecordsInRAM	B-10
MinDataCacheSizeInMB	B-10
MaxDataCacheSizeInMB	B-10
MaxNumCubesInRAM	B-11
IcmSystemReportTranslationMode	B-11
NumConsolidationThreads	B-11
MaxNumConcurrentConsolidations	B-12
NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged	B-13
대규모 애플리케이션에 대한 Financial Management 메모리 설정	B-13
애플리케이션별 설정	B-13
Financial Management 애플리케이션 서버 조정	B-14
Financial Management 조정 예	B-15
애플리케이션 데이터베이스 유지관리	B-16
기본 디자인 고려 사항	B-16
Financial Management용 Oracle 11g 데이터베이스 조정	B-17
소개	B-17
일반적인 성능 이슈	B-18
Oracle 11g 데이터베이스에 대한 조정 지침	B-18
Oracle 초기화 매개변수	B-18
CURSOR_SHARING	B-18
MEMORY_TARGET	B-18
MEMORY_MAX_TARGET	B-19
SGA_TARGET	B-19
SGA_MAX_SIZE	B-19
PGA_AGGREGATE_TARGET	B-20
LOG_BUFFER	B-20
OPTIMIZER_MODE	B-20
OPTIMIZER_INDEX_COST_ADJ	B-20
PROCESSES	B-20
SESSIONS	B-21



TRANSACTIONS	B-21
OPEN_CURSORS	B-21
SESSION_CACHED_CURSORS	B-21
TRACE_ENABLED	B-21
STATISTICS_LEVEL	B-21
TIMED_STATISTICS	B-22
TIMED_OS_STATISTICS	B-22
Oracle 데이터베이스 릴리스 11g에 대한 메모리 설정을 결정하는 방법	B-22
총 메모리 크기 조정(MEMORY_TARGET)	B-22
SGA 크기 조정(SGA_TARGET)	B-24
PGA 크기 조정(PGA_AGGREGATE_TARGET)	B-24
LOG_BUFFER 크기 조정	B-25
Oracle 데이터베이스 릴리스 11g에 대한 프로세스 수를 계산하는 방법	B-25
기타 고려 사항	B-26
공유 서버 및 전용 서버	B-26
온라인 재실행 로그 파일 크기	B-26
테이블 영역 및 세그먼트 조각화	B-27
인덱스 조각화	B-27
DEFERRED_SEGMENT_CREATION 기능을 비활성화합니다.	B-27
DEFERRED_SEGMENT_CREATION 기능을 비활성화합니다.	B-27
정기 유지 관리 및 조정 계획	B-28
자주 묻는 질문(FAQ)	B-28

# 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.com](mailto:epmdoc_ww@oracle.com)으로 전자메일을 보낼 수도 있습니다.

# 1

## Financial Management 정보

### 참조:

- [Financial Management 기능](#)
- [EPM Workspace](#)
- [Financial Management 차원](#)
- [사용자 정의 요소](#)

## Financial Management 기능

Oracle Hyperion Financial Management에서는 다음 기능을 제공합니다.

- 하나의 통합된 엔터프라이즈 재무 정보 뷰를 통해 글로벌 소스의 주요 성과 지표 및 운영 지표를 확장 가능한 웹 기반 애플리케이션에 통합합니다.
- "빠른 가상 마감" 기능은 웹 기반 프로세스 관리, 웹 기반 내부거래 조정, 분개 조정 및 일관성 있는 데이터 및 비즈니스 측정항목 세트를 사용하여 마감 주기를 며칠에서 몇 주까지 단축시킵니다.
- 강력한 다차원 분석을 통해 회사, 원가 중심점, 제품, 브랜드, 고객 및 채널 레벨에서 주요 재무 및 비즈니스 추세, 새로운 수익 출처 및 현금 흐름을 식별하고 보고합니다.
- 유연성 있는 "What If" 시나리오 관리 기능은 실제 결과, 재무 예산, 예측 및 계획을 동적으로 통합하고 보고하여 가정과 팩트가 변경됨에 따라 새로운 명령문을 생성합니다.
- 서식이 미리 지정된 대량 보고서는 내부 관리와 외부 규제 기관 및 정부 기관을 위한 정확한 재무 정보를 동일한 애플리케이션에서 적시에 제공합니다.
- 레거시 애플리케이션, ERP 및 CRM 시스템과의 강력한 데이터 통합, 다중 통화 환산, 할당 등 세계적 수준의 기능을 포함하는 사전 패키징된 기능은 즉시 사용 가능한 상태로, 빠르고 비용 효과적으로 배포됩니다.
- 사용자정의 및 확장 가능한 애플리케이션이 산업 표준 도구를 사용하여 빠르고 비용 효과적으로 사용자의 이슈를 해결합니다.
- 웹용으로 디자인되었으므로 사용자가 어디에서나 표준 웹 브라우저를 통해 글로벌 재무 정보에 쉽고 안전하게 액세스할 수 있습니다. 관계형 데이터 저장영역이 사용되므로 사용자가 24시간 연중무휴로 언제든지 업무상 중요한 데이터를 사용할 수 있습니다.

## EPM Workspace

Oracle Hyperion Financial Management는 Oracle Hyperion Enterprise Performance Management Workspace 내에서 사용할 수 있습니다. 탐색, 즐겨찾기, 관리, 툴 메뉴의 환경설정 또는 기능과 같이 EPM Workspace에서 수행되는 태스크에 대한 자세한 내용은 *Oracle Hyperion Enterprise Performance Management Workspace 사용자 가이드* 및 온라인 도움말을 참조하십시오.

## Financial Management 차원

차원은 조직의 데이터를 설명하며, 일반적으로 관련 멤버의 그룹을 포함합니다. 차원의 예로는 계정, 엔티티 및 기간이 있습니다. Oracle Hyperion Financial Management에서는 8개의 시스템 정의 차원을 제공하며 계정에 적용할 수 있는 사용자정의 차원을 개수에 제한 없이 채울 수 있게 합니다.

차원을 구성하는 요소를 멤버라고 합니다. 예를 들어 GrossMargin과 TotalRevenues는 계정 차원의 멤버입니다.

차원 멤버는 계층으로 정렬되어 있습니다. 상위 레벨 멤버를 *상위 멤버*라고 하고, 상위 멤버 바로 아래의 멤버를 해당 *1차 하위*라고 합니다. 상위 아래의 모든 멤버는 *하위 멤버*라고 합니다. 맨아래 레벨 계층 멤버를 *기준 레벨 멤버*라고 합니다.

일부 데이터 차원에서는 계층을 정의할 수 없습니다. 예를 들어 연도 차원은 계층을 지원하지 않습니다.

데이터는 상위 멤버가 아니라 차원의 기본 레벨 멤버에 입력됩니다. 상위 레벨 멤버의 값은 상위 레벨 멤버의 하위 멤버에서 합산됩니다. 기준 레벨 멤버의 데이터가 계산되는 경우도 있습니다.

다음 섹션에서는 시스템 정의 차원에 대해 설명합니다.

## 시나리오 차원

시나리오 차원은 실제, 예산 또는 예측과 같은 데이터 세트를 나타냅니다. 예를 들어 실제 시나리오에는 일반 원장의 데이터가 포함될 수 있으며 과거 및 현재 비즈니스 작업을 반영합니다. 예산 시나리오에는 대상 비즈니스 작업을 반영하는 데이터가 포함될 수 있습니다. 예측 시나리오에는 일반적으로 이후 기간에 대한 예측에 해당하는 데이터가 포함됩니다. Legal 시나리오에는 적합한 GAAP 형식과 규칙에 따라 계산된 데이터가 포함될 수 있습니다.

애플리케이션에 대해 개수에 제한없이 시나리오를 정의할 수 있으며 기본 빈도, 기본 뷰, 0 데이터 설정 등의 시나리오 차원 멤버 속성을 정의할 수 있습니다.

## 연도 차원

연도 차원은 데이터의 회계 연도 또는 달력 연도를 나타냅니다. 애플리케이션에 1년 이상의 데이터가 포함될 수 있습니다. 애플리케이션을 생성할 때 연도 범위를 지정하고 연도 차원에서 연도를 선택하여 데이터를 처리합니다.

## 기간 차원

기간 차원은 분기, 월 등의 기간을 나타냅니다. 기간 차원에는 기간과 빈도가 포함되며 기간이 계층으로 표시됩니다. 예를 들어 실제 시나리오에서 월별로 데이터를 유지 관리하는 경우 일반적으로 이 시나리오와 관련해서 1년에 12개 기간의 데이터를 사용할 수 있습니다. Oracle Hyperion Financial Management에서는 기간 차원에 대해 연도, 월, 주를 지원합니다.

## 엔티티 차원

엔티티 차원은 관리 및 적합한 보고 구조와 같은 회사의 조직 구조를 나타냅니다. 엔티티는 사업부, 자회사, 공장, 지역, 국가, 법인, 비즈니스 단위, 부서 또는 모든 조직 구성 단위를 나타낼 수 있습니다. 개수에 제한없이 엔티티를 정의할 수 있습니다.

엔티티 차원은 시스템의 통합 차원입니다. 엔티티 차원의 계층 구조는 데이터의 다양한 통합 뷰를 반영합니다. 다양한 계층 구조는 지역 통합, 합법적 통합 또는 작업별 통합에 해당할 수 있습니다. 조직에 있는 개별 멤버 구성요소 간의 모든 관계는 이 차원에 저장되고 유지 관리됩니다. 조직의 엔티티는 기본 엔티티, 종속 엔티티 또는 상위 엔티티로 분류될 수 있습니다. 기본 엔티티는 조직 구조의 맨아래에 있으며 다른 엔티티를 소유하지 않습니다. 종속 엔티티는 조직의 다른 엔티티가 소유합니다. 상위 엔티티는 직접 보고하는 하위 멤버를 하나 이상 포함합니다.

기본 통화 및 보안 클래스와 같은 엔티티 차원 멤버의 속성과 엔티티에서 조정을 허용하고 내부거래 세부정보를 저장할지 여부를 지정하는 속성을 정의합니다.

## 값 차원

값 차원은 애플리케이션에 저장된 값 유형을 나타내며 입력 통화, 상위 통화, 조정, 통합 세부정보 (비례, 제거, 기여금액 세부정보) 등을 포함할 수 있습니다. 예를 들어 엔티티 통화 멤버는 엔티티 값을 로컬 통화로 저장합니다. 상위 통화 멤버는 엔티티 값을 상위 엔티티 통화로 환산하여 저장합니다. 값 차원은 데이터에 적용된 트랜잭션의 감사 정보를 제공하는 데 유용합니다.

## 계정 차원

계정 차원은 기본 계정의 계층을 나타냅니다. 계정은 엔티티 및 시나리오에 대한 재무 데이터를 애플리케이션에 저장합니다. 각 계정에는 계산 동작을 정의하는 수익, 비용 등의 유형이 있습니다.

계정 차원 멤버에 대해 계정 유형, 표시할 소수 자릿수, 계정이 계산된 계정, 통합 계정 또는 내부거래 파트너 계정인지 여부 등의 속성을 정의합니다.

## 내부거래 차원

내부거래 차원은 계정에 대해 존재하는 모든 내부거래 잔액을 나타냅니다. 이 차원은 계정 차원 및 임의의 사용자정의 차원과 함께 사용되는 예약된 차원입니다. Oracle Hyperion Financial Management에서는 계정 및 엔티티에서 내부거래 트랜잭션 세부정보를 추적 및 제거할 수 있습니다. 내부거래 매칭 보고서를 실행하여 내부거래 트랜잭션을 볼 수도 있습니다.

## 뷰 차원

뷰 차원은 주기적, 연 누계, 분기 누계 빈도 등 달력 인텔리전스의 다양한 모드를 나타냅니다. 뷰를 [주기적]으로 설정하면 월별 값이 표시됩니다. 뷰를 [연 누계] 또는 [분기 누계]로 설정하면 해당 연도나 분기의 누적값이 표시됩니다.

## 사용자정의 차원

사용자정의 차원은 계정과 연결된 차원입니다. 이러한 차원을 통해 제품, 시장, 채널, 대차대조표 이동, 제거 유형 등 계정과 연결된 추가 세부정보를 지정할 수 있습니다. 예를 들어 사용자정의 차원에는 제품 라인, 지역, 채널 또는 고객이 포함될 수 있습니다. Sales 및 COGS 계정과 연결된 제품의 사용자정의 차원을 사용하면 제품별 판매 및 비용 세부정보를 추적할 수 있습니다.

## 사용자 정의 요소

Oracle Hyperion Financial Management의 많은 요소는 사용자 정의 요소입니다. 예를 들어 분개를 생성할 때 레이블과 설명을 지정합니다.

아래에는 사용자 정의 요소, 각 요소의 최소 및 최대 길이, 추가 제한 사항이 나와 있습니다. 표에서 각 요소는 위치한 모듈을 기준으로 그룹화되어 있습니다.

**참고:** 메타데이터에는 ASCII 문자만 사용할 수 있습니다.

**표 1-1 사용자 정의 요소에 대한 요구사항**

요소	최소 길이	최대 길이	제한
애플리케이션 프로파일			
언어	1	20	없음
기간 레이블	1	80	<ul style="list-style-type: none"> <li>영숫자 문자만 사용해야 합니다.</li> <li>변모음과 같은 발음 구별 부호, 기호 또는 공백은 사용할 수 없습니다.</li> </ul>
뷰 레이블	1	10	<ul style="list-style-type: none"> <li>영숫자 문자만 사용해야 합니다.</li> <li>변모음과 같은 발음 구별 부호, 기호 또는 공백은 사용할 수 없습니다.</li> </ul>
뷰 설명	0	40	앰퍼샌드(&)는 사용할 수 없습니다.
기간 설명	0	40	앰퍼샌드(&)는 사용할 수 없습니다.
애플리케이션 생성			
애플리케이션 레이블	1	10	<ul style="list-style-type: none"> <li>영숫자 문자만 사용해야 합니다.</li> <li>숫자로 시작할 수 없습니다.</li> <li>공백, 기호, 변모음과 같은 발음 구별 부호, 독일어 대문자 Sharp S와 같은 특수 문자는 사용할 수 없습니다.</li> </ul> <p><b>참고:</b> 애플리케이션 레이블은 대소문자를 구분하지 않습니다. 예를 들어 App1과 APP1은 동일한 애플리케이션 레이블로 간주됩니다.</p>
애플리케이션 설명	1	255	<ul style="list-style-type: none"> <li>앰퍼샌드(&amp;)는 사용할 수 없습니다.</li> <li>공백이 포함될 수 없습니다.</li> </ul>

표 1-1 (계속) 사용자 정의 요소에 대한 요구사항

요소	최소 길이	최대 길이	제한
멤버 레이블	1	80	<p>고유해야 합니다. 레이블에는 공백을 포함하여 최대 80자까지 사용할 수 있지만, 공백으로 시작할 수는 없습니다.</p> <p>다음과 같은 문자는 사용할 수 없습니다.</p> <ul style="list-style-type: none"> <li>• 앰퍼샌드(&amp;)</li> <li>• 별표(*)</li> <li>• At 기호(@)</li> <li>• 쉼표(,)</li> <li>• 중괄호({})</li> <li>• 큰따옴표 ""</li> <li>• 슬래시(/)</li> <li>• 빼기 기호(-)</li> <li>• 숫자 기호(#)</li> <li>• 마침표(.)</li> <li>• 더하기 기호(+)</li> <li>• 세미콜론(;)</li> </ul> <p><b>참고:</b> 엔티티 이름으로 ALL을 사용할 수 없습니다.</p>
멤버 설명	0	40	<p>앰퍼샌드(&amp;)는 사용할 수 없습니다.</p> <p><b>참고:</b> 설명 모드에서 작업하도록 선택하는 경우 멤버 설명에 마침표(.)를 사용하지 마십시오.</p>
별칭 레이블	0	80	<p>앰퍼샌드(&amp;)는 사용할 수 없습니다.</p>
보안			
보안 클래스	1	80	<p>다음과 같은 문자는 사용할 수 없습니다.</p> <ul style="list-style-type: none"> <li>• 별표(*)</li> <li>• At 기호(@)</li> <li>• 쉼표(,)</li> <li>• 중괄호({ })</li> <li>• 큰따옴표 ""</li> <li>• 빼기 기호(-)</li> <li>• 숫자 기호(#)</li> <li>• 마침표(.)</li> <li>• 더하기 기호(+)</li> <li>• 세미콜론(;)</li> <li>• 슬래시 기호(/)</li> </ul>
분개			



표 1-1 (계속) 사용자 정의 요소에 대한 요구사항

요소	최소 길이	최대 길이	제한
분개 레이블	1	20	다음과 같은 문자는 사용할 수 없습니다. <ul style="list-style-type: none"> <li>• 별표(*)</li> <li>• At 기호(@)</li> <li>• 쉼표(,)</li> <li>• 중괄호({})</li> <li>• 큰따옴표 ""</li> <li>• 슬래시(/)</li> <li>• 빼기 기호(-)</li> <li>• 숫자 기호(#)</li> <li>• 마침표(.)</li> <li>• 더하기 기호(+)</li> <li>• 세미콜론(;)</li> </ul>
분개 설명	0	255	없음
분개 그룹	0	30	없음
분개 라인 항목 설명	0	50	없음
로드/추출			
구분자	1	1	다음 문자 중 하나여야 하며, 파일이나 파일 이름에 사용할 수 없습니다. <ul style="list-style-type: none"> <li>• 앰퍼샌드(&amp;)</li> <li>• At 기호(@)</li> <li>• 백슬래시(\)</li> <li>• 캐럿(^)</li> <li>• 콜론(:)</li> <li>• 쉼표(,)</li> <li>• 달러 기호(\$)</li> <li>• 라인( )</li> <li>• 퍼센트 기호(%)</li> <li>• 물음표(?)</li> <li>• 세미콜론(;)</li> <li>• 물결표(~)</li> </ul> <p><b>참고:</b> 파일 전체에서 동일한 구분자를 사용해야 합니다. 한 파일 내에서 여러 구분자를 사용하면 파일을 로드할 때 오류가 발생합니다.</p>
데이터 그리드			
셀 설명	1	1900	없음
라인 항목 세부정보	1	80	없음
노트	0	255	없음

표 1-1 (계속) 사용자 정의 요소에 대한 요구사항

요소	최소 길이	최대 길이	제한
소수점 문자	1	1	다음 문자는 데이터 그리드에 부적합한 소수점 문자입니다. <ul style="list-style-type: none"> <li>백슬래시(\)</li> <li>슬래시(/)</li> <li>빼기 기호(-)</li> <li>더하기 기호(+)</li> </ul>
문서			
문서 이름 (폴더 및 보고서 이름 포함)	1	16	다음 문자는 문서 이름에 부적합한 문자입니다. <ul style="list-style-type: none"> <li>별표(*)</li> <li>At 기호(@)</li> <li>백슬래시(\)</li> <li>콜론(:)</li> <li>쉼표(,)</li> <li>중괄호({})</li> <li>큰따옴표 ""</li> <li>슬래시(/)</li> <li>보다 큼 기호(&gt;)</li> <li>보다 작음 기호(&lt;)</li> <li>라인( )</li> <li>숫자 기호(#)</li> <li>문서 이름의 끝에 있는 마침표(.)</li> <li>더하기 기호(+)</li> <li>물음표(?)</li> <li>세미콜론(;)</li> </ul> <p>참고: 문서 이름에는 후행 또는 선행 공백도 사용할 수 없습니다.</p>

## 2

# 애플리케이션 관리

애플리케이션은 함께 사용되는 엔티티, 계정, 시나리오 및 기타 차원 세트에 의해 이루어집니다. 필요한 개수만큼 애플리케이션을 생성할 수 있습니다. 예를 들어 여러 조직의 세금 데이터에 대해 보고하도록 한 애플리케이션을 설정하고, 다른 조직의 보안 및 환율 커미션 데이터에 대해 보고하도록 다른 애플리케이션을 설정할 수 있습니다.

애플리케이션 관리에는 차원 편집자 및 Financial Management 애플리케이션 생성자/Financial Management Manager 보안 역할이 필요합니다. 역할에 대한 자세한 내용은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

애플리케이션은 애플리케이션 서버에서 실행됩니다. 여러 서버에서 로드 밸런싱을 수행하도록 애플리케이션 서버 클러스터를 설정할 수 있습니다. 지침은 *Oracle Enterprise Performance Management System 설치 및 구성 가이드*를 참조하십시오.

애플리케이션 서버를 구성할 때 지정하는 관리자 그룹의 멤버만 이러한 관리 태스크를 수행할 수 있습니다.

### 시스템상의 사용자

- 사용자 나열
- 사용자 로그아웃

### 서버 및 애플리케이션 관리

- 사용 안함으로 설정된 구성요소 보기
- 연결 활성화 또는 비활성화
- 사용자 로그아웃

### 시스템 메시지

- 보기 - 관리자 그룹의 멤버여야 함
- 삭제 - 관리자 그룹의 멤버여야 함

관리자 그룹에 사용자를 지정하려면 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

## 애플리케이션 관리

다음 단계를 사용하여 애플리케이션을 정의합니다.

1. 애플리케이션이 있는 애플리케이션 서버, 애플리케이션 레이블 및 설명, 달력, 언어 및 빈도 등록정보가 지정되는 애플리케이션 웹을 생성합니다. [애플리케이션 생성](#)을 참조하십시오.
2. 애플리케이션에 액세스할 수 있는 사용자 및 각 사용자의 액세스 유형을 비롯한 애플리케이션 보안을 정의합니다. [애플리케이션 보안 관리](#)를 참조하십시오.
3. 계정, 엔티티, 시나리오 및 사용자정의 차원과 애플리케이션 설정, 통합 메소드 및 통화를 비롯한 애플리케이션 메타데이터를 정의합니다. [메타데이터 관리](#)를 참조하십시오.
4. 데이터, 데이터 양식, 멤버 목록, 규칙 및 분개를 애플리케이션에 로드합니다.

## 애플리케이션 생성

애플리케이션은 일련의 분석 또는 보고 요구사항을 충족하는 차원 및 차원 멤버 세트입니다. 예를 들어 세금 데이터를 보고하는 애플리케이션과 판매 분석에 사용되는 별도의 애플리케이션이 있을 수 있습니다.

Oracle Hyperion Tax Provision 애플리케이션에 대한 자세한 내용은 Tax Provision 설명서를 참조하십시오.

애플리케이션을 생성하는 방법은 여러 가지가 있습니다.

- 새 애플리케이션을 생성합니다. 이 옵션은 [애플리케이션 생성] 마법사를 실행합니다. 애플리케이션 달력, 모듈 구성 및 사용자정의 차원을 정의할 수 있습니다. [새 애플리케이션 생성](#)을 참조하십시오.
- 애플리케이션에서 복사합니다. 애플리케이션의 애플리케이션 달력, 모듈 구성 및 사용자정의 차원을 보고 [애플리케이션 생성] 마법사를 통해 수정하여 새 애플리케이션을 생성합니다. [애플리케이션에서 복사](#)를 참조하십시오.
- 프로파일 파일에서 애플리케이션을 생성합니다. 달력을 보고 수정할 수도 있습니다. [프로파일 파일에서 애플리케이션 생성](#)을 참조하십시오.

## 새 애플리케이션 생성

새 애플리케이션을 생성하려면 [애플리케이션 관리자] 보안 역할이 있어야 합니다.

애플리케이션을 생성하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **애플리케이션** 탭에서 **작업, 새로 작성** 순으로 선택합니다.  
시스템에서 [애플리케이션 생성] 마법사를 실행합니다.
3. **애플리케이션 옵션**에서 **새 애플리케이션**을 선택하고 다음을 누릅니다.
4. **애플리케이션 등록정보** 섹션에서 **이름**에 새 애플리케이션의 이름을 입력합니다.

애플리케이션 레이블에는 최대 10자의 영숫자 문자나 12바이트를 사용할 수 있지만 숫자로 시작할 수 없으며 공백, 앰퍼샌드(&) 기호, 특수 문자 또는 5자를 초과하는 국제 문자를 포함할 수 없습니다. 또한 독일어 대문자 ¶(유니코드 U+1E9E) 및 소문자 ß(유니코드 U+00DF) 또는 터키어 점 없는 소문자 ı(유니코드 U+0131)를 사용할 수 없습니다.

애플리케이션 레이블은 대/소문자를 구분하지 않습니다. 예를 들어 App1과 APP1은 동일한 애플리케이션으로 간주됩니다. HFM, HSV 및 HSX는 예약된 이름이며 애플리케이션 레이블이나 설명에 사용할 수 없습니다.

### ▲ 주의:

다른 Oracle Hyperion Financial Management 애플리케이션 서버에 있더라도 동일한 이름으로 애플리케이션을 생성하지 마십시오. 서로 다른 Financial Management 애플리케이션 서버에 있지만 이름이 같은 애플리케이션은 하나의 Oracle Hyperion Shared Services 서버에서 함께 사용할 수 없습니다.

5. 설명에 애플리케이션에 대한 설명을 입력합니다.  
애플리케이션 설명에는 최대 255자를 사용할 수 있으며 공백을 포함할 수 있습니다. 앰퍼샌드 (&)는 사용할 수 없습니다. HFM, HSV 및 HSX는 예약된 이름이며 애플리케이션 설명에 사용할 수 없습니다.
6. 애플리케이션 유형 목록에서 유형을 선택합니다.
  - 통합
  - 세금 규정
7. 클러스터 목록에서 새 애플리케이션을 실행할 애플리케이션 서버 클러스터를 선택합니다.  
서버 클러스터가 표시되지 않는 경우 서버 클러스터를 등록해야 할 수도 있습니다. *Oracle Enterprise Performance Management System 설치 및 구성 가이드*를 참조하십시오.
8. 사용자 관리 프로젝트 목록에서 애플리케이션을 추가할 Shared Services 프로젝트를 선택합니다.



주:

각 애플리케이션은 프로젝트에 속해 있어야 합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

9. 언어에서 애플리케이션 전체에 사용되는 설명에 대해 최대 10개의 언어를 지정할 수 있습니다. 언어 이름은 심표로 구분합니다. 각 언어 이름에는 최대 20자를 사용할 수 있습니다. 공백은 한 문자로 계산됩니다.



주의:

애플리케이션을 생성한 후에는 애플리케이션의 언어를 변경할 수 없습니다.

10. 연도, 기간 및 뷰 차원을 사용하여 달력을 정의합니다. 연도 차원의 멤버는 애플리케이션의 시작 연도와 총 연도 수를 정의합니다. 기간 차원의 멤버 계층은 기준 레벨 및 요약 레벨 기간을 정의합니다. 뷰 차원은 기간누계 뷰(예: 연도(YTD), 반기(HYTD), 분기(QTD) 및 월(MTD))를 정의합니다.



주의:

애플리케이션을 생성한 후에는 애플리케이션의 시작 연도, 기간 설명 또는 레이블을 변경할 수 없습니다.

- 연도 섹션의 시작 연도에 달력 시작 연도를 입력하십시오.
- 연도 수에는 애플리케이션에 포함할 총 연도 수를 입력합니다.

 주:

애플리케이션의 연도 수는 기존 애플리케이션에 대해 수정할 수 있는 유일한 프로파일 정의입니다. **애플리케이션 수정**을 참조하십시오.

## 11. 기간 섹션에서 포함할 기간을 선택합니다.

## • 월

포함할 기간(반기, 분기 또는 삼분기)을 선택합니다.

시작 월 드롭다운 목록에서 달력의 첫번째 월을 선택합니다.

## • 분기

**선택사항:** 반기를 포함하려면 선택합니다.

## • 사용자정의. 사용자정의 달력을 정의하는 경우 기간 레이블 접두어 및 기준 기간 수를 정의해야 합니다. 플랫폼 목록이 생성되며 나중에 계층을 수정할 수 있습니다.

– 기간 레이블 접두어에 포함할 기간의 접두어를 입력합니다.

레이블에는 공백을 포함하여 최대 10자를 사용할 수 있습니다.

– 기준 기간 수에 연도의 기간 수를 입력합니다.

기간 수로 10 미만의 수를 입력하고 레이블 접두어로 NewPeriod를 입력하면 NewPeriod1 등의 기간이 계층에 추가됩니다. 기간 수가 10을 초과하지만 100 미만인 경우 기간은 NewPeriod01로 시작합니다. 기간 수가 100을 초과하지만 1000 미만인 경우 기간은 001로 시작하는 것과 같은 방식이 사용됩니다.

 주:

수동 달력을 정의하는 경우 기간 레이블 접두어 및 기준 기간 수를 비워 둡니다. 그런 다음 필요한 빈도와 기간을 입력할 수 있습니다.

## 12. 다음을 누릅니다.

## 빈도 정의

빈도는 데이터를 입력할 수 있는 기간 레벨(예: 월 또는 분기)을 지정합니다. 빈도 및 해당 뷰(YTD, HYTD, QTD, MTD)는 달력을 정의할 때 선택한 기간을 기준으로 생성됩니다. 빈도를 추가, 수정 및 삭제할 수 있습니다. 이전에 정의한 각 언어로 각 빈도와 뷰를 설명하는 레이블을 입력할 수도 있습니다.

기본적으로 [빈도] 열의 첫번째 행에는 YTD(연 누계)가 제공됩니다. YTD 레이블은 삭제하거나 변경할 수 없습니다. 하지만 정의한 각 언어로 YTD 설명을 입력할 수 있습니다.

 주:

정의할 수 있는 빈도 외에도 각 애플리케이션에는 두 개의 시스템 정의 빈도와 해당 뷰인 [시나리오 뷰]와 [주기적]이 포함되어 있습니다.

달력을 수동으로 정의하도록 선택한 경우 [빈도] 그리드가 비어 있으므로 필요한 뷰와 해당 설명을 입력해야 합니다. 기간 차원의 각 레벨에 대해 하나의 빈도를 입력해야 합니다.

### ▲ 주의:

애플리케이션을 생성한 후에는 애플리케이션의 빈도 설명이나 레이블을 변경할 수 없습니다.

빈도를 정의하려면 다음을 수행합니다.

1. 빈도 화면에서 (+) 빈도 추가 버튼을 누르거나 작업, 빈도 추가 순으로 선택합니다.

2. 하나 이상의 빈도 및 설명을 입력합니다.

기본적으로 첫 번째 빈도는 YTD입니다. 정의한 빈도 수는 기간 트리의 세대 수와 같아야 합니다.

### ✎ 주:

레이블에는 최대 40자를 사용할 수 있습니다. 설명에는 최대 80자를 사용할 수 있습니다.

+ - \* / # { } ; , @ "" 문자는 레이블에 사용할 수 없습니다.

### 💡 팁:

애플리케이션을 생성한 후에는 빈도를 수정할 수 없으므로 각 언어로 작성된 각 빈도에 대한 설명을 포함해야 합니다.

3. 다음을 누릅니다.

## 기간 편집

기간 계층은 정의한 기간과 빈도의 조합입니다. 기간을 추가하거나 삭제하여 이 계층을 변경할 수 있습니다. 예를 들어 4사분기에 다른 달을 추가하여 13개월 연도를 표시할 수 있습니다.

### ✎ 주:

기준 기간 수는 0보다 커야 합니다.

기간을 선택하면 기간 레이블이나 설명을 변경할 수 있습니다. [연도] 기간에 대한 레이블이나 설명은 편집할 수 없습니다.

 주:

언어를 기준으로 기간 레이블을 편집할 수는 없습니다. 한 언어로 기간 레이블을 변경하면 변경사항이 애플리케이션 프로파일에 정의된 다른 모든 언어로 전달됩니다. 하지만 각 언어에 대해 고유한 기간 설명을 사용할 수 있습니다.

계층에 동위 및 1차 하위 기간을 추가하려면 [동위 멤버 및 1차 하위 구성요소 기간 추가](#)를 참조하십시오.



기간을 입력하고 편집하려면 다음을 수행합니다.

1. 기간 계층에서 기간을 선택하고 기간 레이블 또는 설명을 입력하거나 편집합니다.  
기간 레이블에는 최대 40자를 사용할 수 있습니다. 기간 설명에는 최대 80자를 사용할 수 있습니다. 공백은 한 문자로 계산됩니다.
2. 각 기간에 대한 설명이 포함되었는지 확인합니다.  
설명을 포함하지 않고 나중에 추가하는 경우 애플리케이션 프로파일을 수정하고 해당 프로파일을 사용하는 모든 애플리케이션을 다시 생성해야 합니다.


## 동위 멤버 및 1차 하위 구성요소 기간 추가

기간 계층에 동위 기간과 하위 기간을 하나 또는 여러 개 추가할 수 있습니다.

하위 기간 또는 동위 기간을 하나 추가하려면 다음을 수행합니다.

1. 1차 하위 구성요소 또는 동위 멤버 기간을 하나 추가할 기간을 강조 표시합니다.
2. 옵션을 선택합니다.
  - **1차 하위 삽입 버튼**()을 누르거나 **작업, 1차 하위 삽입** 순으로 선택합니다.
  - **동위 삽입 버튼**()을 누르거나 **작업, 동위 삽입** 순으로 선택합니다.
3. 새 동위 멤버 또는 1차 하위 구성요소 기간의 이름을 입력합니다.

1차 하위 구성요소 또는 동위 멤버 기간을 여러 개 추가하려면 다음을 수행합니다.

1. 1차 하위 구성요소 또는 동위 멤버 기간을 여러 개 추가할 기간을 강조 표시합니다.
2. **여러 항목 삽입 버튼**()을 누르거나 **작업, 여러 항목 삽입** 순으로 선택합니다.
3. 계층에 추가할 기간 수를 입력합니다.
4. 새 기간의 레이블 접두어를 입력하고 **확인**을 누릅니다.

기간 수로 10 미만의 수를 입력하고 레이블 접두어로 NewPeriod를 입력하면 NewPeriod1 등의 기간이 계층에 추가됩니다. 기간 수가 10을 초과하지만 100 미만인 경우 기간은 NewPeriod01로 시작합니다. 기간 수가 100을 초과하지만 1000 미만인 경우 기간은 001로 시작하는 것과 같은 방식이 사용됩니다.

5. 기간 추가가 완료되면 **다음**을 누릅니다.

## 기간 삭제

계층에서 기간을 삭제하면 해당 기간의 모든 하위 멤버도 삭제됩니다.



 주:

[연도] 기간은 삭제할 수 없습니다.

기간을 삭제하려면 다음을 수행합니다.

1. 삭제할 기간을 선택합니다.
2. 기간 삭제를 누르거나 **작업, 기간 삭제** 순으로 선택합니다.

다음을 누릅니다.

## 기능 정의

[기능] 화면에는 사용으로 설정된 애플리케이션 모듈을 기반으로 애플리케이션 설정 및 기본값이 나열됩니다. 기본값은 변경할 수 있습니다.

## 애플리케이션 설정 지정

애플리케이션 설정에 대한 자세한 내용은 [애플리케이션 설정 정의](#)를 참조하십시오.

다음 설정을 지정할 수 있습니다.

- **애플리케이션 통화** - 애플리케이션에서 사용할 통화입니다. 통화를 지정하면 해당 통화가 통화 차원에 추가됩니다. [통화 정의](#)를 참조하십시오.
- **대차일치 계정의 환율 계정** - BALANCE 계정의 환율 계정입니다. 계정을 지정하면 해당 계정이 계정 차원에 추가됩니다. [계정 정의](#)를 참조하십시오.
- **플로우 계정의 환율 계정** - FLOW 계정의 환율 계정입니다. 계정을 지정하면 해당 계정이 계정 차원에 추가됩니다. [계정 정의](#)를 참조하십시오.
- **대차일치 계정의 환산 방법** - BALANCE 계정의 환산 방법입니다. [계정 정의](#)를 참조하십시오.
- **플로우 계정의 환산 방법** - FLOW 계정의 환산 방법입니다. [계정 정의](#)를 참조하십시오.

애플리케이션 설정을 지정하려면 다음을 수행합니다.

1. 기능 화면에서 다음 애플리케이션 설정 값을 지정하거나 기본값을 사용합니다.
  - **애플리케이션 통화** - 애플리케이션 통화를 입력합니다. 모든 통화가 나열되며 선택할 수 있습니다. 예를 들어 US를 입력하면 USD - US Dollar 통화가 표시되고 선택할 수 있습니다.
  - **대차일치 계정의 환율 계정** - BALANCE 계정에 사용할 환율 계정을 입력합니다. 계정 이름에는 공백을 포함하여 최대 80자까지 사용할 수 있지만, 공백으로 시작할 수는 없습니다. [계정 정의](#)를 참조하십시오.
  - **플로우 계정의 환율 계정** - FLOW 계정에 사용할 환율 계정을 입력합니다. 계정 이름에는 공백을 포함하여 최대 80자까지 사용할 수 있지만, 공백으로 시작할 수는 없습니다. [계정 정의](#)를 참조하십시오.
  - **대차일치 계정의 환산 방법** - 드롭다운 목록에서 다음과 같이 BALANCE 계정의 환산 방법을 선택합니다.
    - PVA - 주기적 값 환산 방법에 사용
    - VAL - 환율 환산 방법의 값 사용

- **플로우 계정의 환산 방법** - 드롭다운 목록에서 다음과 같이 FLOW 계정의 환산 방법을 선택합니다.
    - PVA - 주기적 값 환산 방법에 사용
    - VAL - 환율 환산 방법의 값 사용
2. 애플리케이션 모듈을 사용 또는 사용 안함으로 설정하려면 **애플리케이션 모듈 사용 또는 사용 안함**을 참조하십시오.

## 애플리케이션 모듈 사용 또는 사용 안함

애플리케이션을 생성하면 기본적으로 모든 애플리케이션 모듈이 활성화됩니다. 사용자는 적절한 보안 역할이 있는 모듈을 볼 수 있습니다.

애플리케이션 유형에 따라 일부 모듈이 필요하지 않을 수도 있습니다. 애플리케이션 관리자인 경우 특정 애플리케이션 모듈을 비활성화하도록 선택할 수 있습니다. 예를 들어 특정 애플리케이션에 대해 분개 또는 내부거래 트랜잭션을 비활성화할 수 있습니다. 모듈을 비활성화하면 모든 애플리케이션 사용자에게 표시되지 않습니다.

### 주:

비활성화할 모듈과 연관된 문서(예: 분개 또는 내부거래 트랜잭션 보고서)가 있는 경우 해당 문서도 시스템에서 제거해야 합니다.

모듈을 변경한 후 변경사항을 보려면 애플리케이션을 닫았다가 다시 엽니다.

애플리케이션 모듈 구성 정보는 `application name_Moduleconfig.xml`이라는 XML 파일로 저장됩니다. [애플리케이션 요소 로드] 및 [애플리케이션 요소 추출] 페이지에서 모듈 구성 정보를 로드하고 추출할 수 있습니다.

애플리케이션 모듈을 사용 또는 사용 안함으로 설정하려면 다음을 수행합니다.

1. 기능 화면에서 **예**를 선택하여 애플리케이션 모듈을 사용으로 설정하거나 **아니오**를 선택하여 사용 안함으로 설정하거나 다음 기본값을 사용합니다.
  - **프로세스 제어 사용**

### 주:

프로세스 제어를 사용하지 않는 경우 제출 단계를 사용할 수 없습니다.

- 제출 단계 사용
  - 소유권 관리 사용
  - 분개 사용
  - 데이터 관리 사용
  - 태스크 감사 사용
  - 내부거래 트랜잭션 사용
  - 지분 매입 사용
2. 다음을 누릅니다.

## 사용자정의 차원 생성

차원 화면을 사용하여 차원 및 차원 보안 등록정보를 구성합니다. 사용자정의 차원을 추가 또는 삭제하고 사용자정의 차원 이름 및 별칭을 지정할 수 있습니다.

차원 테이블에서는 애플리케이션에 대한 처음 두 개의 사용자정의 차원이 자동으로 표시됩니다. 이러한 사용자정의 차원은 환율 및 통합 메소드 정보에 사용되며 차원의 크기가 [크게]로 설정되어야 합니다. 첫 번째 차원은 "시작 통화" 및 통합 메소드 정보에 사용되고 두 번째 차원은 "대상 통화" 정보에 사용됩니다. 다음 두 항목에 대한 사용자정의 차원의 차원 이름(약식 레이블) 및 차원 별칭(긴 레이블)을 지정할 수 있습니다. 크기를 변경할 수 없으며 이러한 차원은 삭제할 수 없습니다.

기본 차원 외에도 다음 지침에 따라 사용자정의 차원을 추가로 생성할 수 있습니다.

- 애플리케이션에 고유한 사용자정의 차원 이름 및 차원 별칭을 입력해야 합니다. 이름과 별칭은 모두 고유해야 합니다. 예를 들어 차원 이름이 PROD인 경우 차원 별칭을 PROD로 지정할 수 없습니다. 사용자정의 차원 이름은 통화 이름과 동일할 수 없습니다.
- 차원 이름에는 최대 10자를 사용할 수 있습니다.
- 차원 별칭에는 최대 20자를 사용할 수 있습니다.
- 차원 이름과 별칭에 공백을 사용할 수 없으며 비워 둘 수도 없습니다.
- 사용자정의 차원의 크기로 [작게], [보통] 또는 [크게]를 선택해야 합니다.

최대 사용자정의 차원 수는 각 사용자정의 차원의 크기와 데이터베이스 유형에 따라 달라집니다.

- 큰 사용자정의 차원은 20억 개 멤버를 지원하며 데이터베이스 열에 4바이트의 저장소 공간이 필요합니다.
- 중간 차원은 32,000개 멤버를 지원하며 2바이트가 필요합니다.
- 작은 차원은 최대 128을 지원하며 1바이트가 필요합니다.

최대 사용자정의 차원 수의 계산 방법은 데이터베이스 유형에 관계없이 동일하지만 데이터베이스 유형에 따라 사용 가능한 총 공간 크기가 결정됩니다.

총 물리적 사용자정의 열 수는 Oracle, Microsoft SQL Server 또는 IBM DB2 데이터베이스에서 사용자정의 차원 저장소에 사용할 수 있는 총 바이트 수를 제어합니다.

- Oracle은 최대 21개의 물리적 사용자정의 열을 지원하며, 사용 가능한 총 공간은  $21 \times 8 = 168$ 바이트입니다.
- SQL Server는 5개의 물리적 사용자정의 열을 지원하며, 사용 가능한 총 공간은  $5 \times 8$ 바이트 = 40바이트입니다.
- IBM DB2는 최대 900바이트의 기본 키를 지원하며, 사용 가능한 총 공간은 800바이트입니다.

두 개 이상의 사용자정의 차원이 필요합니다. 다음은 세 개 이상의 사용자정의 차원을 계산하기 위한 공식입니다.

Migrating applications:  $4 \times LD + 2 \times MD + SD \leq \text{MAXBYTES} - 8$

New applications:  $4 \times LD + 2 \times MD + SD \leq \text{MAXBYTES}$

$LD$  = [크게] 차원 수,  $MD$  = [보통] 차원 수 및  $SD$  = [작게] 차원 수입니다.

$\text{MAXBYTES}$  = 40(Microsoft SQL Server), 168(Oracle), 800(IBM DB2)입니다.

모든 실질적인 용도의 사용자정의 차원에는 [보통]을 사용하는 것이 좋습니다.

애플리케이션 수명 중에 멤버 수가 128개를 초과할 수 있는 경우 사용자정의 차원에 [작게] 크기를 사용하지 않는 것이 좋습니다.

애플리케이션 수명 중에 해당 차원의 멤버 수가 32,536개를 초과할 수 있는 경우에만 사용자정의 차원에 [크게] 크기를 사용하는 것이 좋습니다.

다음 예에서는 샘플 사용자정의 차원을 보여 줍니다.

사용자정의 차원 이름	사용자정의 차원 별칭	사용자정의 차원 크기	통화에 사용
Prod	Products	크게	From 통화
Mkt	Markets	크게	To 통화
Flows	BSFlows	작게	N/A
Cust	Customers	크게	N/A

사용자정의 차원을 생성하려면 다음을 수행합니다.

1. 차원 테이블에서 (+) 사용자정의 차원 생성 버튼을 누르거나 작업, 사용자정의 차원 생성 순으로 선택합니다.
2. 약식 이름에 고유한 차원 이름을 입력합니다.
3. 별칭 이름에 고유한 차원 별칭을 입력합니다.
4. 차원 크기에 대해 작게, 보통 또는 크게 옵션 중 하나를 선택합니다.
5. 추가 차원을 생성하려면 이러한 단계를 반복합니다.
6. **선택사항:** 애플리케이션 생성 단계를 모두 완료한 경우 **애플리케이션 생성**을 누릅니다.

애플리케이션이 성공적으로 생성되면 성공적으로 애플리케이션이 생성되었고 메타데이터가 로드되었다는 확인 메시지가 표시됩니다. 확인 메시지에서 **확인**을 누릅니다.

#### 주:

애플리케이션을 생성한 후에는 차원 수를 수정할 수 없습니다. 추가 사용자정의 차원이 필요한 경우 추가 사용자정의 차원으로 새 애플리케이션을 생성해야 합니다.

## 애플리케이션 프로파일 저장

애플리케이션 생성 마법사의 모든 단계가 완료되고 애플리케이션 생성을 누르면 시스템은 마법사 탐색 중 사용자가 선택한 옵션을 사용하여 이진 애플리케이션 프로파일(.PER) 파일을 생성합니다. 그런 다음 이 프로파일 파일을 통해 애플리케이션이 생성됩니다. 메타데이터가 업데이트되어 선택한 설정으로 로드됩니다.

애플리케이션에 대한 프로파일 파일이 생성되면 해당 프로파일 파일을 사용하여 새 애플리케이션을 생성할 수 있습니다. **프로파일 파일에서 애플리케이션 생성**을 참조하십시오.

## 프로파일 파일에서 애플리케이션 생성

새 애플리케이션을 생성하는 경우 시스템은 마법사 탐색 중 선택한 옵션을 사용하여 이전 애플리케이션 프로파일(.PER) 파일을 생성합니다. 그런 다음 이 프로파일 파일을 통해 애플리케이션이 생성됩니다. 기존 애플리케이션의 프로파일 파일을 사용하여 새 애플리케이션을 생성할 수 있습니다.

프로파일 파일을 사용하는 경우 시스템은 애플리케이션 설정의 기본값을 사용합니다.

프로파일 파일에서 애플리케이션을 생성하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **애플리케이션** 탭에서 **작업, 새로 작성** 순으로 선택합니다.  
시스템에서 [애플리케이션 생성] 마법사를 실행합니다.
3. **애플리케이션 옵션**에서 **프로파일 파일에서 애플리케이션**을 선택하고 다음을 누릅니다.
4. **프로파일 파일 찾아보기** 화면에서 **찾아보기**를 누르고, 애플리케이션 프로파일(.PER) 파일을 선택하고, 다음을 누릅니다.
5. 필요한 경우 애플리케이션 등록정보를 수정하고 **애플리케이션 생성**을 누릅니다.

## 애플리케이션에서 복사

애플리케이션을 생성하기 위해 기존 애플리케이션에서 애플리케이션 프로파일을 복사할 수 있습니다. 애플리케이션의 애플리케이션 달력, 모듈 구성 및 사용자정의 차원을 보고 [애플리케이션 생성] 마법사를 통해 수정하여 새 애플리케이션을 생성할 수 있습니다.

애플리케이션을 생성하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **애플리케이션** 탭에서 **작업, 새로 작성** 순으로 선택합니다.  
시스템에서 [애플리케이션 생성] 마법사를 실행합니다.
3. **애플리케이션 옵션**에서 **애플리케이션에서 복사**를 선택하고 다음을 누릅니다.
4. **애플리케이션 선택** 화면에서 복사할 애플리케이션을 선택합니다.  
애플리케이션 설정 및 모듈 구성을 비롯하여 선택한 애플리케이션의 정보로 마법사가 채워집니다. 현재 값을 변경할 수 있습니다.
  - [애플리케이션 등록정보] 화면에서 애플리케이션을 생성하려면 **애플리케이션 생성**을 누릅니다.
  - 애플리케이션 정보를 수정하려면 다음을 눌러 마법사에서 탐색합니다.

## 애플리케이션 수정

애플리케이션을 생성한 후에는 일반적으로 애플리케이션 프로파일에 대한 수정이 허용되지 않습니다. 그러나 관리자는 다음 설정을 변경할 수 있습니다.

- 애플리케이션 모듈을 사용 또는 사용 안함으로 설정합니다. **애플리케이션 모듈 사용 또는 사용 안함**을 참조하십시오.
- 애플리케이션에 사용된 연도 수를 늘립니다.

애플리케이션이 지원하는 연도 수를 수정할 수 있으나 다음과 같은 제한이 있습니다.

- 새 프로파일 종료 연도가 현재 종료 연도를 초과해야 합니다. 연도 수를 늘릴 수만 있습니다. 연도 수를 줄이는 것은 지원되지 않습니다.
- 새 종료 연도는 2100 미만이어야 합니다.
- 애플리케이션 프로파일 시작 연도는 수정할 수 없습니다.

#### 예

Current start year: 2005

Current number of years: 10 (End year 2015)

Request for modification: 15 (End year 2020)

애플리케이션 프로파일 종료 연도를 처리하려면 먼저 시스템에서 애플리케이션을 관리 모드로 설정해야 합니다. 현재 애플리케이션 프로세스가 중지되고 애플리케이션에 로그인한 모든 사용자가 강제로 로그아웃됩니다.

애플리케이션을 수정하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **애플리케이션** 탭에서 수정할 애플리케이션을 선택합니다.
3. **애플리케이션 수정**을 누르거나, **작업, 애플리케이션 수정** 순으로 선택하거나, 애플리케이션을 마우스 오른쪽 버튼으로 누르고 **애플리케이션 수정**을 선택합니다.  
대부분의 애플리케이션 등록정보는 회색으로 표시됩니다.
4. **연도 수**를 변경하려면 해당 애플리케이션의 연도 수를 입력합니다.
5. **애플리케이션 수정**을 누릅니다.

수정되는 애플리케이션을 처리하기 전에 시스템에서 경고 메시지가 표시됩니다. [예]를 누르면 애플리케이션이 [관리 모드]로 설정되고 연도 수가 검증됩니다. 검증 오류가 없으면 연도 차원을 업데이트하고 [관리 모드]를 사용 안함으로 설정합니다.

## 애플리케이션 등록

설치하는 동안 서버 클러스터를 구성하고 등록합니다. 애플리케이션을 만든 후 기본 서버 클러스터에 애플리케이션을 등록할 수 있습니다.

애플리케이션을 등록하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리, 애플리케이션** 순으로 선택합니다.
3. 애플리케이션 목록에서 애플리케이션을 선택합니다.
4. 등록을 누르거나 **작업, 등록** 순으로 선택합니다.
5. 클러스터 목록에서 서버 클러스터를 선택합니다.
6. **사용자 관리 프로젝트** 목록에서 Oracle Hyperion Shared Services 프로젝트를 선택합니다.

7. 등록을 누릅니다.

## 애플리케이션 열기

Oracle Hyperion Financial Management에서는 모든 데이터가 애플리케이션 내에서 처리됩니다. 한 번에 여러 애플리케이션을 열고 작업할 수 있습니다.

애플리케이션에 액세스하려면 애플리케이션의 사용자로 지정되어야 합니다.

애플리케이션을 열려면 다음을 수행합니다.

1. **탐색, 애플리케이션, 통합** 순으로 선택합니다.
2. 애플리케이션을 선택합니다.



주:

애플리케이션이 표시되지 않으면 **새로고침**을 누릅니다.

## 애플리케이션 닫기

작업 중인 현재 애플리케이션을 닫거나, 여러 애플리케이션이 열려 있는 경우 동시에 모두 닫을 수 있습니다.

애플리케이션을 닫으려면 다음 작업 중 하나를 수행합니다.

- **파일, 닫기, 현재** 또는 **모두** 순으로 선택합니다.
- 애플리케이션이 열려 있는 탭의 맨위에 있는 X를 누릅니다.

## 애플리케이션 서버 클러스터 변경

관리 모듈을 사용하여 애플리케이션을 관리합니다. 기본적으로 관리 모듈은 첫 번째 사용 가능한 클러스터/서버를 사용합니다. 이 옵션을 사용하여 연결된 서버를 변경할 수 있으며, 관리 모듈이 재로드됩니다.

서버 클러스터를 변경하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리, 클러스터 변경** 순으로 선택합니다.
3. [클러스터 서버] 목록에서 클러스터를 선택합니다.
4. **확인**을 누릅니다.

## 애플리케이션 목록 보기

[애플리케이션] 테이블에는 시스템에서 사용가능한 모든 Oracle Hyperion Financial Management 애플리케이션 목록이 포함됩니다. 이 테이블에는 애플리케이션 이름과 설명이 표시됩니다.

애플리케이션 목록을 보려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.

2. 관리, 애플리케이션 순으로 선택합니다.

## 애플리케이션 삭제

애플리케이션을 삭제하기 전에 현재 애플리케이션을 사용 중인 다른 사용자가 없는지 확인합니다.

애플리케이션을 삭제하려면 다음과 같은 보안 역할이 있어야 합니다.

- Shared Services: Financial Management 관리자( 또는Manager) Shared Services: Financial Management 관리자(Administrator) 및 Shared Services: Financial Management 애플리케이션 생성자
- 애플리케이션: 애플리케이션 관리자
- 애플리케이션: 프로비저닝 관리자

애플리케이션을 삭제하려면 다음을 수행합니다.

1. 탐색, 관리, 통합 관리 순으로 선택합니다.
2. 관리, 애플리케이션 순으로 선택합니다.
3. 삭제할 애플리케이션을 선택합니다.
4. 삭제를 누르거나 작업, 삭제 순으로 선택합니다.
5. 애플리케이션이 열려 있으면 경고 메시지가 표시됩니다. 옵션을 선택합니다.
  - 애플리케이션을 강제로 삭제하려면 **강제 삭제**를 누릅니다.  
로그인한 사용자는 애플리케이션이 삭제된 후 오류 메시지를 받습니다.
  - 애플리케이션 사용자를 보려면 **사용자 표시**를 누릅니다. [시스템 사용자] 페이지에서 사용자를 로그아웃시킬 수 있습니다.
  - 애플리케이션 삭제를 취소하려면 **취소**를 누릅니다.
6. 사용자나 다른 사용자가 애플리케이션을 열지 않은 경우 확인 프롬프트가 표시됩니다. 예를 눌러 애플리케이션을 삭제합니다.

## 애플리케이션 요소 로드

애플리케이션을 생성한 후 메타데이터, 멤버 목록, 규칙 및 보안 파일을 로드합니다. 개별 파일을 로드하거나, 로드할 파일을 여러 개 선택하거나, 모든 파일을 한 번에 로드할 수 있습니다. 여러 파일을 로드하면 파일이 적절한 순서로 로드됩니다. 로드하기 전에 파일을 스캔하여 확인할 수도 있습니다.

각 프로세스를 개별적으로 검토할 수 있도록 각 로드 프로세스에서 로그 파일이 생성됩니다. 로드 프로세스가 완료되면 오류를 볼 수 있도록 로그에 대한 링크가 표시됩니다. 로드 파일 중 하나가 성공적으로 완료되지 않은 경우 오류를 수정하고 재로드할 수 있습니다.

애플리케이션 요소 파일은 특정 파일 형식을 사용해야 하며, 로드 프로세스에 대해 여러 옵션을 사용할 수 있습니다. 다음 섹션을 참조하십시오.

- 애플리케이션 보안 로드
- 메타데이터 로드
- 멤버 목록 로드
- 규칙 로드



애플리케이션 요소를 로드하려면 다음을 수행합니다.

1. 애플리케이션을 엽니다.
2. **통합, 로드, 애플리케이션 요소** 순으로 선택합니다.
3. 로드할 파일 이름을 입력하거나 **찾아보기**를 눌러 로드할 파일을 찾습니다.
4. 로드 프로세스에 대한 옵션을 지정합니다.
5. **선택 사항: 스캔**을 눌러 파일 형식이 정확한지 확인합니다.
6. **로드**를 눌러 개별 파일을 로드하거나 **모두 로드**를 누릅니다.

기존 파일을 재로드하는 경우 해당 파일을 다시 사용할지를 묻는 경고 프롬프트가 표시됩니다. 다시 사용하려면 **예**를 누릅니다.



**팁:**

파일 옵션을 기본값으로 재설정하려면 **재설정** 또는 **모두 재설정**을 누릅니다.

7. **선택사항:** 로그 파일을 다운로드하려면 **로그 다운로드**를 누릅니다. **열기**를 눌러 로그 파일을 표시하거나 **저장**을 누르고 로컬에서 파일을 저장할 위치를 선택합니다.

## 애플리케이션 요소 추출

애플리케이션 요소를 추출하고, 텍스트 편집기에서 정보를 보고 수정한 다음 요소를 애플리케이션에 재로드할 수 있습니다. 이 기능은 여러 파일을 동시에 업데이트해야 하는 경우에 유용할 수 있습니다. 개별 파일을 추출하거나, 추출할 파일을 여러 개 선택하거나, 모든 파일을 한 번에 추출할 수 있습니다.

애플리케이션 요소 파일은 특정 파일 형식을 사용해야 하며, 추출 프로세스에 대해 여러 옵션을 사용할 수 있습니다. 다음 섹션을 참조하십시오.

- [애플리케이션 보안 추출](#)
- [메타데이터 추출](#)
- [멤버 목록 추출](#)
- [규칙 추출](#)

추출 프로세스가 완료되면 오류를 볼 수 있도록 로그에 대한 링크가 표시됩니다.

애플리케이션 요소를 추출하려면 다음을 수행합니다.

1. 애플리케이션을 엽니다.
2. **통합, 추출, 애플리케이션 요소** 순으로 선택합니다.
3. 추출 프로세스에 대한 옵션을 지정합니다.
4. **추출**을 눌러 개별 파일을 추출하거나 **모두 추출**을 누릅니다.
5. **다운로드**를 눌러 추출한 파일을 다운로드합니다.
6. **선택사항:** 로그 파일을 다운로드하려면 **로그 다운로드**를 누릅니다. **열기**를 눌러 로그 파일을 표시하거나 **저장**을 누르고 로컬에서 파일을 저장할 위치를 선택합니다.

## Financial Management 애플리케이션 복제

애플리케이션 콘텐츠를 복제하여 새 애플리케이션을 생성할 수 있습니다. 애플리케이션을 복제하려면 애플리케이션 관리자 역할이 있어야 하며 Shared Services에서 Financial Management 애플리케이션 생성자 역할에 대해 프로비저닝되어 있어야 합니다.

동일한 환경에 애플리케이션을 복제하고 이름을 바꿀 수도 있고 다른 프로젝트 및 클러스터에 새 애플리케이션을 등록할 수도 있습니다. 동일한 데이터베이스에서만 애플리케이션을 복제할 수 있습니다.

기본적으로 프로세스는 애플리케이션 프로파일 정보, 메타데이터 및 보안 정보를 복제합니다. 데이터 테이블, 태스크 감사 및 데이터 감사 테이블을 소스 애플리케이션에서 대상 애플리케이션으로 복제하도록 선택할 수도 있습니다.

애플리케이션을 복제하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리, 애플리케이션** 순으로 선택합니다.
3. 복제할 애플리케이션을 선택합니다.
4. **애플리케이션 복제**를 누르거나, **작업, 애플리케이션 복제** 순으로 선택하거나, 애플리케이션 이름을 마우스 오른쪽 버튼으로 누르고 **애플리케이션 복제**를 선택합니다.
5. **이름**에 생성할 새 애플리케이션의 애플리케이션 이름을 입력합니다.
6. **설명**에 애플리케이션 설명을 입력합니다.
7. 클러스터에서 애플리케이션이 등록될 클러스터를 선택합니다.
8. **사용자 관리 프로젝트**에서 애플리케이션이 등록될 프로젝트를 선택합니다.
9. **선택사항:** 데이터 감사 및 태스크 감사 테이블을 소스에서 대상 애플리케이션으로 복제하려면 **감사 테이블 복제** 옵션을 선택합니다.
10. **선택사항:** 데이터 테이블을 소스에서 대상 애플리케이션으로 복제하려면 **데이터 테이블 복제** 옵션을 선택합니다.
11. **복제**를 누릅니다.

복제 프로세스를 시작하면 관리 모드가 사용으로 설정됩니다. 모든 사용자가 해당 애플리케이션에서 로그아웃되며 프로세스가 완료될 때까지 다시 로그인할 수 없습니다.

애플리케이션 복제 태스크 진행 상태를 모니터링할 수 있는 관리 태스크 페이지가 표시됩니다.

시스템에서 소스 애플리케이션을 성공적으로 복제한 후 사용자가 지정한 클러스터 및 프로젝트에 새로 생성된 애플리케이션을 등록하면 사용자가 새 애플리케이션을 열 수 있습니다.

## 애플리케이션 임포트

동일 유형 데이터베이스를 사용하여 데이터베이스 간에 또는 EPM 시스템 간에 애플리케이션을 임포트할 수 있지만 두 EPM 시스템 모두 Oracle Database 또는 SQL Server를 사용해야 합니다. 소스 애플리케이션은 11.1.2.4까지의 이전 릴리스일 수 있지만 이전 애플리케이션을 업그레이드하려면 특정 Financial Management 서버의 **EPM System Configurator**에서 Financial Management 애플리케이션 업그레이드 태스크를 실행해야 합니다. 애플리케이션을 임포트하려면 애플리케이션 관리자 역할이 있어야 하며 Shared Services에서 Financial Management 애플리케이션 생성자 역할에 대해 프로비저닝되어 있어야 합니다.

동일한 환경에 애플리케이션을 импорт하고 이름을 바꿀 수도 있고 다른 프로젝트 및 클러스터에 새 애플리케이션을 등록할 수도 있습니다.

모든 연도 및 시나리오에 대한 데이터를 импорт할 수도 있고 특정 연도 및 시나리오를 선택할 수도 있습니다.

기본적으로 프로세스는 애플리케이션 프로파일 정보, 메타데이터 및 보안 정보를 복제합니다. 태스크 및 데이터 감사 정보를 소스 애플리케이션에서 대상 애플리케이션으로 импорт하도록 선택할 수도 있습니다.

### 애플리케이션 импорт 사전 필수 조건

애플리케이션을 импорт하려면 먼저 다음 단계를 수행해야 합니다.

- 소스 및 대상 데이터베이스 간에 데이터 통신 링크를 생성해야 합니다.
- Oracle 데이터베이스를 사용하는 경우 데이터베이스 링크를 생성해야 합니다.
- SQL Server 데이터베이스를 사용하는 경우 링크된 서버를 생성해야 합니다.
- HFM 데이터베이스에 데이터베이스 링크 또는 링크된 서버를 생성할 수 있는 권한이 있는지 확인해야 합니다.
- 역할을 통해 사용자를 프로비저닝하는 것이 아니라 데이터베이스 링크 또는 링크된 서버에 구성된 데이터베이스 사용자에게 대해 일회성 명시적 DB 권한을 부여해야 합니다.

애플리케이션을 импорт하려면 다음을 수행합니다.

1. 대상 EPM System에 로그인합니다.
2. **탐색, 관리, 통합 관리** 순으로 선택합니다.
3. **관리, 애플리케이션 импорт** 순으로 선택합니다.
4. Oracle 데이터베이스 간에 импорт하는 경우 **데이터베이스 링크**를 선택하고 SQL Server 데이터베이스 간에 импорт하는 경우 **링크된 서버**를 선택합니다.  
데이터베이스 링크 또는 링크된 서버를 선택하면 소스 애플리케이션이 나열됩니다.
5. 애플리케이션 목록에서 소스 애플리케이션을 선택합니다.
6. **대상 애플리케이션 이름** 및 **대상 애플리케이션 설명**을 지정합니다.  
소스 애플리케이션은 이 이름 및 설명으로 импорт됩니다.
7. 클러스터에서 애플리케이션이 등록될 클러스터를 선택합니다.
8. **사용자 관리 프로젝트**에서 애플리케이션이 등록될 프로젝트를 선택합니다.
9. **선택사항:** 데이터 감사 및 태스크 감사 테이블을 소스에서 대상 애플리케이션으로 импорт하려면 **감사 테이블 импорт** 옵션을 선택합니다.
10. **선택사항:** 데이터 테이블을 소스에서 대상 애플리케이션으로 импорт하려면 **데이터 테이블 импорт** 옵션을 선택합니다.  
데이터 테이블 임포트를 선택하는 경우 시나리오 및 연도 필터도 지정할 수 있습니다.
11. **선택사항:** 타겟 애플리케이션에 복사한 후 소스 데이터 테이블을 삭제하려면 **소스 데이터 삭제** 옵션을 선택합니다.
12. **임포트를 누릅니다.**

임포트 프로세스를 시작하면 시스템에서 데이터 집약적 작업임을 알려 주는 경고를 표시하고 작업을 계속할 것인지 묻습니다.

그런 다음, 애플리케이션 импорт 태스크 진행 상태를 모니터링할 수 있는 관리 태스크 페이지가 표시됩니다.

시스템에서 소스 애플리케이션을 성공적으로 импорт한 후 사용자가 지정한 클러스터 및 프로젝트에 새로 생성된 애플리케이션을 등록하면 사용자가 새 애플리케이션을 열 수 있습니다.

## Oracle 데이터베이스에 대한 데이터베이스 링크 생성

애플리케이션을 импорт하려면 먼저 소스 및 대상 데이터베이스 간에 데이터베이스 링크를 생성해야 합니다.

대상 데이터베이스에서 다음 단계를 수행합니다.

소스 Oracle DB, HFM Schema에 대한 데이터베이스 링크를 생성하려면 다음을 수행합니다.

1. sysdba 사용자로 로그인하여 HFM Schema에 대한 데이터베이스 링크를 생성할 권한을 부여합니다.

```
GRANT CREATE DATABASE LINK TO hfm;
```

2. 대상 시스템의 HFM Schema에 로그인하여 다음 명령을 수행합니다.

```
CREATE DATABASE LINK <link name> CONNECT TO<hfm schema name>IDENTIFIED BY HFM1 USING '///host name:<port>/<service name>';
```

예를 들어 호스트 **SLCK58001**에 있는 HFM Schema를 연결하고 Oracle이 포트 **1521**에서 실행 중인 경우 다음을 수행합니다.

```
CREATE DATABASE LINK ToTestSystem CONNECT TO HFM IDENTIFIED BY HFM1 USING '///slck58001.xxxx:1521/service name';
```

3. 검증 단계: 다음 명령은 소스 시스템의 애플리케이션을 나열해야 합니다.

```
Select * from HSX_DATASOURCES@ToTestSystem
```

4. 데이터베이스 링크를 삭제하는 명령은 다음과 같습니다.

```
drop database link ToTestSystem;
```

5. 모든 데이터베이스 링크를 나열하는 명령은 다음과 같습니다.

```
select * from all_db_links
```

## SQL Server 데이터베이스의 링크된 서버 생성

SQL Server 데이터베이스를 사용하는 경우 애플리케이션을 импорт하려면 먼저 대상 및 소스 데이터베이스 간에 링크된 서버를 생성해야 합니다.

대상 데이터베이스에서 다음 단계를 수행합니다.

소스 SQLServer 인스턴스에 링크된 서버를 생성하려면 다음을 수행합니다.

1. 대상 HFM 데이터베이스에서 다음 4개의 명령(각각 exec로 시작)을 순서대로 실행합니다.

```
exec sp_addlinkedserver N'<linked server name>',@srvproduct=N'',
@provider=N'SQLNCLI', @datasrc=N'<host name>'
```

```
exec sp_addlinkedsrvlogin N'<linked server
name>',@useself=N'FALSE',@rmtuser=N'<hfm db>',@rmtpassword=N'<hfm db
password>'
```

```
exec sp_serveroption @server=N'<linked server
name>',@optname=N'rpc',@optvalue=N'true'
```

```
exec sp_serveroption @server=N'<linked server name>',@optname=N'rpc
out',@optvalue=N'true'
```

예를 들어 **stm10082**의 **sqlserver**에 대해 이름이 **ToTestSystem**인 링크된 서버를 생성하며 소스 sqlserver HFM 데이터베이스 인증서가 **hfm/hyperion**에 있는 경우 다음을 수행합니다.

```
exec sp_addlinkedserver
N'ToTestSystem',@srvproduct=N'',@provider=N'SQLNCLI',@datasrc=N'STM10082.xxx.COM'
```

```
exec sp_addlinkedsrvlogin
N'ToTestSystem',@useself=N'FALSE',@rmtuser=N'hfm', @rmtpassword=N'password'
```

```
exec
sp_serveroption@server=N'ToTestSystem',@optname=N'rpc',@optvalue=N'true'
```

```
exec sp_serveroption @server=N'ToTestSystem',@optname=N'rpc
out',@optvalue=N'true'
```

2. 검증 단계: 다음 명령은 소스 SQL Server의 애플리케이션을 나열해야 합니다.

```
select * from [ToTestsystem].[hfm].[dbo].HSX_DATASOURCES;
```

3. 링크된 서버를 삭제하는 명령은 다음과 같습니다.

```
sp_dropserver'<linked server name>', 'droplogins'
```

```
Example: sp_dropserver 'ToTestSystem', 'droplogins'
```

4. 모든 링크된 서버를 나열하는 명령은 다음과 같습니다.

```
sp_linkedservers
```

## 시스템 메시지 작업

시스템 메시지 로그에 Oracle Hyperion Financial Management 시스템 메시지 목록이 표시됩니다. 로그에서 메시지 요약과 개별 메시지의 세부정보를 보고 메시지를 인쇄할 수 있습니다. 오류 메시지의 경우 X가 표시되고 정보 메시지의 경우 i가 표시됩니다. 로그에서 시스템 메시지를 삭제할 수 있습니다. 메시지는 삭제할 때까지 로그에 유지됩니다.

시스템 메시지를 보려면 애플리케이션 서버를 구성할 때 지정한 관리자 그룹의 멤버여야 합니다. 관리자 그룹에 사용자를 지정하려면 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

다음 절차를 참조하십시오.

- [시스템 메시지 보기](#)
- [시스템 메시지 삭제](#)

## 시스템 메시지 보기

시스템 메시지를 보려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리, 시스템, 메시지** 순으로 선택합니다.
3. **선택 사항: 필터** 패널에서 필터 기준을 선택합니다.
  - 애플리케이션
  - 서버
  - 날짜 범위
    - 모든 날짜를 보려면 **모두 포함**을 선택합니다.
    - 날짜 범위를 지정하려면 **범위**를 선택하고 **분, 시간, 일** 또는 **월**을 지정합니다.
    - 특정 날짜를 지정하려면 **사용자정의**를 선택하고 **시작 및 종료** 날짜를 입력하거나 달력 아이콘을 눌러 날짜를 선택합니다.
4. 시스템 메시지 목록에서 세부정보를 보려는 메시지를 선택합니다.
5. 메시지를 두 번 눌러 열거나, **보기**를 누르거나, **작업, 보기** 순으로 선택합니다.
6. 시스템 메시지 세부정보 보기를 완료하면 **확인**을 누릅니다.

## 시스템 메시지 삭제

관리자 그룹에 지정된 경우 시스템 메시지 로그에서 시스템 메시지를 삭제할 수 있습니다.

시스템 메시지를 삭제하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리, 시스템, 메시지** 순으로 선택합니다.
3. 시스템 메시지 목록에서 삭제할 메시지를 하나 이상 선택합니다.
4. 다음 작업 중 하나를 수행합니다.
  - 선택한 메시지를 삭제하려면 **선택 항목 삭제**를 누르거나 **작업, 선택 항목 삭제** 순으로 선택합니다.

- 현재 필터 설정을 기준으로 메시지를 삭제하려면 **현재 필터 설정에 따라 삭제를 누르거나 작업, 필터링된 항목 삭제** 순으로 선택합니다.
- 모든 시스템 메시지를 삭제하려면 **모두 삭제를 누르거나 작업, 모두 삭제** 순으로 선택합니다.

 **팁:**

메시지 목록을 새로고치려면 **새로고침을 누르거나 작업, 새로고침** 순으로 선택합니다.

## 시스템 메시지 세부정보 문자열

일부 프로세스는 오류와 같은 시스템 메시지에 대한 기술 정보 문자열을 반환합니다. 문자열에는 고유 식별 오류 참조 번호와 다양한 정보 필드가 포함됩니다. 다음 예와 같이 필드는 세미콜론으로 구분되고 각 필드의 레이블 뒤에는 콜론이 붙습니다.

```
Error Reference Number: {219EB33B-BF50-11D6-A43E-0000863DCCF1}
```

```
Num: 0x800415c6; Type: 1; DTime: 1/3/12 12:20:10 PM; Svr: SERVER1; File: CHsxServerImpl.cpp; Line: 1842; Ver: 3.0.0.196;
```

다음 표에서는 시스템 메시지 필드에 대해 설명합니다.

**표 2-1 시스템 메시지 세부정보 문자열의 필드**

필드	설명
Num	16진수 형식의 오류 번호
유형	<i>내부에서만 사용</i>
DTime	오류 타임스탬프
Svr	오류가 발생한 컴퓨터의 컴퓨터 이름입니다.
File	오류가 적용되는 소스 코드 파일의 이름입니다.
Line	오류가 적용되는 소스 코드 파일의 라인 번호입니다.
Ver	오류가 적용되는 DLL의 버전 번호입니다.

## 애플리케이션 액세스 관리

기능 관리를 사용하여 다음과 같은 애플리케이션의 액세스 관련 설정을 제어할 수 있습니다.

- 지정된 애플리케이션의 모든 사용자 로그아웃. [사용자 로그아웃](#)을 참조하십시오.
- 모든 사용자의 애플리케이션 액세스 사용/사용 안함 설정. [연결 비활성화 및 활성화](#)를 참조하십시오.

애플리케이션 서버를 구성할 때 지정한 관리자 그룹의 멤버만 액세스 관련 설정을 관리할 수 있습니다. 관리자 그룹에 사용자를 지정하려면 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

## 사용자 로그아웃

백업 및 복원과 같은 시스템 전체 프로세스를 수행하기 위해 애플리케이션이나 애플리케이션 서버에서 모든 사용자가 로그오프되게 할 수 있습니다.

애플리케이션 서버를 구성할 때 지정한 관리자 그룹의 멤버만 사용자를 로그아웃할 수 있습니다.

로그아웃 작업은 선택한 애플리케이션의 모든 사용자가 강제로 로그아웃되게 합니다.

### 주:

선택한 사용자가 로그아웃되게 하려면 **시스템 사용자 관리**를 참조하십시오.

애플리케이션이나 서버에서 모든 사용자가 로그오프되게 하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리(Administration), 시스템, 관리(Manage)** 순으로 선택합니다.
3. **모든 사용자 로그아웃**을 누르거나 **작업, 모든 사용자 로그아웃** 순으로 선택하고 확인 프롬프트에서 **예**를 누릅니다.
4. [로그아웃 결과] 대화상자에서 **확인**을 누릅니다.

## 연결 비활성화 및 활성화

연결을 비활성화하면 새 사용자가 특정 서버나 애플리케이션에 로그인할 수 없습니다. 사용자 로그아웃 기능과 함께 연결 비활성화 기능을 사용할 수 있습니다. 예를 들어 애플리케이션에 로그인을 비활성화하고 애플리케이션에 로그인된 사용자를 로그아웃한 다음 메타데이터를 로드하고 애플리케이션 연결을 활성화할 수 있습니다.

사용자 액세스에 사용할 수 있는 옵션은 다음과 같습니다.

- **연결 사용 안함** - 이 옵션은 관리자를 비롯한 모든 사용자의 연결을 사용 안함으로 설정합니다. 어떤 사용자도 선택한 애플리케이션에 액세스할 수 없습니다.
- **관리자에 대해서만 연결 사용** - 관리자에게만 애플리케이션 액세스가 허용됩니다.
- **모든 사용자에게 대해 연결 사용** - 모든 사용자에게 애플리케이션 액세스가 허용됩니다.

애플리케이션이나 서버에 대한 사용자 연결을 비활성화하거나 활성화하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리(Administration), 시스템, 관리(Manage)** 순으로 선택합니다.
3. 다음 작업 중 하나를 수행합니다.
  - **연결 사용 안함**을 누르거나 **작업, 연결 사용 안함** 순으로 선택합니다.
  - **관리자에 대해서만 연결 사용**을 누르거나 **작업, 관리자에 대해서만 연결 사용** 순으로 선택합니다.
  - **모든 사용자에게 대해 연결 사용**을 누르고 **작업, 모든 사용자에게 대해 연결 사용**을 선택합니다.



4. 연결 목록을 새로고치려면 **새로고침**을 누르거나 **작업, 새로고침** 순으로 선택합니다.

## 애플리케이션 연결 상태 보기

서버 목록 또는 애플리케이션 목록별로 연결 상태를 볼 수 있습니다.

서버별로 보는 경우 서버 이름, 해당 서버에 적용 가능한 애플리케이션 목록이 해당 상태(사용 또는 사용 안함)와 함께 표시됩니다.

애플리케이션별로 보는 경우 애플리케이션 이름, 애플리케이션이 상주하는 서버 및 해당 상태(사용 또는 사용 안함)가 표시됩니다.

애플리케이션 연결 상태를 보려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리(Administration), 시스템, 관리(Manage)** 순으로 선택합니다.
3. 메뉴 아이콘을 사용하여 다음과 같이 표시 유형 간에 토글할 수 있습니다.
  - 서버에서 목록을 표시하는 경우 **애플리케이션별로 보려면 누르기**를 선택합니다.
  - 애플리케이션별로 목록을 표시하려면 **서버별로 보려면 누르기**를 선택합니다.

## 시스템 사용자 관리

[시스템 사용자] 기능을 사용하면 시스템의 사용자를 보고 애플리케이션이나 서버의 사용자를 로그오프할 수 있습니다. 사용자가 사용 중인 모듈 및 수행 중인 작업을 볼 수 있습니다. [사용자 보기](#) 및 [사용자 로그아웃](#)을 참조하십시오.



**주:**

시스템의 사용자를 보려면 Financial Management 관리자 보안 역할이 지정되어야 합니다.

애플리케이션 서버를 구성할 때 지정한 관리자 그룹의 멤버만 사용자를 로그아웃할 수 있습니다. 관리자 그룹에 사용자를 지정하려면 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

## 사용자 보기

모든 애플리케이션과 서버에 대해 로그인한 사용자를 보거나 목록을 필터링하여 특정 서버와 애플리케이션의 사용자를 볼 수 있습니다.



**주:**

시스템의 사용자를 보려면 Financial Management 관리자 보안 역할이 지정되어야 합니다.

각 로그인된 사용자에 대해 제공되는 정보는 다음과 같습니다.

- 사용자 이름
- 사용자가 작업 중인 현재 모듈
- 사용자의 현재 작업
- 작업이 시작된 시간
- 서버 이름
- 애플리케이션 이름
- 상태

로그온된 사용자를 보려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리, 시스템, 사용자** 순으로 선택합니다.
3. **선택 사항:** 필터 패널에서 애플리케이션 또는 서버를 기준으로 사용자를 필터링합니다.

## 선택한 사용자 또는 모든 사용자 로그아웃

백업 및 복원과 같은 시스템 전체 프로세스를 수행하기 위해 애플리케이션이나 애플리케이션 서버에서 사용자를 로그오프할 수 있습니다. 예를 들어 서버에 로그인된 사용자를 로그오프하고 더 이상 서버에 로그인하지 못하도록 할 수 있습니다. [연결 비활성화 및 활성화](#)를 참조하십시오.

애플리케이션 서버를 구성할 때 지정한 관리자 그룹의 멤버만 사용자를 로그아웃할 수 있습니다.

사용자를 로그아웃해도 즉시 연결이 해제되지는 않습니다. 사용자가 로그아웃되기 전에 수행 중인 프로세스가 완료되는 동안 5분 정도 지연될 수 있습니다.

IIS(Microsoft Internet Information Services)에서 제한 시간 설정을 변경하거나 EPM Configurator에서 [웹 세션 시간 초과] 설정을 변경하여 사용자 세션 시간 초과를 제어할 수 있습니다. 기본 제한 시간 설정은 20분입니다.

시스템에서 사용자를 로그오프하면 사용자에 대해 [중지됨] 상태가 표시됩니다. 또한 로그아웃 후 첫 번째 사용자 작업 뒤에 즉시 로그오프된 사용자에게 알립니다.

애플리케이션이나 서버에서 사용자를 로그오프하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리, 시스템, 사용자** 순으로 선택합니다.
3. 사용자 목록에서 로그아웃할 사용자를 하나 이상 선택합니다.
4. 다음 작업 중 하나를 수행합니다.
  - **선택한 사용자 로그아웃**을 누르거나 **작업, 선택한 사용자 로그아웃** 순으로 선택합니다.
  - **모든 사용자 로그아웃**을 누르거나 **작업, 모든 사용자 로그아웃** 순으로 선택합니다.

### **팁:**

사용자 목록을 새로고치려면 **새로고침**을 누르거나 **작업, 새로고침** 순으로 선택합니다.

## 서버 및 애플리케이션 관리

서버 및 애플리케이션을 관리하려면 관리자여야 합니다.

관련 항목:

- 관리 모드 활성화 및 비활성화
- 서버 동기화

### 관리 모드 활성화 및 비활성화

관리자인 경우 관리 모드를 활성화하여 백업 및 복원 태스크와 같은 관리 유지 태스크를 수행하는 동안 사용자가 로그인하지 못하도록 할 수 있습니다.

애플리케이션을 관리 모드로 변경하면 모든 사용자가 시스템에서 로그아웃됩니다. 대기 중인 태스크(예: 통합, 데이터 로드 또는 메타데이터 로드)이 있는 경우에는 애플리케이션이 관리 모드로 변경되지 않고 종료 전에 대기 중인 태스크가 완료될 때까지 기다립니다. 태스크가 완료되면 애플리케이션에 대해 관리 모드를 활성화할 수 있습니다.

관리 모드를 비활성화할 때까지 사용자는 애플리케이션에 로그인할 수 없습니다.

관리 모드를 활성화하거나 비활성화하려면 다음을 수행합니다.

1. 탐색, 관리, 통합 관리 순으로 선택합니다.
2. 시스템, 애플리케이션 순으로 선택합니다.
3. 애플리케이션을 선택합니다.
4. 다음 작업 중 하나를 수행합니다.
  - 툴바에서 관리 모드 버튼을 누릅니다.
  - 작업, 관리 모드 순으로 선택합니다.
  - 애플리케이션 이름을 마우스 오른쪽 버튼으로 누르고 관리 모드를 선택합니다.
5. 애플리케이션에 대한 작업을 마쳤으면 애플리케이션을 선택하고 관리 모드를 비활성화합니다.

### 서버 동기화

Oracle Hyperion Financial Management 애플리케이션 서버 간의 동기화는 시스템 시간을 기준으로 합니다. 시계를 변경하면 이러한 동기화에 영향을 미칠 수 있습니다. 일광 절약 시간으로 변경하거나 DST(일광 절약 시간)에서 변경하는 경우 시간을 변경하기 전에 서버를 중지하고 시간을 변경한 후에 다시 시작하는 것이 좋습니다.

### 태스크 감사

[태스크 감사] 기능을 사용하여 사용자가 수행한 태스크를 볼 수 있습니다. 날짜 범위, 애플리케이션 서버, 사용자 및 수행된 태스크를 기준으로 감사 태스크를 필터링할 수 있습니다.

[애플리케이션 관리자] 역할이 지정된 경우 태스크 감사 정보를 보고 내보내고 삭제할 수 있습니다. 관리자는 아니지만 [태스크 감사 보기] 역할이 할당된 경우 태스크 감사 정보를 보고 내보낼 수 있습니다.

태스크 감사에 기록되는 사용자 태스크는 다음과 같습니다.

- 할당
- 계정별 자동 일치
- ID별 자동 일치
- 차트 논리(계산)
- 통합
- 트랜잭션 생성
- 데이터 감사 제거
- 데이터 지우기
- 데이터 복사
- 데이터 입력
- 데이터 추출
- 데이터 로드
- 데이터 잠금
- 데이터 잠금 해제됨
- 트랜잭션 모두 삭제
- 부적합한 레코드 삭제
- 트랜잭션 삭제
- 문서 첨부
- 문서 첨부 취소
- 트랜잭션 편집
- IC 기간 관리
- IC 원인 코드 관리
- IC 수동 일치
- 계정별 IC 일치 보고서
- ID별 IC 일치 보고서
- IC 트랜잭션 보고서
- 유희
- 분개 입력
- 분개 기간 마감됨
- 분개 기간 개설됨
- 분개 게시
- 분개 템플릿 항목
- 분개 게시취소
- 엔티티 잠금/잠금 해제
- 로그오프
- 로그인
- 멤버 목록 추출

- 멤버 목록 로드
- 멤버 목록 스캔
- 메타데이터 추출
- 메타데이터 로드
- 메타데이터 로드 차이
- 메타데이터 스캔
- 애플리케이션 수정
- 트랜잭션 모두 게시
- 트랜잭션 게시
- 규칙 추출
- 규칙 로드
- 규칙 스캔
- 보안 추출
- 보안 로드
- 태스크 감사 제거
- 트랜잭션 추출
- 트랜잭션 로드
- 트랜잭션 스캔
- 환산
- 트랜잭션 모두 대응 취소
- IC 트랜잭션 대응 취소
- 트랜잭션 모두 게시취소
- 트랜잭션 게시취소

태스크 감사 로그에 포함되는 정보는 다음과 같습니다.

- 사용자 이름
- 수행된 작업
- 작업 시작 시간
- 작업 종료 시간
- 서버 이름
- 설명
- 현재 모듈

태스크 감사 로그 정보는 (*APPNAME\_TASK\_AUDIT*) 테이블에 저장됩니다. 테이블에 있는 정보를 백업하거나 파일에 추출한 후 테이블을 지울 수 있습니다. 로그 크기를 모니터링하고 정기적으로 지워야 합니다.

태스크 감사를 수행하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리, 감사, 태스크** 순으로 선택합니다.

3. **선택 사항:** 필터 패널에서 필터 기준을 선택합니다.
  - 애플리케이션
  - 서버
  - 날짜 범위
    - 모든 날짜를 보려면 **모두 포함**을 선택합니다.
    - 날짜 범위를 지정하려면 **범위**를 선택하고 **분**, **시간**, **일** 또는 **월**을 지정합니다.
    - 특정 날짜를 지정하려면 **사용자정의**를 선택하고 **시작** 및 **종료** 날짜를 입력하거나 달력 아이콘을 눌러 날짜를 선택합니다.
  - 사용자
  - 태스크
4. **선택사항:** 감사 정보를 CSV 파일로 익스포트하려면 **익스포트**를 누르거나 **작업**, **익스포트** 순으로 선택하고 다운로드 지침을 따릅니다.
5. **선택사항:** 현재 필터 설정을 기준으로 항목을 삭제하려면 **현재 필터 설정에 따라 삭제**를 누르거나 **작업**, **필터링된 항목 삭제** 순으로 선택합니다.
6. **선택사항:** 로그에서 모든 항목을 삭제하려면 **모두 삭제**를 누르거나 **작업**, **모두 삭제** 순으로 선택합니다.

 **주:**

로그를 지울 때 지우기 프로세스의 레코드는 로그에 유지되며 지울 수 없습니다.

## 데이터 감사

[데이터 감사] 기능을 사용하여 사용자가 수행한 데이터 변경사항을 볼 수 있습니다. 날짜 범위, 애플리케이션 서버, 사용자 및 차원 멤버를 기준으로 데이터 변경사항을 필터링할 수 있습니다.

데이터 변경사항을 감사할 시나리오와 계정에 `EnableDataAudit` 메타데이터 속성을 사용할 수 있습니다. 시나리오 감사 설정이 계정 감사 설정을 재정의합니다. 시나리오에 대해 `EnableDataAudit` 속성을 `Yes`로 설정하면 `EnableDataAudit`가 `False`로 설정된 계정을 포함하여 시나리오의 모든 계정이 감사됩니다. 시나리오에 대해 `EnableDataAudit` 속성을 `Override`로 설정하면 `EnableDataAudit`가 `True`로 설정된 모든 계정이 감사됩니다. 시나리오 및 계정 멤버의 감사를 비활성화하려면 `EnableDataAudit` 속성을 `No`로 변경합니다.

[애플리케이션 관리자] 역할이 지정된 경우 데이터 감사 정보를 보고 내보내고 삭제할 수 있습니다. 관리자는 아니지만 [데이터 감사 보기] 역할이 할당된 경우 데이터 감사 정보를 보고 내보낼 수 있습니다.

데이터 감사에 기록되는 사용자 작업은 다음과 같습니다.

- 데이터 입력
- 데이터 지우기
- 데이터 복사
- 데이터 로드

- 분개 입력

데이터 감사 로그에 포함되는 정보는 다음과 같습니다.

- 사용자 이름
- 수행된 작업
- 수정한 시간
- 서버 이름
- POV
- POV에 대해 입력된 값

데이터 감사 로그 정보는 `APPNAME_DATA_AUDIT` 테이블에 저장됩니다. 테이블에 있는 정보를 백업하거나 추출할 수 있습니다. 로그 크기를 모니터하고 정기적으로 지워야 합니다.

데이터 감사를 수행하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리, 감사, 데이터** 순으로 선택합니다.
3. **선택 사항:** [POV]에서 차원을 눌러 차원 멤버를 선택합니다.
4. **선택 사항:** 필터 패널에서 필터 기준을 선택합니다.
  - 애플리케이션
  - 서버
  - 날짜 범위
    - 모든 날짜를 보려면 **모두 포함**을 선택합니다.
    - 날짜 범위를 지정하려면 **범위**를 선택하고 **분, 시간, 일** 또는 **월**을 지정합니다.
    - 특정 날짜를 지정하려면 **사용자정의**를 선택하고 **시작** 및 **종료** 날짜를 입력하거나 달력 아이콘을 눌러 날짜를 선택합니다.
  - 사용자
5. **선택사항:** 감사 정보를 CSV 파일로 익스포트하려면 **익스포트**를 누르거나 **작업, 익스포트** 순으로 선택하고 다운로드 지침을 따릅니다.
6. **선택사항:** 현재 필터 설정을 기준으로 항목을 삭제하려면 **현재 필터 설정에 따라 삭제**를 누르거나 **작업, 필터링된 항목 삭제** 순으로 선택합니다.
7. **선택사항:** 로그에서 모든 항목을 삭제하려면 **모두 삭제**를 누르거나 **작업, 모두 삭제** 순으로 선택합니다.

#### 주:

로그를 지울 때 지우기 프로세스의 레코드는 태스크 감사 로그에 유지되며 지울 수 없습니다.

## 실행 중인 태스크 모니터

[실행 중인 태스크] 모듈을 사용하여 실행 중인 태스크를 보고 종료할 수 있습니다. 애플리케이션 서버, 사용자, 수행된 태스크 및 태스크 상태를 기준으로 실행 중인 태스크를 필터링할 수 있습니다.

기본적으로 실행 중인 태스크는 데이터베이스에 900초(15분) 동안 유지됩니다.

AutoClearDeadTasks 시스템 설정을 수정하여 기본값인 15분이 지난 뒤 태스크 실행을 유지할 것인지 제거할 것인지 변경할 수 있습니다. [사용가능 구성 설정](#)을 참조하십시오.

- [실행 중인 태스크 보기](#)
- [실행 중인 태스크 중지](#)
- [실행 중인 태스크 새로고침](#)

## 실행 중인 태스크 보기

실행 중인 태스크를 보려면 다음을 수행합니다.

1. 다음 작업 중 하나를 수행합니다.
  - 애플리케이션에서 **통합**, **유지관리**, **실행 중인 태스크** 순으로 선택합니다.
  - **탐색**, **관리(Administer)**, **통합 관리**, **관리(Administration)**, **시스템**, **실행 중인 태스크** 순으로 선택합니다.
2. **선택 사항**: 필터 패널에서 필터 기준을 선택합니다.
  - 애플리케이션
  - 서버
  - 사용자
  - 태스크
  - 상태

다음과 같은 태스크를 보고 종료할 수 있습니다.

- 통합
- 데이터 로드
- 데이터 추출
- 데이터를 데이터베이스로 추출 익스포트
- 트랜잭션 모두 게시
- 트랜잭션 모두 게시취소
- 트랜잭션 모두 삭제
- 트랜잭션 모두 대응 취소
- ID별 자동 일치
- 계정별 자동 일치
- IC 일치 보고서
- ID별 IC 일치 보고서
- 계정별 IC 일치 보고서
- IC 트랜잭션 보고서
- 분개 보고서

다음과 같은 상태 유형을 기준으로 실행 중인 태스크를 필터링할 수 있습니다.

- 초기화 중



- 실행 중
- 일시 중지됨
- 초기화 취소
- 중지됨
- 중단됨
- 완료됨
- 응답 없음
- 스케줄링된 중지
- 스케줄링된 시작

## 실행 중인 태스크 중지

태스크를 시작한 사용자나 관리자 역할에 지정된 사용자만 실행 중인 태스크를 종료할 수 있습니다.

태스크를 중지하려면 다음을 수행합니다.

1. [실행 중인 태스크] 모듈에서 중지할 태스크를 선택합니다.
2. 태스크 중지를 누르거나 **작업, 태스크 중지** 순으로 선택합니다.

## 실행 중인 태스크 새로고침

여러 데이터 또는 많은 데이터를 로드하거나 내부거래 트랜잭션 파일을 로드하는 경우 [실행 중인 태스크] 모듈에 태스크가 즉시 표시되지 않을 수 있습니다. [실행 중인 태스크]을 열 때 데이터나 내부거래를 로드하는 동안 파일 전송이 지연되면 시스템에서 기다리라는 메시지가 표시되고 화면이 자동으로 새로 고쳐진 후 해당 태스크가 표시됩니다. 기본적으로 시스템은 [실행 중인 태스크] 화면을 5초마다 자동으로 새로 고치고 최대 120회까지 자동 새로고침 프로세스를 수행합니다. 120회 이후에 실행 중인 태스크가 없으면 새로고침 주기가 중지되므로 수동 새로고침을 수행해야 합니다. 새로고침 주기는 실행 중인 모든 태스크가 완료될 때까지 계속됩니다.

### 주:

Financial Management System 관리자인 경우 기본 새로고침 횟수 및 기본 새로고침 간격을 구성할 수 있습니다. [구성 설정 변경](#)을 참조하십시오.

실행 중인 태스크 목록을 새로 고치려면 다음을 수행합니다.

1. 태스크 목록에서 **새로고침**을 누르거나 **작업, 새로고침** 순으로 선택합니다.
2. 드롭다운 목록에서 새로고침 모드를 선택합니다.
  - 기본값
  - 수동
  - 5초
  - 10초
  - 15초
  - 30초

- 60초



주:

새로고침 모드 설정은 다음에 애플리케이션에 로그인할 때까지 유지됩니다.

## 부적합한 레코드 스캔 및 지우기

[부적합한 레코드 지우기] 기능을 사용하여 애플리케이션에서 부적합한 레코드를 스캔하고 제거할 수 있습니다. 부적합한 레코드를 지우려면 관리자 보안 액세스 권한이 있어야 합니다.

[부적합한 레코드 지우기] 프로세스를 실행하면 데이터베이스, 네트워크 및 Oracle Hyperion Financial Management 환경에 영향을 주며 성능 이슈가 발생할 수 있습니다. 이 프로세스는 사용자가 Financial Management 환경에 액세스하지 않는 유지 관리 창에서 실행해야 합니다.

부적합한 레코드를 스캔하고 삭제하려면 다음을 수행합니다.

1. 부적합한 레코드를 삭제할 애플리케이션을 엽니다.
2. 통합, 데이터, 관리 순으로 선택합니다.
3. [데이터 관리] 페이지에서 **부적합한 레코드 지우기**를 확장합니다.



주:

이 기능은 [관리자] 보안 액세스 권한이 있는 사용자만 사용할 수 있습니다.

4. 작업을 선택합니다.
  - 부적합한 레코드를 스캔하려면 **스캔**을 누릅니다.
  - 레코드를 지우려면 **부적합한 레코드 지우기**를 누릅니다.

## HFM 인사이트를 사용하여 시스템 상태 모니터

HFM 인사이트는 시스템 관리자가 Oracle Hyperion Financial Management 시스템 상태를 모니터할 수 있는 대시보드를 제공합니다. 일정 기간 동안 시스템 구성요소에서 수집한 데이터를 표시하고 관리자가 필요한 경우 정정 조치를 수행할 수 있습니다.

## HFM 인사이트 액세스

HFM 인사이트에 액세스하려면 Oracle Hyperion Financial Management 애플리케이션 관리자 역할이 있어야 합니다.



주:

모니터링을 위해 애플리케이션 프로세스가 작동 중일 필요는 없습니다.

HFM 인사이트에 액세스하려면 다음을 수행합니다.

1. 탐색, 관리, 통합 관리 순으로 선택합니다.
2. 시스템, 인사이트 순으로 선택합니다.

## HFM 인사이트 기본 대시보드

HFM 인사이트를 열면 기본 페이지에서는 각 서버에서 실행 중인 애플리케이션과 해당 상태가 각 행에 표시된 테이블을 보여줍니다. 다중 서버 환경에서는 서버별로 애플리케이션을 필터링할 수 있습니다.

테이블에는 필터에서 선택한 각 애플리케이션 및 서버 조합 행이 포함되어 있습니다. 예를 들어 세 개의 애플리케이션과 두 개의 서버를 선택한 경우 테이블에는 6개의 행이 표시됩니다(서버 1의 애플리케이션 1, 서버 2의 애플리케이션 2 등). 각 행에 대한 다음 정보가 표시됩니다.

- 애플리케이션 아이콘, 애플리케이션 이름 및 서버 이름
- 다음 KPI(주요 성과 지표)를 기반으로 하는 차트. 데이터는 다음과 같이 최근 2주로 제한됩니다.
  - 사용자 - 지난 2주 동안의 사용자 수
  - 오류 - 지난 2주 동안의 오류 수
  - 태스크 - 지난 2주 동안의 태스크 수
  - 메모리 - 물리적 메모리(MB)
  - CPU - CPU 사용량
- 상태 개요:
  - 상태 아이콘은 다음 상태 중 하나를 표시합니다.
    - \* 관리 모드
    - \* 충돌
    - \* 다운(프로세스가 로드되지 않음)
    - \* 응답하지 않음
    - \* 작동(프로세스가 가동 중임)
  - 시작 시간 - 마지막으로 XFMDatasource 프로세스가 시작된 시간입니다.
  - 작동 시간 - 작동 시간 계산에서는 시작 시간과 최종 ping 시간 사이의 차이를 구합니다.
  - 최종 상태 검사 - HsxServer 로그의 XMFDataSource 프로세스에 대한 최종 성공 ping 타임스탬프입니다.

기본적으로 모든 메트릭이 표시됩니다. 개별 그래프에서 메트릭을 선택 취소할 수 있습니다. 예를 들어 CPU만 선택하고 다른 메트릭은 선택 취소할 수 있습니다. 그래프는 선택 항목에 따라 스케일링됩니다.

언제든 [새로고침] 옵션을 사용하여 화면을 새로고칠 수 있습니다.

## HFM 인사이트에서 애플리케이션 필터링

**애플리케이션 필터** - 자신이 애플리케이션 관리자인 모든 애플리케이션을 나열합니다. 하나 이상의 애플리케이션을 선택할 수 있습니다. 기본적으로 목록의 처음 다섯 개 애플리케이션이 선택되며 해당 애플리케이션의 상태가 테이블에 표시됩니다.

**서버 필터** - 사용가능한 모든 서버를 나열합니다. 하나 이상의 서버를 선택할 수 있습니다. 시스템에 서버가 하나만 있는 경우 서버 필터가 표시되지 않습니다. 서버 필터가 표시되면 기본적으로 첫번째 서버가 선택되고 해당 서버의 애플리케이션 상태가 테이블에 나열됩니다.

**필터 표시** - 그래프에 표시할 메트릭을 선택할 수 있습니다. 다중 선택 목록입니다. 기본적으로 모든 메트릭이 선택되며 그래프에는 사용자, 오류, 태스크, 메모리, CPU 등 모든 메트릭이 표시됩니다.

## HFM 인사이트의 애플리케이션 세부정보

애플리케이션 아이콘을 누르면 해당 서버에서 실행 중인 애플리케이션에 대한 추가 세부 메트릭이 표시된 새 탭이 열립니다.

탭 제목은 <애플리케이션 이름>@<서버 이름>입니다.

- **표시** - 표시할 기간에 대한 필터입니다. 기본적으로 최근 14일 동안의 데이터가 표시됩니다.
- **새로고침** - [수동] 또는 [자동 새로고침]을 선택합니다. 기본값은 [수동]입니다.

애플리케이션 페이지는 다음 섹션을 보여줍니다.



### 애플리케이션 요약

- 애플리케이션 아이콘, 애플리케이션 이름 및 서버 이름
- 상태 개요, 기본 페이지에 있는 대로 표시
- **애플리케이션 차원** - 각 차원의 애플리케이션 차원 및 총 멤버 수
- **재시작** - XFMDatasource 프로세스를 재시작합니다.

### 통계

통계 섹션에는 선택한 시나리오 및 연도 조합을 기준으로 자세한 데이터 통계가 표시됩니다. 통계를 수동으로 업데이트할 시기를 선택할 수 있습니다.

통계를 업데이트하려면 다음을 수행합니다.

1. **통계** 섹션에서 **데이터** 탭을 누릅니다.
2. **멤버 선택기**()를 누르고 시나리오 및 연도를 선택한 다음 **확인**을 누릅니다.
3. **실행 버튼**()을 눌러 프로세스를 실행하여 선택한 시나리오 및 연도 조합에 대한 통계를 반환합니다.

### 주:

**실행** 버튼을 누르고 통계를 업데이트하면 프로세스가 실행되는 기간 동안 버튼이 사용 안함으로 설정됩니다. 태스크가 백그라운드에서 실행되며 실행되는 동안 애플리케이션의 다른 부분을 탐색할 수 있습니다. 완료되면 버튼이 다시 사용으로 설정되고 테이블이 업데이트된 통계로 새로고쳐집니다. **새로고침**을 눌러 업데이트된 값을 볼 수도 있습니다.

특정 테이블에 레코드가 없는 경우 해당 테이블에 대한 통계가 표시되지 않습니다.

## 데이터 탭

- 데이터(시나리오/연도 조합별 DCE 및 DCN 총 레코드 개수)
- 부적합한 레코드(부적합한 멤버 ID), DCE 및 DCN만 해당, 총 개수만(엔티티별 아님)
- 모든 기간의 0에 가까운 값 레코드
- 모든 기간의 0에 가까운 값 계산 레코드
- PFlow 내역(시나리오/연도별)
- 입력 데이터(연도의 특정 기간이 입력 또는 라인 항목 세부정보인 경우)

## 테이블 개수 탭

테이블 개수 탭에는 다음 Oracle Hyperion Financial Management 시스템 테이블의 레코드 개수와 최종 실행 시간이 표시됩니다.

- 태스크 감사
- 데이터 감사
- 시스템 메시지(오류 로그 테이블)
- 규칙 프로파일 데이터

## 성능 개요

성능 섹션에는 기본 화면과 동일한 KPI를 기반으로 차트가 표시됩니다.

## 사용자

**빈번한 사용자** - 태스크 감사 데이터를 기반으로 특정 기간 동안 세션 수가 가장 많은 사용자 상위 10명을 보여주는 막대 그래프입니다. 기본적으로 최근 14일 동안의 데이터가 표시되지만 [표시] 필터를 사용하여 기간을 변경할 수 있습니다.

**세션 세부정보** - 사용자 이름, 시작 시간, 종료 시간 및 기간을 비롯한 사용자 세션 세부정보입니다. 모든 열을 정렬할 수 있습니다.

사용자를 기준으로 필터링할 수 있습니다. 사용자 검색 상자에 전체 또는 부분 사용자 이름을 입력합니다.

특정 기간 동안 로그인한 사용자만 보려면 기간 검색 상자에 시간 값을 입력합니다. 예를 들어 1시간 동안 로그인한 사용자를 보려면 1h를 입력합니다. 퍼센트 기호(%)를 와일드카드로 사용할 수 있습니다. 예를 들어 1시간 미만인 기간의 사용자를 보려면 검색 상자에 %h를 입력합니다.

## 메모리

메모리 섹션은 시스템 메시지 데이터를 기반으로 시스템 메모리 매개변수 그래프를 보여줍니다. 포함되는 내용은 다음과 같습니다.

- 전체 메모리
- 사용된 메모리
- NumCubesInRAM
- NumDataRecordsInRAM
- NumRecordsInLargestCube
- MinDataCacheSizeInMB
- MaxDataCacheSizeInMB

- MaxNumCubesInRAM

메모리 매개변수에 대한 자세한 내용은 [사용가능 구성 설정](#)을 참조하십시오.

## 태스크

이 섹션에는 태스크 정보를 표시하는 네 개의 탭이 있습니다.

### 태스크 뷰

태스크 뷰는 태스크 감사 데이터를 기반으로, 선택된 시간 범위에 가장 자주 실행된 태스크 상위 10개를 보여줍니다. 예를 들어 지난 2주 동안 발생한 로그인 수 또는 통합을 신속하게 볼 수 있습니다.

- **메트릭** - 다음과 같은 빈도 시간 메트릭 필터입니다.
  - 빈도 - 선택된 시간 범위에 발생한 태스크 개수
  - 총 시간 - 선택된 시간 범위에 발생한 태스크에 걸린 총 시간
  - 평균 시간 - 선택된 시간 범위에 발생한 태스크에 걸린 평균 시간
- **사용자** - 태스크를 수행한 사용자를 기준으로 필터링합니다. 사용자 검색 상자에 사용자 이름 전체 또는 부분을 입력하거나 퍼센트 기호(%)를 와일드카드로 사용하십시오.

### 사용자 뷰

[사용자 뷰] 탭에는 특정 태스크를 수행한 상위 10명의 사용자를 보여주는 막대 차트가 있습니다.

- **메트릭** - 다음과 같은 태스크 시간 필터입니다.
  - 빈도 - 선택된 시간 범위에 사용자가 수행한 태스크 개수
  - 총 시간 - 선택된 시간 범위에 수행한 태스크에 걸린 총 시간
  - 평균 시간 - 선택된 시간 범위에 수행한 태스크에 걸린 평균 시간
- **태스크** - 태스크 기준 필터. 기본적으로 [통합]이 선택되어 있습니다.

### 월 뷰

월 뷰는 월별 기준으로 특정 태스크를 보여주는 막대 차트를 표시합니다. 태스크 활동을 월별로 비교할 수 있습니다.

- **메트릭** - 다음과 같은 태스크 시간 필터입니다.
  - 빈도 - 선택된 한 개 또는 여러 개 월에 수행한 태스크 개수
  - 총 시간 - 선택된 한 개 또는 여러 개 월의 태스크에 걸린 총 시간
  - 평균 시간 - 선택된 한 개 또는 여러 개 월의 태스크에 걸린 평균 시간
- **태스크** - 태스크 기준 필터. 기본적으로 [통합]이 선택되어 있습니다.

### 태스크 세부정보

이 탭은 태스크 감사 데이터를 기반으로 각 태스크에 대한 자세한 정보를 보여줍니다. 사용자 이름, 활동 이름, 시작 시간, 종료 시간, 기간 및 설명이 포함되어 있습니다. 모든 열을 정렬할 수 있습니다.

사용자, 활동, 기간 및 설명을 기준으로 필터링할 수 있습니다.

## 오류

오류 섹션은 시스템 메시지 데이터를 기준으로 가장 자주 발생하는 오류 10개를 표시합니다.

- **오류 빈도** - 오류 코드를 표시합니다. 오류 코드 위로 마우스를 가져가면 설명을 볼 수 있습니다.
- **오류 세부정보** - 특정 시간 범위에 대해 선택한 서버의 선택한 애플리케이션에 대한 오류 코드, 시스템 메시지 요약, 오류 날짜 및 시간을 표시합니다. 오류 코드 및 날짜-시간을 기준으로 정렬할 수 있습니다. 오류 코드 및 시스템 메시지 요약별로 필터링할 수 있습니다.

# 3

## 애플리케이션 보안 관리

보안 및 액세스 권한을 사용하면 Oracle Hyperion Financial Management 애플리케이션과 애플리케이션 요소에 대한 액세스를 제어할 수 있습니다. 보안을 설정하면 데이터를 보호하고 권한 없는 사용자의 데이터 변경을 방지할 수 있습니다. 예를 들어 애플리케이션 내의 특정 데이터 요소나 양식에 대한 액세스를 제한할 수 있습니다.

보안은 다음 두 레벨로 설정됩니다.

- 외부 제공자에 의한 인증
- Financial Management 보안. 이 경우 사용자와 그룹이 애플리케이션에 지정되고, 애플리케이션 요소가 보안 클래스에 지정됩니다.

다음 두 가지 방법으로 Financial Management 애플리케이션의 보안을 설정할 수 있습니다.

- 보안 파일을 애플리케이션에 로드합니다. [애플리케이션 보안 로드](#)를 참조하십시오.
- Oracle Hyperion Shared Services Console을 사용하여 보안 정보를 설정합니다. [Financial Management에서 Shared Services Console 실행](#)을 참조하십시오.

이러한 보안 역할은 애플리케이션 관리에 필요합니다. 역할 할당에 대한 자세한 내용은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

보안 역할	설명
Financial Management 관리자	관리 기능에 액세스하는 데 필요합니다.
차원 편집자	차원 생성을 위해 임포트 프로파일을 생성 및 관리하고 수동으로 차원을 생성 및 관리합니다. Classic 애플리케이션 관리 옵션에 액세스하는 데 필요합니다.
Financial Management 애플리케이션 생성자/ Financial Management Manager	애플리케이션을 생성합니다. 이 역할이 할당된 사용자는 애플리케이션을 생성할 수 있지만 액세스 권한이 있는 차원만 변경할 수 있습니다. Financial Management 애플리케이션 생성자는 통합 애플리케이션과 일반 애플리케이션을 생성할 수 있습니다. 또한 애플리케이션을 생성하려면 사용자가 Configuration Utility에서 지정된 [애플리케이션 생성자] 그룹의 멤버여야 합니다.

## 애플리케이션 보안 고려 사항

Oracle Hyperion Financial Management 보안을 사용하면 애플리케이션 요소와 태스크를 유연성 있게 보호할 수 있습니다. 애플리케이션 요소를 생성할 때 보안 클래스가 지정되므로 애플리케이션을 설정하기 전에 보안 시스템을 디자인해야 합니다.

한 애플리케이션에 대해 보안 시스템을 디자인한 후 다른 애플리케이션에 로드하거나 백업하기 위해 보안 요소를 추출할 수 있습니다. [애플리케이션 보안 로드](#) 및 [애플리케이션 보안 추출](#)을 참조하십시오.

Financial Management에서 보안을 설정하기 전에 다음 질문을 고려해야 합니다.



- Financial Management 태스크와 애플리케이션 요소를 어떤 방법으로 그룹화 및 분류하시겠습니까?
- 사용자를 어떤 방법으로 그룹화하시겠습니까?
- 사용자와 그룹에 대해 어떤 액세스 권한 레벨을 지정해야 합니까?
- 애플리케이션 요소를 생성할 때 어떤 보안 클래스를 지정하시겠습니까?

## Financial Management에서 Shared Services Console 실행

Oracle Hyperion Financial Management 애플리케이션의 보안을 설정하려면 먼저 다음 태스크를 수행해야 합니다.

1. Financial Management 애플리케이션을 생성합니다.
2. 애플리케이션에 사용자와 그룹을 지정하고 사용자 역할을 지정하여 사용자를 프로비저닝합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

그런 다음 Oracle Hyperion Shared Services Console을 사용하여 Financial Management 애플리케이션의 보안을 설정할 수 있습니다. 콘솔에서 다음과 같은 애플리케이션 태스크를 수행할 수 있습니다.

- 사용자 및 그룹 지정
- 보안 클래스에 사용자 권한 지정
- 보안 보고서 실행

Shared Services Console을 실행하려면 다음을 수행합니다.

1. Financial Management에서 **탐색, 관리** 순으로 선택합니다.
2. **Shared Services Console**을 선택합니다.

## 보안 클래스를 지정할 사용자 및 그룹 선택

프로비저닝 관리자 역할에 지정된 사용자만 사용자와 그룹을 정의할 수 있습니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

### 주:

Hyperion Financial Management에서는 사용자 및 그룹의 이름이 동일할 수 없습니다.

기본적으로 [액세스 제어] 페이지에는 프로비저닝된 처음 100개 그룹과 사용자가 표시됩니다. 테이블에 먼저 그룹이 표시된 다음 사용자가 표시됩니다. 테이블에서 사용자와 그룹을 해당 아이콘으로 식별할 수 있습니다.

애플리케이션의 사용자와 그룹을 선택하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에서 **애플리케이션 그룹**을 확장하고 애플리케이션 이름을 마우스 오른쪽 버튼으로 누른 다음 **액세스 제어 지정**을 선택합니다.
2. **사용자/그룹**을 선택하거나 **작업, 사용자/그룹** 순으로 선택합니다.

3. 옵션(사용자 또는 그룹)을 선택합니다.
4. **사용가능 사용자** 또는 **사용가능 그룹**에서 애플리케이션에 지정할 사용자와 그룹을 선택하고 화살표 키를 사용하여 **선택한 사용자** 또는 **선택한 그룹** 열로 이동합니다.
5. 확인을 누릅니다.

## 보안 클래스 설정

보안 클래스는 애플리케이션 요소에 대한 사용자 및 그룹 액세스 권한을 결정합니다.

다음 절차를 수행할 수 있습니다.

- [보안 클래스 생성](#)
- [보안 클래스 삭제](#)
- [보안 클래스 선택](#)

### 주:

프로비저닝 관리자 역할에 지정된 사용자만 애플리케이션의 보안 클래스를 정의할 수 있습니다.

애플리케이션의 보안 클래스를 정의한 후 계정과 엔티티 같은 애플리케이션 요소에 보안 클래스를 지정할 수 있습니다.

애플리케이션 요소에 대한 사용자나 그룹의 액세스 권한은 사용자나 그룹이 속한 보안 클래스 및 애플리케이션 요소와 연결된 보안 클래스에 따라 달라집니다.

[Default]라는 시스템 생성 보안 클래스는 애플리케이션의 일부로 생성되며 사용자가 삭제하거나 수정할 수 없습니다. [Default] 보안 클래스에 액세스 권한을 지정할 수 있습니다. 보안 클래스가 지정되지 않은 멤버는 기본값 보안 클래스가 있는 것처럼 처리됩니다.

## 보안 클래스 생성

보안 클래스를 생성하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에서 **애플리케이션 그룹**을 확장하고 애플리케이션 이름을 마우스 오른쪽 버튼으로 누른 다음 **액세스 제어 지정**을 선택합니다.
2. **보안 클래스 추가**를 선택하거나 **작업, 보안 클래스 추가** 순으로 선택합니다.
3. 이름에서 보안 클래스 이름을 입력합니다.

이름은 고유해야 하며, 최대 80자를 포함할 수 있습니다. 공백을 사용할 수 있지만 공백으로 시작할 수는 없습니다. 다음 문자는 허용되지 않습니다.

- 별표(\*)
- At 기호(@)
- 심표(,)
- 중괄호({})
- 큰따옴표""
- 빼기 기호(-)

- 숫자 기호(#)
  - 마침표(.)
  - 더하기 기호(+)
  - 세미콜론(;)
  - 슬래시 기호(/)
4. 확인을 누릅니다.

## 보안 클래스 삭제

보안 클래스가 더 이상 필요하지 않은 경우 보안 클래스 차원에서 삭제할 수 있습니다. 애플리케이션에서 보안 클래스를 삭제하기 전에 이 보안 클래스가 지정된 애플리케이션 요소에서 보안 클래스 연결을 해제해야 합니다.

엔티티, 계정 또는 시나리오의 경우 메타데이터 파일의 보안 클래스를 수정하여 보안 클래스 연결을 해제할 수 있습니다. 분개의 경우 분개 파일을 수정하거나 분개 처리 모듈에서 분개의 보안 클래스를 업데이트하여 보안 클래스 연결을 해제할 수 있습니다.

보안 클래스를 삭제하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에서 **애플리케이션 그룹**을 확장하고 애플리케이션 이름을 마우스 오른쪽 버튼으로 누른 다음 **액세스 제어 지정**을 선택합니다.
2. 보안 클래스 목록에서 테이블의 뷰를 기준으로 보안 클래스 행/열 머리글을 선택합니다.

### 💡 팁:

여러 보안 클래스를 삭제하려면 Ctrl 키를 사용하고 행/열 머리글을 선택합니다.

3. **보안 클래스 삭제**를 선택하거나 **작업, 보안 클래스 삭제** 순으로 선택합니다.
4. **예**를 눌러 삭제를 확인합니다.

## 보안 클래스 선택

기본적으로 [보안 클래스] 페이지에는 사용 가능한 보안 클래스가 표시되며, 처음 100개의 보안 클래스가 표시됩니다.

애플리케이션의 보안 클래스를 선택하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에서 **애플리케이션 그룹**을 확장하고 애플리케이션 이름을 마우스 오른쪽 버튼으로 누른 다음 **액세스 제어 지정**을 선택합니다.
2. **보안 클래스**를 선택하거나 **작업, 보안 클래스** 순으로 선택합니다.
3. **사용가능 보안 클래스**에서 애플리케이션에 지정할 보안 클래스를 선택하고 화살표 키를 사용하여 **선택한 보안 클래스** 열로 이동합니다.
4. **확인**을 누릅니다.

## 보안 클래스에 대한 사용자 액세스 권한 지정

사용자 및 그룹을 정의하고 보안 클래스를 생성한 후 애플리케이션에서 각 보안 클래스에 대한 각 사용자와 그룹의 액세스 유형을 지정할 수 있습니다.

사용자에게 5가지 액세스 유형([없음], [메타데이터], [읽기], [승인 요청] 또는 [모두]) 중 하나를 지정할 수 있습니다. 피벗 기능을 사용하여 액세스 권한 지정 테이블에 대한 두 개의 뷰를 전환할 수 있습니다. 예를 들어 행에 사용자와 그룹이 있고 열에 보안 클래스가 있는 경우 [피벗]을 누르면 사용자와 그룹은 열로 이동하고 보안 클래스는 행으로 이동합니다.

사용자에게 보안 클래스에 대한 액세스 권한을 부여할 때 전자메일 알림을 활성화할 수 있습니다. 전자메일 알림은 내부거래 보고에 사용할 수 있으며, 보안 클래스를 사용하는 시나리오 및 엔티티에 대해 프로세스 단위 상태 변경을 사용자에게 알리는 데도 사용할 수 있습니다. 프로세스 제어에 대한 전자메일 알림을 받으려면 사용자에게 보안 클래스에 대한 [모두] 또는 [승인 요청] 액세스 권한이 있어야 합니다.

### 주:

애플리케이션의 애플리케이션 관리자 역할에 지정된 사용자는 애플리케이션의 모든 정보에 액세스할 수 있습니다.

**표 3-1 사용자 액세스 레벨**

액세스 레벨	설명
없음	보안 클래스에 지정된 요소에 대한 액세스 권한이 없습니다.
메타데이터	지정된 멤버를 목록으로 볼 수 있지만 멤버의 데이터를 보거나 수정할 수는 없습니다.
읽기	보안 클래스에 지정된 요소의 데이터를 볼 수 있지만 승격시키거나 거부할 수는 없습니다.
승격	보안 클래스에 지정된 요소의 데이터를 보고 승격시키거나 거부할 수 있습니다.
모두	보안 클래스에 지정된 요소의 데이터를 수정하고 승격시키거나 거부할 수 있습니다.

보안 클래스에 사용자 액세스 권한을 지정하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에서 **애플리케이션 그룹**을 확장하고 애플리케이션 이름을 마우스 오른쪽 버튼으로 누른 다음 **액세스 제어 지정**을 선택합니다.
2. 액세스 권한을 지정할 셀을 선택합니다.

### 팁:

Shift와 Ctrl 키를 사용하면 여러 셀을 선택할 수 있습니다. 열 또는 행 머리글을 눌러 열 또는 행을 선택합니다. 열과 행의 표시를 변경하려면 **피벗**을 누릅니다.

3. 마우스 오른쪽 버튼을 누르고 지정할 액세스 레벨을 선택합니다.

 주:

테이블 1을 참조하십시오.

- 없음
  - 메타데이터
  - 읽기
  - 승격
  - 모두
4. **선택 사항:** 전자메일 알림을 추가하려면 테이블에서 셀을 선택하고 마우스 오른쪽 버튼을 누른 다음 **전자메일 알림 사용**을 선택합니다.

 주의:

알림 프로세스에서는 MSAD, LDAP 또는 Native Directory와 같은 인증 파일에 저장된 전자메일 주소를 사용합니다.

 주:

전자메일 알림을 제거하려면 셀을 선택하고 **전자메일 알림 사용 안 함**을 누릅니다.

5. **저장**을 누릅니다.

## 전자메일 알림 설정

내부거래 트랜잭션에 대해, 그리고 프로세스 관리 검토 프로세스 중에 전자메일 알림을 사용할 수 있습니다. 전자메일 알림은 시스템의 주요 이벤트 또는 데이터 변경을 강조 표시하는 데 도움이 됩니다. 예를 들어 내부거래 트랜잭션이 일치하지 않아 일치시켜야 한다거나 프로세스 단위가 다음 승인 요청 레벨에 대해 준비되었다는 전자메일 알림을 보낼 수 있습니다.

 주:

알림 프로세스에서는 LDAP, MSAD 또는 Native Directory와 같은 외부 인증 제공자에 저장된 전자메일 주소를 사용합니다.

[애플리케이션 관리] 역할이 할당된 사용자는 전자메일 알림을 자동으로 받지 않습니다. [애플리케이션 관리자] 역할이 지정된 사용자가 전자메일 알림을 받으려면 사용자를 별도의 사용자로 설정하고 알림을 받는 보안 역할을 지정합니다.

## 보안 보고서 실행

애플리케이션에 대한 보안을 설정하는 중에 선택한 정보에 대한 보안 보고서를 실행할 수 있습니다. 사용자별 클래스, 사용자별 역할, 사용자별 클래스 및 역할, 그룹별 사용자에게 대한 보고서를 실행할 수 있습니다. 보고서는 온라인으로 볼 수도 있고, 파일로 내보낼 수도 있습니다.

보안 보고서를 생성하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에서 **애플리케이션 그룹**을 확장하고 애플리케이션 이름을 마우스 오른쪽 버튼으로 누른 다음 **액세스 제어 지정**을 선택합니다.
2. **보안 보고서**를 선택하거나 **작업, 보안 보고서** 순으로 선택합니다.
3. 보고서 옵션을 선택합니다.
  - **액세스 권한**을 선택하는 경우 옵션을 선택합니다.
    - 사용자별 클래스
    - 사용자별 역할
  - **그룹별 사용자**
4. 보고서 형식: PDF, RTF, HTML, XML, XSLX를 선택합니다.



**주:**

보안 보고서에서는 HFM 형식이 지원되지 않습니다.

5. **선택 사항:** 템플릿을 선택합니다.
6. 옵션을 선택합니다.
  - **보고서 실행**은 새 페이지에서 보고서를 엽니다.
  - **파일로 익스포트**는 선택한 보고서 파일 형식으로 보고서를 저장합니다.

## 애플리케이션 보안 로드



**주의:**

Classic Oracle Hyperion Financial Management 애플리케이션의 보안 클래스만 로드할 수 있습니다.

애플리케이션에 다른 정보를 로드하려면 먼저 애플리케이션 보안을 로드해야 합니다. 여러 애플리케이션 요소를 한 번에 로드하는 경우 보안 파일이 먼저 로드됩니다.

보안 정보 로드 파일은 ASCII 또는 Unicode 형식일 수 있습니다. 보안 정보 로드 파일의 기본 파일 이름 확장자는 SEC입니다.

사용자, 보안 클래스, 역할 액세스 및 보안 클래스 액세스를 로드할 수 있습니다.

사용자나 그룹에서 역할을 제거하려면 Shared Services Console에서 역할을 수정해야 합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

 주:

웹 팝업 차단외 예외에 Financial Management를 추가하는 것이 좋습니다. 데이터 로드와 같은 일부 태스크를 수행하면 태스크 상태를 표시하는 상태 창이 팝업됩니다. 컴퓨터에 팝업 차단이 설정되어 있으면 상태 창이 표시되지 않습니다.

애플리케이션 보안을 로드하려면 다음을 수행합니다.

1. 애플리케이션을 엽니다.
2. 통합, 로드, 애플리케이션 요소 순으로 선택합니다.
3. 보안 섹션에서 로드할 파일 이름을 입력하거나 **찾아보기**를 눌러 파일을 찾습니다.

 주:

기본적으로 애플리케이션 보안 정보 파일은 SEC 파일 확장자를 사용합니다. 로드 프로세스에서는 TXT 또는 CSV와 같은 다른 파일 확장자가 허용되지만 SEC 파일 확장자를 사용하는 것이 좋습니다.

4. **선택 사항:** 새 보안 정보를 로드하기 전에 **모두 지우기**를 선택하여 애플리케이션에 대한 보안 정보를 지웁니다.

 주의:

[애플리케이션 관리자] 및 [프로비저닝 관리자] 역할이 지정된 경우에만 [모두 지우기] 옵션을 사용할 수 있습니다. 또한 이 옵션을 사용하는 경우 지우기를 수행하는 사용자를 포함하여 모든 사용자가 이 프로세스에서 제거되기 때문에 사용자를 재프로비저닝해야 합니다. 사용자 프로비저닝에 대한 자세한 내용은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오. 지우기 옵션을 선택하기 전에 [보안 정보 지우기 및 로드](#)를 검토하십시오.

5. 구분자에서 파일의 정보를 구분하는 데 사용되는 문자를 선택합니다. 적합한 문자는 다음과 같습니다.

, ~ @ \$ % ^ & | : ; ? \

 주:

파일 이름이나 파일에서 다른 모든 방식으로 사용되지 않는 문자를 사용해야 합니다. 예를 들어 엔티티 설명에 심표를 사용하는 경우에는 심표를 구분자로 사용할 수 없습니다.

6. 필터 섹션에서 로드할 보안 정보 유형을 선택합니다.

 **팁:**

필터 선택을 재설정하려면 **재설정**을 누릅니다.

7. 로드를 누릅니다.
8. **선택사항:** 로그 파일을 다운로드하려면 **로그 다운로드**를 누릅니다. 열기를 눌러 로그 파일을 표시하거나 **저장**을 누르고 로컬에서 파일을 저장할 위치를 선택합니다.

## 보안 정보 지우기 및 로드

보안 파일 로드를 시작하기 전에 애플리케이션에서 보안 정보를 지운 다음 새 보안 정보를 로드할 수 있습니다. 예를 들어 보안을 로드하는 동안 Class1 보안 클래스를 Class2로 변경하려는 경우 Class1 보안 클래스를 참조하는 모든 애플리케이션 요소를 변경해야 합니다.

시스템에서 보안 클래스 정보를 사용하는 애플리케이션 요소에 대해 새 보안 참조를 생성하기 때문에 새 보안 정보를 로드하기 전에 사전 필수 단계를 수행해야 하며, 새 보안 정보를 로드한 후 후속 단계를 수행해야 합니다.

보안 정보를 지우고 새 보안 파일을 로드하려면 다음을 수행합니다.

1. 애플리케이션에서 애플리케이션 요소를 추출합니다. **보안 정보를 지우기 전에 수행할 작업**을 참조하십시오.
2. 기존 보안 정보를 지우고 새 보안 파일을 로드하도록 선택합니다.
3. 애플리케이션에 애플리케이션 요소를 로드합니다. **보안 정보를 지운 후 수행할 작업**을 참조하십시오.

 **주:**

이러한 절차를 수행하려면 애플리케이션 관리자 보안 역할에 지정되어야 합니다.

## 보안 정보를 지우기 전에 수행할 작업

보안 정보를 지우고 보안 파일을 로드하기 전에 보안 클래스 정보를 이용하는 지정된 애플리케이션 요소에 대해 이 작업을 수행해야 합니다.

### 메타데이터

보안 정보를 지우고 로드하기 전에 메타데이터를 업데이트하려면 다음을 수행합니다.

1. 애플리케이션에서 모든 애플리케이션 메타데이터 요소를 추출합니다.
2. 필요에 따라 메타데이터 요소의 보안 클래스 정보를 변경합니다.

### 분개

보안 정보를 지우고 로드하기 전에 분개를 업데이트하려면 다음을 수행합니다.

1. 애플리케이션에서 게시된 분개의 게시를 취소합니다.
2. 분개 상태가 [작업]으로 돌아가도록 승인된 분개를 거부합니다.
3. 모든 분개를 추출합니다.



4. 필요에 따라 분개 보안 클래스 정보를 변경합니다.

#### 그리드

보안 정보를 지우고 로드하기 전에 그리드를 업데이트하려면 다음을 수행합니다.

1. 보안 클래스가 지정된 모든 그리드를 추출합니다.
2. 필요에 따라 그리드 보안 클래스 정보를 변경합니다.

#### 데이터 양식

보안 정보를 지우고 로드하기 전에 데이터 양식을 업데이트하려면 다음을 수행합니다.

1. 보안 클래스가 지정된 모든 데이터 양식을 추출합니다.
2. 필요에 따라 데이터 양식 보안 클래스 정보를 변경합니다.

## 보안 정보를 지운 후 수행할 작업

보안 정보를 지우고 보안 파일을 로드한 후 보안 클래스 정보를 이용하는 지정된 애플리케이션 요소에 대해 이 태스크를 수행해야 합니다.

#### 메타데이터

메타데이터를 업데이트하려면 다음을 수행합니다.

1. 메타데이터 정보가 지워졌는지 확인합니다.
2. 업데이트된 메타데이터 파일을 애플리케이션에 로드합니다.

#### 분개

보안 정보를 지우고 로드한 후 분개를 업데이트하려면 다음을 수행합니다.

1. 업데이트된 분개 파일을 로드합니다.
2. 보안 정보를 지우고 로드하기 전에 게시 취소한 분개를 게시합니다.
3. 보안 정보를 지우고 로드하기 전에 거부한 분개를 승인합니다.

#### 그리드

보안 정보를 지우고 로드한 후 그리드를 업데이트하려면 다음을 수행합니다.

1. 업데이트된 그리드 파일을 로드합니다.
2. 기존 문서를 덮어쓰는 옵션을 선택합니다.

#### 데이터 양식

보안 정보를 지우고 로드한 후 데이터 양식을 업데이트하려면 다음을 수행합니다.

1. 업데이트된 데이터 양식 파일을 로드합니다.
2. 기존 문서를 덮어쓰는 옵션을 선택합니다.

#### 폴더

보안 정보를 지우고 로드한 후 폴더를 업데이트하려면 다음을 수행합니다.

1. 부정확한 보안 클래스가 지정되었을 수 있는 폴더를 삭제합니다.

2. 애플리케이션에 새 폴더를 추가합니다.

#### 보고서

보안 정보를 지우고 로드한 후 보고서를 업데이트하려면 다음을 수행합니다.

1. 지정된 보안 클래스를 사용하여 모든 보고서를 재로드합니다.
2. 해당되는 경우 새 보안 클래스 지정을 제공합니다.

#### 태스크 목록

보안 정보를 지우고 로드한 후 태스크 목록을 업데이트하려면 다음을 수행합니다.

1. 지정된 보안 클래스를 사용하여 모든 태스크 목록을 재로드합니다.
2. 해당되는 경우 새 보안 클래스 지정을 제공합니다.

## 애플리케이션 보안 추출

애플리케이션 보안을 추출하여 텍스트 편집기에서 보거나 수정할 수 있습니다. 애플리케이션에서 애플리케이션 보안을 추출하는 경우 멀티바이트 문자 세트(MBCS)를 지원하는 형식으로 파일을 저장합니다. 기본적으로 애플리케이션 보안 파일은 SEC 파일 확장자를 사용합니다.

다음 유형의 보안 정보를 추출할 수 있습니다.

- 사용자 및 그룹
- 보안 클래스
- 역할 액세스
- 보안 클래스 액세스

#### 주:

주기적으로 보안을 백업 파일로 추출하는 것이 좋습니다. 보안 정보 백업에 대한 자세한 내용은 *Oracle Enterprise Performance Management System 설치 및 구성 가이드*를 참조하십시오.

애플리케이션 보안을 추출하려면 다음을 수행합니다.

1. 애플리케이션을 엽니다.
2. **통합, 추출, 애플리케이션 요소** 순으로 선택합니다.
3. **보안** 섹션의 **구분자**에서 파일의 정보를 구분하는 데 사용되는 문자를 선택합니다.

적합한 문자는 다음과 같습니다.

, ~ @ \$ % ^ & | : ; ? \

 주:

파일 이름이나 파일에서 다른 모든 방식으로 사용되지 않는 문자를 사용해야 합니다. 예를 들어 엔티티 설명에 심표를 사용하는 경우에는 심표를 구분자로 사용할 수 없습니다.

4. 필터에서 추출할 보안 유형을 선택합니다.

 팁:

선택을 재설정하려면 **재설정**을 누릅니다.

5. 추출을 누릅니다.
6. 브라우저에 표시되는 다운로드 지침에 따라 추출된 파일을 다운로드합니다.  
지침은 사용 중인 웹 브라우저에 따라 달라집니다. 설정한 웹 디렉토리에 파일을 저장해야 합니다.
7. **선택사항:** 로그 파일을 다운로드하려면 **로그 다운로드**를 누릅니다. 열기를 눌러 로그 파일을 표시하거나 **저장**을 누르고 로컬에서 파일을 저장할 위치를 선택합니다.

# 4

## 메타데이터 관리

메타데이터는 차원 이름, 멤버 이름, 등록정보, 환율, 보안 등의 데이터를 설명하고 저장하는 애플리케이션의 구조적 요소로 정의됩니다.

메타데이터를 정의하려면 XML 또는 APP 메타데이터 파일을 생성하고 애플리케이션에 로드합니다.

Oracle Hyperion Financial Management는 메타데이터에서 ASCII 문자만 사용할 수 있습니다.

 주:

메타데이터를 로드하려면 먼저 애플리케이션의 보안을 설정해야 합니다.

Financial Management용 샘플 애플리케이션을 설치하면 샘플 메타데이터 파일이 포함됩니다. 이러한 파일은 Financial Management를 설치한 디렉토리의 Sample Applications 폴더에 있습니다.

## 계정 정의

계정 차원은 애플리케이션의 계정 차트를 정의합니다. [계정 멤버 속성] 테이블의 속성으로 계정을 정의합니다.

표 4-1 계정 멤버 속성

속성	설명
AccountType	<p>(필수) 다음 값 중 하나입니다.</p> <ul style="list-style-type: none"> <li>ASSET - 회사의 자산을 나타내는 값을 저장합니다.</li> <li>LIABILITY - 회사의 부채를 나타내는 해당 시점의 잔액을 저장합니다.</li> <li>REVENUE - 주기적 값과 연 누계 값을 저장하며, 값이 양수인 경우 순가치가 증가합니다.</li> </ul> <p><b>참고:</b> 4.1 이전의 Oracle Hyperion Financial Management 릴리스에서는 이 계정 유형을 Income이라고 했습니다.</p> <ul style="list-style-type: none"> <li>EXPENSE - 주기적 값과 연 누계 값을 저장하며, 값이 양수인 경우 순가치가 감소합니다.</li> <li>FLOW - 주기적 값과 연 누계 값을 저장합니다.</li> <li>BALANCE - 특정 시점에 관련된 부호 없는 값을 저장합니다.</li> <li>BALANCERECURRING - 특정 시점에 관련이 있고 미래 기간에 재발생하는 부호 없는 값을 저장합니다.</li> <li>CURRENCYRATE - 환율 정보를 저장합니다.</li> <li>GROUPLABEL - 그룹화 용도에 계정을 사용합니다.</li> <li>DYNAMIC - 보고 있는 데이터에서 동적으로 계정 값이 계산됨을 나타냅니다.</li> </ul> <p>계정 유형 동작을 참조하십시오.</p>
CalcAttribute	<p>이 계정에 대해 규칙 파일에서 수행되는 계산 설명입니다.</p> <p>이 정보는 데이터 양식과 데이터 그리드에 셀 정보의 일부로 표시되며, 공백을 포함하여 최대 80자까지 사용할 수 있습니다.</p>
CustomTop	<p>사용자정의 차원의 계층에서 계정에 적합한 TopMember</p> <p>모든 하위 멤버를 포함하여 지정된 멤버만 계정에 적합합니다.</p> <p>속성 수는 애플리케이션에 대해 정의된 사용자정의 차원 수를 기반으로 합니다. 속성 이름이 변경되어 사용자정의 차원 별칭을 반영합니다. 예를 들어 속성 이름의 사용자정의가 사용자정의 차원 별칭으로 바뀝니다.</p>
DefaultParent 설명	<p>계정의 기본 상위</p> <p>계정 설명</p> <p>설명에는 공백을 포함하여 최대 80자까지 사용할 수 있으며 앰퍼샌드(&amp;) 또는 백슬래시(\)는 사용할 수 없습니다.</p>

표 4-1 (계속) 계정 멤버 속성

속성	설명
EnableCustomAggr	<p>현재 계정에 대해 사용자정의 차원 데이터가 합산되는지 여부</p> <p>이 속성은 가산이 아닌 특수 합계에 사용됩니다. 계정이 사용자정의 차원에 합산될 수 있으면 Y를 지정하고, 합산될 수 없으면 N을 지정합니다.</p> <p>속성 수는 애플리케이션에 대해 정의된 사용자정의 차원 수를 기반으로 합니다. 속성 이름이 변경되어 사용자정의 차원 별칭을 반영합니다. 예를 들어 속성 이름의 사용자정의가 사용자정의 차원 별칭으로 바뀝니다.</p>
EnableDataAudit	<p>계정을 감사할 수 있는지 여부</p> <p>계정 감사를 활성화하려면 Y를 지정하고, 감사를 비활성화하려면 N을 지정합니다. 기본값은 N입니다. 계정이나 시나리오에 적용할 경우 이 속성은 감사할 수 있는 대상을 결정합니다.</p>
ICPTopMember	<p>계정의 내부거래 PartnerTopMember</p> <p>지정한 멤버와 모든 하위 멤버가 계정에 적합합니다. 엔티티 차원의 다른 모든 멤버는 계정에 적합하지 않습니다.</p>
IsCalculated	<p>계정이 계산되는지 여부</p> <p>기본 레벨 계정만 계산할 수 있습니다. 기본 레벨 계정이 계산되는 경우 수동으로 값을 입력할 수 없습니다. 계정이 계산되면 Y를 지정하고, 그렇지 않으면 N을 지정합니다.</p>
IsConsolidated	<p>계정 값이 상위 엔티티에 통합되는지 여부입니다. 계정을 통합하지 않으면 통합하는 동안 무시됩니다. 통합 시 계정을 통합하려면 Y를 지정하고, 통합하지 않으려면 N을 지정합니다.</p>
IsICP	<p>계정이 내부거래 계정인지 여부를 지정합니다.</p> <ul style="list-style-type: none"> <li>• 자체 ICP 트랜잭션을 비롯한 ICP 트랜잭션이 해당 계정에 대해 허용되면 Y</li> <li>• ICP 트랜잭션이 해당 계정에 대해 허용되지 않으면 N</li> <li>• ICP 트랜잭션이 해당 계정에 대해 허용되지만 ICP 트랜잭션에서 계정 자체가 제한되면 R</li> </ul>

표 4-1 (계속) 계정 멤버 속성

속성	설명
Member	<p>계정의 이름을 지정합니다. 이 속성은 필수입니다. 이름은 고유해야 하며 공백을 포함하여 최대 80자까지 사용할 수 있지만, 공백으로 시작할 수는 없습니다.</p> <p>다음과 같은 문자는 계정 이름에 사용하지 마십시오.</p> <ul style="list-style-type: none"> <li>• 별표(*)</li> <li>• At 기호(@)</li> <li>• 쉼표(,)</li> <li>• 중괄호({})</li> <li>• 큰따옴표""</li> <li>• 보다 큼 기호(&gt;)</li> <li>• 보다 작음 기호(&lt;)</li> <li>• 빼기 기호(-)</li> <li>• 숫자 기호(#)</li> <li>• 마침표(.)</li> <li>• 더하기 기호(+)</li> <li>• 세미콜론(;)</li> <li>• 슬래시 기호(/)</li> </ul>
NumDecimalPlaces	<p>계정 값에서 소수점 오른쪽에 표시할 자릿수를 지정합니다. 이 속성은 필수입니다. 0에서 9 사이의 값을 지정합니다.</p>
PlugAcct	<p>내부거래 트랜잭션의 불일치를 식별하는 데 사용되는 계정 이름을 지정합니다. 계정에 대해 Is ICP 속성이 선택된 경우 PlugAcct 속성은 필수입니다. 공백이거나 적합한 계정의 이름이어야 합니다. 공백이면 계정에 대한 내부거래 제거가 처리되지 않습니다.</p>
SecurityClass	<p>계정 데이터에 액세스할 수 있는 사용자를 정의하는 보안 클래스 이름을 지정합니다. 보안 클래스 이름에는 최대 80자까지 입력할 수 있습니다. 보안 액세스는 계정 데이터에만 적용됩니다.</p>
Submission Group	<p>단계별 제출을 사용하는 애플리케이션에 대한 제출 그룹을 지정합니다.</p> <p>제출 그룹을 지정하려면 1에서 9 사이의 숫자를 입력하고, 프로세스 제어에서 계정을 제외하려면 0을 입력합니다. 기본값은 1입니다.</p>
UserDefined1, UserDefined2, UserDefined3	<p>계정에 대한 사용자정의 정보를 저장합니다. 최대 256자까지 입력할 수 있습니다. UserDefined1, UserDefined2 및 UserDefined3 함수는 이 속성에 저장된 텍스트를 검색합니다.</p>

표 4-1 (계속) 계정 멤버 속성

속성	설명
UsesLineItems	<p>계정에 라인 항목이 포함될 수 있는지 여부를 지정합니다. 선택하면 라인 항목이 활성화된 시나리오에서 계정에 라인 항목이 사용됩니다. 계정에 라인 항목이 사용되면 Y를 지정하고, 계정에 라인 항목이 사용되지 않으면 N을 지정합니다.</p> <p><b>주의:</b> 라인 항목 세부정보를 입력한 후 이 속성을 변경하면 저장된 라인 항목 세부정보가 해당 계정에 더 이상 적합하지 않을 수 있습니다. 다음 동작이 발생합니다.</p> <ul style="list-style-type: none"> <li>계정이 라인 항목을 승인했으나 현재 승인할 수 없는 경우 데이터베이스에 저장된 라인 항목 세부정보가 더 이상 적합하지 않습니다. 합계만 표시됩니다.</li> <li>계정이 라인 항목을 승인하지 않았으나 현재 승인할 수 있는 경우 합계 금액은 있지만 계정에 대한 해당 라인 항목 세부정보가 없습니다. 합계가 라인 항목 세부정보와 일치하도록 합계를 추출한 후 라인 항목 세부정보 데이터로 로드할 수 있습니다.</li> </ul>
XBRL Tags	<p>계정의 XBRL 태그를 지정합니다. 최대 225자까지 입력할 수 있습니다.</p>

## 계정 유형 동작

각 계정에 계정 유형이 있습니다. 계정 유형에 따라 하위 계정이 상위 계정으로 집계되는 방식과 시간이 지남에 따라 계정 잔액이 누계되는 방식이 결정됩니다. 기본 레벨 계정에 데이터를 입력하면 결과가 계층에서 자동으로 롤업됩니다.

계정 유형에 따라 하위 값을 상위 값에 더할 것인지 또는 뺄 것인지가 결정됩니다. 이 결정을 통해 계정 차트에 직접 재무 계산을 빌드할 수 있습니다.

### 주:

데이터를 로드하는 경우 Oracle Hyperion Financial Management는 데이터의 자연 기호가 양수라고 가정합니다.

예를 들어 ASSET 계정 유형은 전체 기간의 합계를 계산하지 않습니다. ASSET 계정을 차변에 기입하면 입력한 값이 해당 계정에 추가됩니다. 대변에 기입하면 값을 뺍니다. 이 계정 유형에 대한 기본 환산은 DefaultRateforBalance 계정 필드의 값입니다.

REVENUE 계정은 연 누계 합계를 제공합니다. DYNAMIC 계정 유형은 사용자정의 차원, 기간 및 기간 누계 보기에 대한 상위 값의 정확한 계산에 필요합니다. GROUPLABEL 계정 유형을 사용하여 합계로 합산될 필요가 없는 관련 계정을 그룹화할 수 있습니다. 예를 들어 대차대조표 계정을 그룹화하는 대차대조표 계정이라는 최상위 레벨 계정을 생성할 수 있습니다. GROUPLABEL을 제외한 모든 계정 유형이 데이터를 저장합니다.





주:

FLOW, BALANCE 또는 DYNAMIC 계정 유형에는 스케일링이 적용되지 않습니다.

표 4-2 계정 유형 동작

유형	YTD 합계	차변	대변	기본 환산
ASSET	아니요	추가	하위	DefaultRateForBalance 계정
LIABILITY	아니요	하위	추가	DefaultRateForBalance 계정
REVENUE	예	하위	추가	DefaultRateForFlow 계정
EXPENSE	예	추가	하위	DefaultRateForFlow 계정
FLOW	예	추가	하위	없음
BALANCE	아니요	추가	하위	없음
BALANCE RECURRING	아니요	추가	하위	없음
CURRENCYRATE	아니요	N/A	N/A	N/A
GROUPLABEL	N/A	N/A	N/A	N/A
DYNAMIC	N/A	N/A	N/A	N/A

다음 테이블에서는 특정 유형의 상위 계정에 합산되는 경우 계정 유형이 동작하는 방식을 보여줍니다. 이 열은 상위 계정의 계정 유형을 나타냅니다. 예를 들어 합산하는 경우 ASSET 계정 값이 상위 ASSET 및 EXPENSE 계정에는 추가되지만 상위 LIABILITY 및 REVENUE 계정에서는 공제됩니다.



주:

약어는 계정 유형의 처음 한 자나 두 자를 나타냅니다. 열에 No가 표시되는 경우 해당 계정 유형이 상위 계정으로 합산되지 않습니다.

표 4-3 상위 계정으로 집계되는 동안의 계정 유형 동작

유형	상위 계정									
계정 유형	A	L	R	E	F	B	BR	C	G	D
ASSET	추가	하위	하위	추가	추가	추가	추가	아니요	아니요	아니요
LIABILITY	하위	추가	추가	하위	추가	추가	추가	아니요	아니요	아니요
REVENUE	하위	추가	추가	하위	추가	추가	추가	아니요	아니요	아니요
EXPENSE	추가	하위	하위	추가	추가	추가	추가	아니요	아니요	아니요
FLOW	추가	추가	추가	추가	추가	추가	추가	아니요	아니요	아니요

표 4-3 (계속) 상위 계정으로 집계되는 동안의 계정 유형 동작

유형	상위 계정									
계정 유형	A	L	R	E	F	B	BR	C	G	D
BALANCE추가	추가	추가	추가	추가	추가	추가	추가	아니요	아니요	아니요
BALANCE추가 RECURRING	추가	추가	추가	추가	추가	추가	추가	아니요	아니요	아니요
CURRENCYRATE	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
GROUPPLANNING	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
DYNAMIC	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요

다음 예에서는 계정 유형이 상위 계정으로 합산되는 방식을 보여 줍니다.

```
·Total Assets 80
└─Fixed Assets 100
└─Amortization 20
```

이 예에서 Total Assets는 ASSET 계정이며 Fixed Assets(ASSET 계정) 및 Amortization(LIABILITY 계정)의 상위입니다. 계정이 상위 계정으로 합산되면 Fixed Assets 값 100이 추가되고 Amortization 값 20이 빠지므로 Total Assets의 결과 값은 80이 됩니다.

## 동적 계정 정의

동적 계정은 데이터가 요청될 때 동적으로 계산되는 값을 포함하는 계정입니다. 동적 계정의 값은 저장되지 않습니다. 동적 계산의 가장 일반적인 유형은 비율 계산입니다.

동적 계정과 계산을 정의하려면 다음을 수행합니다.

1. 동적 계정 유형을 사용하는 계정을 설정합니다.  
기존 계정만 동적일 수 있습니다.

 주:

IsCalculated, IsConsolidated, EnableCustomAggr, UsesLineItems 등의 계정 속성은 동적 계정에서 무시됩니다.

2. 규칙 파일에서 Sub Dynamic () 섹션을 생성합니다.
3. 규칙 파일에서 계산을 정의합니다.

계산 작성에 대한 자세한 내용은 규칙 생성 가이드라인을 참조하십시오.

## 사용자정의 멤버 정의

사용자정의 차원은 계정 차원과 연결되며 계정에 대한 추가 세부정보를 제공합니다. 사용자정의 멤버 속성 테이블의 속성을 사용하여 사용자정의 멤버를 정의합니다.

표 4-4 사용자정의 멤버 속성

속성	설명
DefaultParent	사용자정의 차원 멤버의 기본 상위를 지정합니다.
설명	사용자정의 멤버에 대한 설명을 지정합니다. 설명에는 공백을 포함하여 최대 80자까지 사용할 수 있습니다.
IsCalculated	기준 레벨 사용자정의 계정의 계산 여부를 지정합니다. 기준 레벨 사용자정의 계정이 계산되는 경우 수동으로 값을 입력할 수 없습니다. 사용자정의 계정이 계산되면 Y를 지정하고, 사용자정의 계정이 계산되지 않으면 N을 지정합니다.
Member	<p>사용자정의 멤버의 이름을 지정합니다. 이 속성은 필수입니다. 이름은 고유해야 하며 공백을 포함하여 최대 80자까지 사용할 수 있지만, 공백으로 시작할 수는 없습니다.</p> <p>사용자정의 차원 멤버의 이름은 통합 메소드의 이름과 중복될 수 없습니다.</p> <p>다음과 같은 문자는 사용자정의 멤버 이름에 사용하지 마십시오.</p> <ul style="list-style-type: none"> <li>• 별표(*)</li> <li>• At 기호(@)</li> <li>• 쉼표(,)</li> <li>• 중괄호({})</li> <li>• 큰따옴표 ""</li> <li>• 빼기 기호(-)</li> <li>• 숫자 기호(#)</li> <li>• 마침표(.)</li> <li>• 더하기 기호(+)</li> <li>• 세미콜론(;)</li> <li>• 슬래시(/)</li> </ul>
SecurityClass	사용자정의 차원 데이터에 액세스할 수 있는 사용자를 정의하는 보안 클래스 이름을 지정합니다. 보안 클래스 이름에는 최대 80자까지 입력할 수 있습니다. 보안 액세스는 데이터에만 적용됩니다.
Submission Group	<p>제출 그룹을 지정합니다. 값은 0에서 99 사이의 숫자일 수 있습니다.</p> <p>기본적으로는 값이 비어 있습니다. 비어 있는 값은 기본적으로 값 1로 설정됩니다.</p> <p>제출 그룹을 0으로 설정하면 계정이 검토 프로세스에 포함되지 않습니다.</p>

표 4-4 (계속) 사용자정의의 멤버 속성

속성	설명
SwitchSignForFlow	<p>다음 규칙을 사용하는 FLOW 계정의 기호 변경 (Debit 또는 Credit)을 지정합니다.</p> <ul style="list-style-type: none"> <li>• ASSET에서 LIABILITY</li> <li>• LIABILITY에서 ASSET</li> <li>• EXPENSE에서 REVENUE</li> <li>• REVENUE에서 EXPENSE</li> <li>• BALANCE에서 FLOW</li> <li>• FLOW에서 BALANCE</li> </ul> <p>계정의 기호가 전환되면 Y를 지정하고, 계정의 기호가 전환되지 않으면 N을 지정합니다.</p>
SwitchTypeForFlow	<p>다음 규칙을 사용하는 FLOW 계정의 계정 유형 변경을 지정합니다.</p> <ul style="list-style-type: none"> <li>• ASSET에서 EXPENSE</li> <li>• EXPENSE에서 ASSET</li> <li>• LIABILITY에서 REVENUE</li> <li>• REVENUE에서 LIABILITY</li> <li>• BALANCE에서 FLOW</li> <li>• FLOW에서 BALANCE</li> </ul> <p>계정의 계정 유형이 전환되면 Y를 지정하고, 계정의 계정 유형이 전환되지 않으면 N을 지정합니다.</p>
UserDefined1, UserDefined2, UserDefined3	<p>차원 멤버에 대한 사용자정의의 정보를 저장합니다. 최대 256자까지 입력할 수 있습니다. UserDefined1, UserDefined2 및 UserDefined3 함수는 이 속성에 저장된 텍스트를 검색합니다.</p>

## 엔티티 멤버 정의

엔티티는 사업부, 자회사, 공장, 지역, 국가, 법인, 비즈니스 단위, 부서 또는 모든 조직 구성 단위와 같은 회사의 조직 구조를 나타냅니다. 엔티티는 데이터의 통합 경로를 정의합니다. 엔티티 차원에 대해 통합을 실행하면 차원 계층에 정의된 대로 데이터가 하위 멤버에서 상위 멤버로 롤업됩니다. 엔티티 1차 하위 멤버를 여러 상위 멤버 아래에 배치하여 통합 경로를 여러 개 생성할 수 있습니다. 엔티티 멤버 속성 테이블의 속성을 사용하여 엔티티 멤버를 정의합니다.

표 4-5 엔티티 멤버 속성

속성	설명
AllowAdjFromChildren	<p>상위 엔티티에 1차 하위 구성요소의 분개 계시를 허용할지 여부를 지정합니다. 여러 상위 멤버로 롤업되는 엔티티의 경우 상위 엔티티에 대해 이 속성을 활성화할 수 있습니다. 1차 하위 구성요소의 분개 계시가 허용되면 Y를 지정하고, 1차 하위 구성요소의 분개 계시가 허용되지 않으면 N을 지정합니다.</p>

표 4-5 (계속) 엔티티 멤버 속성

속성	설명
AllowAdjs	이 엔티티에 대해 분개 계시를 허용할지 여부를 지정합니다. 엔티티에 대해 분개 계시가 허용되면 Y를 지정하고, 엔티티에 대해 분개 계시가 허용되지 않으면 N을 지정합니다.
DefaultParent	엔티티의 기본 상위를 지정합니다.
DefCurrency	엔티티의 기본 통화를 지정합니다. 이 속성은 필수입니다.
설명	엔티티에 대한 설명을 지정합니다. 설명에는 공백을 포함하여 최대 80자까지 사용할 수 있습니다.
HoldingCompany	엔티티 멤버의 소유자를 식별하는 엔티티의 지주 회사를 지정합니다. 엔티티의 이름이거나 비어 있을 수 있습니다.
IsICP	엔티티가 내부거래 트랜잭션에서 파트너일 수 있는지 여부를 지정합니다. 엔티티가 내부거래 엔티티이면 Y를 지정하고, 엔티티가 내부거래 엔티티가 아니면 N을 지정합니다. ICP를 선택한 멤버는 자동으로 ICP 차원에 멤버로 표시됩니다.
Member	<p>엔티티의 이름을 지정합니다. 이 속성은 필수입니다. 이름은 고유해야 하며 공백을 포함하여 최대 80자까지 사용할 수 있지만, 공백으로 시작할 수는 없습니다. 다음과 같은 문자는 엔티티 이름에 사용하지 마십시오.</p> <ul style="list-style-type: none"> <li>• 별표(*)</li> <li>• At 기호(@)</li> <li>• 심표(.)</li> <li>• 중괄호({ })</li> <li>• 큰따옴표 ""</li> <li>• 빼기 기호(-)</li> <li>• 숫자 기호(#)</li> <li>• 마침표(.)</li> <li>• 더하기 기호(+)</li> <li>• 세미콜론(;)</li> <li>• 슬래시(/)</li> </ul> <p>엔티티 이름으로 ALL을 사용할 수 없습니다.</p>
SecurityAsPartner	이 속성을 사용하여 내부거래 파트너 역할을 하는 엔티티에 대해 보안 클래스를 지정할 수 있습니다. ICP 엔티티에 적합한 보안 클래스의 이름을 지정합니다.
SecurityClass	엔티티 데이터에 액세스할 수 있는 사용자에게 적합한 보안 클래스의 이름을 지정합니다. 보안 클래스 이름에는 최대 80자까지 입력할 수 있습니다.
UserDefined1, UserDefined2, UserDefined3	엔티티에 대한 사용자정의 정보를 저장합니다. 최대 256자까지 입력할 수 있습니다. UserDefined1, UserDefined2 및 UserDefined3 함수는 이 속성에 저장된 텍스트를 검색합니다.

## 시나리오 멤버 정의

시나리오 차원은 예산, 실제 또는 예측과 같은 관련 데이터 세트를 나타냅니다. 시나리오 멤버 속성 테이블의 속성을 사용하여 시나리오 멤버를 정의합니다.

시나리오의 빈도는 데이터를 입력할 수 있는 기간 레벨을 지정합니다. 주기적 또는 연 누계 데이터 값을 입력하고 볼 수 있습니다. 예를 들어 데이터를 연 누계 값으로 입력하는 경우 데이터 뷰로 [주기적]을 선택하면 연 누계 값에서 주기적 값이 자동으로 파생됩니다.

각 시나리오에 대해 누락된 데이터를 표시하는 방법을 지정할 수 있습니다. Oracle Hyperion Financial Management에서는 보고서에 표시하고 요약 기간을 계산하기 위해 누락된 데이터를 0으로 해석합니다. 누락된 데이터의 0이 현재 기간(주기적)의 0으로 해석되는지 또는 YTD(연 누계)의 0으로 해석되는지를 지정할 수 있습니다.

또한 시나리오별로 프로세스 관리 옵션을 활성화합니다. 프로세스 관리를 활성화할지 여부를 선택하고, 프로세스 단위에 대한 최대 검토 레벨을 선택하고, 단계별 제출의 시작 연도를 선택할 수 있습니다.

**표 4-6 시나리오 멤버 속성**

속성	설명
ConsolidateYTD	<p>통합에 대한 데이터 뷰(연 누계 또는 주기적)를 지정합니다. 이 속성은 필수입니다. YTD를 사용하려면 Y를 지정하고, 주기적을 사용하려면 N을 지정합니다.</p> <p>ConsolidateYTD를 N으로 설정하는 경우 ZeroViewForAdj 및 ZeroViewForNonadj 옵션을 모두 Periodic으로 설정해야 합니다.</p>
DefaultFreq	<p>데이터 입력이 시나리오에 적합한 기간 유형을 지정합니다. 이 속성은 필수입니다.</p> <p>예를 들어 값으로 매월을 지정하면 월 기반 기간의 입력 데이터만 추출할 수 있고 분기 기반이나 연도 기반의 기간에서는 추출할 수 없습니다. 해당 빈도는 애플리케이션 프로파일에서 정의해야 합니다.</p>
DefaultParent	시나리오의 기본 상위를 지정합니다.
DefaultView	<p>POV 막대에서 시나리오 뷰가 선택된 경우에 사용할 데이터 뷰(연 누계 또는 주기적)를 지정합니다. 이 속성은 필수입니다. YTD 또는 [주기적]을 지정합니다.</p> <p>라인 항목 세부정보를 입력한 후 시나리오의 기본 뷰를 변경하는 경우 먼저 라인 항목 세부정보를 추출한 후 저장해야 합니다. 그런 다음 뷰를 변경하기 전에 시나리오에서 라인 항목 세부정보를 삭제합니다. 추출한 라인 항목 세부정보를 재로드하기 전에 새 기본 뷰와 일치하도록 변경해야 합니다.</p>
DefFreqForICTrans	<p>내부거래 트랜잭션 데이터의 기본 빈도를 지정합니다. 이 속성은 적합한 빈도여야 하며, 최대 80자까지 사용할 수 있습니다. 이 속성은 기본적으로 비어 있습니다.</p>
설명	시나리오에 대한 설명을 지정합니다. 설명에는 공백을 포함하여 최대 80자까지 사용할 수 있습니다.

표 4-6 (계속) 시나리오 멤버 속성

속성	설명
EnableDataAudit	<p>데이터 감사 로그에서 시나리오의 데이터 변경사항을 추적할지 여부를 지정합니다. 계정이나 시나리오에 적용할 경우 이 속성은 감사할 수 있는 항목을 결정합니다. 다음 값 중 하나를 지정합니다.</p> <ul style="list-style-type: none"> <li>모든 계정을 자동으로 감사하려면 Y를 지정합니다. EnableDataAudit가 False로 설정된 계정도 감사됩니다.</li> <li>EnableDataAudit가 True로 설정된 계정만 감사하려면 O를 지정합니다.</li> <li>모든 계정을 감사하지 않으려면 N을 지정합니다.</li> </ul>
MaximumReviewLevel	<p>시나리오의 프로세스 단위에 대한 최대 검토 레벨을 지정합니다. 각 프로세스 단위에 최대 10개 검토 레벨을 사용할 수 있습니다. 1에서 10 사이의 검토 레벨을 지정합니다. 이 속성은 필수입니다.</p>
Member	<p>시나리오의 이름을 지정합니다. 이 속성은 필수입니다. 이름은 고유해야 하며 공백을 포함하여 최대 80자까지 사용할 수 있지만, 공백으로 시작할 수는 없습니다.</p> <p>다음과 같은 문자는 시나리오 이름에 사용하지 마십시오.</p> <ul style="list-style-type: none"> <li>별표(*)</li> <li>At 기호(@)</li> <li>침표(.)</li> <li>중괄호({})</li> <li>큰따옴표""</li> <li>빼기 기호(-)</li> <li>숫자 기호(#)</li> <li>마침표(.)</li> <li>더하기 기호(+)</li> <li>세미콜론(;)</li> <li>슬래시 기호(/)</li> </ul>
PhasedSubmissionStartYear	<p>프로세스 관리 단계별 제출이 활성화된 애플리케이션에서 단계별 제출의 시작 연도를 지정합니다.</p>
SecurityClass	<p>시나리오 데이터에 액세스할 수 있는 사용자를 정의하는 적합한 보안 클래스의 이름을 지정합니다. 보안 클래스 이름에는 최대 80자까지 입력할 수 있습니다. 예를 들어 시나리오에 대한 액세스 권한이 [없음]인 사용자는 해당 시나리오에 대해 분개 기간을 열 수 있습니다.</p>
SupportsProcessManagement	<p>시나리오에서 프로세스 관리를 지원하는지 여부를 지정합니다. 다음 값 중 하나를 지정합니다.</p> <ul style="list-style-type: none"> <li>전자메일 알림 없이 프로세스 관리를 사용하려면 Y</li> <li>프로세스 관리를 사용하지 않으려면 N</li> <li>프로세스 관리 및 전자메일 알림을 사용하려면 A</li> </ul>

표 4-6 (계속) 시나리오 멤버 속성

속성	설명
UserDefined1, UserDefined2, UserDefined3	시나리오에 대한 사용자정의 정보를 저장합니다. 최대 256자까지 입력할 수 있습니다. UserDefined1, UserDefined2 및 UserDefined3 함수는 이 속성에 저장된 텍스트를 검색합니다.
UsesLineItems	<p>이 시나리오에서 계정이 라인 항목 세부정보를 사용할 수 있는지 여부를 지정합니다. 시나리오에서 라인 항목을 승인할 수 있으면 Y를 지정하고, 시나리오에서 라인 항목을 승인할 수 없으면 N을 지정합니다.</p> <p>라인 항목 세부정보를 입력한 후 이 속성을 변경하면 저장된 라인 항목 세부정보가 해당 시나리오에 더 이상 적합하지 않을 수 있습니다. 다음 동작이 발생합니다.</p> <ul style="list-style-type: none"> <li>• 시나리오가 라인 항목을 수락했으나 현재는 수락할 수 없는 경우 데이터베이스에 저장된 라인 항목 세부정보가 더 이상 적합하지 않게 됩니다. 합계만 표시됩니다.</li> <li>• 시나리오가 라인 항목을 수락하지 않았으나 현재는 수락할 수 있는 경우 합계 금액은 있지만 시나리오에 대한 해당 라인 항목 세부정보는 없습니다. 합계가 라인 항목 세부정보와 일치하도록 합계를 추출한 후 라인 항목 세부정보 데이터로 로드할 수 있습니다.</li> <li>• 초기에 시나리오에서 라인 항목 세부정보를 허용하지 않으며 시나리오에 대한 데이터가 있는 경우 시나리오 또는 시나리오의 모든 계정에 대해 라인 항목 세부정보를 사용으로 설정할 수 없습니다. 해당 시나리오에 대해 라인 항목을 사용으로 설정하려면 기존 데이터를 삭제하고 메타데이터의 라인 항목을 사용으로 설정하고 애플리케이션으로 데이터를 재로드해야 합니다.</li> </ul>
ZeroViewForAdj	<p>기간에 대해 누락된 조정된 데이터 값을 해석하는 방법을 지정합니다. 이 속성은 필수입니다. YTD 또는 [주기적]을 지정합니다.</p> <p>ConsolidateYTD를 N으로 설정하는 경우 ZeroViewForAdj 및 ZeroViewForNonadj 옵션을 모두 Periodic으로 설정해야 합니다.</p>
ZeroViewForNonadj	<p>기간에 대해 누락된 조정되지 않은 데이터 값을 해석하는 방법을 지정합니다. 이 속성은 필수입니다. YTD 또는 [주기적]을 지정합니다.</p> <p>ConsolidateYTD를 N으로 설정하는 경우 ZeroViewForAdj 및 ZeroViewForNonadj 옵션을 모두 Periodic으로 설정해야 합니다.</p>

## 애플리케이션 설정 정의

애플리케이션 설정은 전체 Oracle Hyperion Financial Management 애플리케이션에 적용됩니다. 애플리케이션 설정에 따라 애플리케이션의 다음 정보가 결정됩니다.



- 조직이 기간별 조직을 사용하여 동적입니까?
- 어떤 차원이 보호됩니까?
- 사용되는 기본 환산율과 메소드는 무엇입니까?
- ICP 가중치는 얼마입니까?
- 통합 규칙이 적용됩니까?
- 기본 통화는 무엇입니까?

애플리케이션 설정 속성 테이블의 속성을 사용하여 애플리케이션 설정을 정의합니다.

**표 4-7 애플리케이션 설정 속성**

속성	설명
ConsolidationRules	통합 규칙이 지원되는지 여부를 지정합니다. 다음 값 중 하나를 지정합니다. 사용자 정의 규칙의 Consolidate() 루틴으로 작성된 규칙을 사용하려면 Y 값 차원의 비례 값을 파생하려면 R. 비례 데이터는 저장되지 않습니다. 기본 통합 및 제거를 사용하려면 N
DefaultCurrency	애플리케이션의 기본 통화를 지정합니다. 이 속성은 필수입니다.
DefaultRateForBalanceAccounts	ASSET 또는 LIABILITY 계정에 사용할 환산율이 포함된 계정입니다. 이 속성은 필수입니다.
DefaultRateForFlowAccounts	REVENUE 또는 EXPENSE 계정에 사용할 환산율이 포함된 계정입니다. 이 속성은 필수입니다.
DefaultValueForActive	활성 계정의 기본값을 지정합니다. 이 속성은 필수입니다. 1차 하위 구성요소가 비활성으로 간주되며 상위로 통합되지 않으면 0을 지정하고, 1차 하위 구성요소가 활성으로 간주되며 상위로 통합되면 1을 지정합니다.
EnableMetadataSecurityFiltering	사용자가 모든 차원 멤버를 볼 수 있는지 또는 액세스 권한이 있는 차원 멤버만 볼 수 있는지를 지정합니다. 다음과 같은 차원 멤버가 필터링됩니다. <ul style="list-style-type: none"> <li>• 시나리오</li> <li>• 엔티티</li> <li>• 내부거래 파트너(ICP)</li> <li>• 계정</li> <li>• 사용자정의</li> </ul> 사용자에게 액세스 권한이 없는 차원 멤버를 필터링하려면 Y를 지정합니다. 이 속성의 기본값은 N입니다.
FDMAppName	Oracle Hyperion Financial Data Quality Management 애플리케이션의 이름
ICPEntitiesAggregationWeight	값 차원의 [ICP Top] 멤버에 합산되는 내부거래 파트너 엔티티 [ICP Entities] 금액의 백분율을 지정합니다. 이 속성은 필수입니다. 백분율은 백으로 스케일링되므로 100%는 1.0입니다.

표 4-7 (계속) 애플리케이션 설정 속성

속성	설명
MaxCellTextSize	셀 텍스트에 사용할 수 있는 최대 문자 수를 지정합니다. 1900 이상 최대 2,147,483,646까지의 값을 지정합니다. 0에서 1899 사이의 값은 적합하지 않습니다. 기본값은 8,000입니다.
MaxNumDocAttachments	사용자당 문서 첨부 파일의 최대 개수를 지정합니다. 제한 없음을 나타내는 -1이나 최대 2,147,483,647까지의 양수를 지정합니다. 기본값은 -1입니다.
MaxDocAttachmentSize	문서 첨부 파일 크기의 최대 바이트 수를 지정합니다. 제한 없음을 나타내는 -1이나 최대 2,147,483,646까지의 양수를 지정합니다. 기본값은 -1입니다.
NodeSecurity	노드에 대한 보안 액세스 유형을 지정합니다. 이 속성은 필수입니다. 엔티티의 보안 액세스를 기준으로 노드 데이터를 확인하려면 [엔티티]를 선택하고, 상위 멤버의 보안 액세스를 기준으로 노드 데이터를 확인하려면 [상위]를 선택합니다.
OrgByPeriodApplication	새 통합 구조가 이전 통합 구조와 함께 애플리케이션에 존재할 수 있는지 여부를 지정합니다. 새 조직 구조를 허용하려면 Y를 지정하고, 현재 조직 구조만 허용하려면 N을 지정합니다.
SupportSubmissionPhaseforAccounts	애플리케이션의 계정에 대해 프로세스 관리의 단계별 제출이 지원되는지 여부를 지정합니다. 적합한 값은 Y 또는 N입니다. 기본값은 N입니다.
SupportSubmissionPhaseforCustom	애플리케이션의 사용자정의 멤버에 대해 프로세스 관리의 단계별 제출이 지원되는지 여부를 지정합니다. 적합한 값은 Y 또는 N입니다. 기본값은 N입니다. 속성 수는 애플리케이션에 대해 정의된 사용자정의 차원 수를 기반으로 합니다. 속성 이름이 변경되어 사용자정의 차원 별칭을 반영합니다. 예를 들어 속성 이름의 사용자정의가 사용자정의 차원 별칭으로 바뀝니다.
SupportSubmissionPhaseforICP	애플리케이션의 ICP 멤버에 대해 프로세스 관리의 단계별 제출이 지원되는지 여부를 지정합니다. 적합한 값은 Y 또는 N입니다. 기본값은 N입니다.
UsePVAForBalanceAccounts	BALANCE 계정의 기본 환산 방법을 지정합니다. 주기적 값(PVA) 환산 방법을 사용하려면 Y를 지정하고, 환율 값(VAL) 환산 방법을 사용하려면 N을 지정합니다.
UsePVAForFlowAccounts	FLOW 계정의 기본 환산 방법을 지정합니다. 주기적 값(PVA) 환산 방법을 사용하려면 Y를 지정하고, 환율 값(VAL) 환산 방법을 사용하려면 N을 지정합니다.

표 4-7 (계속) 애플리케이션 설정 속성

속성	설명
UseSecurityForAccounts	애플리케이션의 계정이 보안으로 보호되는지 여부를 지정합니다. 계정에 보안이 적용되면 Y를 지정하고, 보안이 적용되지 않으면 N을 지정합니다.
UseSecurityForCustom	애플리케이션의 사용자정의 차원이 보안으로 보호되는지 여부를 지정합니다. 사용자정의 차원에 보안이 적용되면 Y를 지정하고, 사용자정의 차원에 보안이 적용되지 않으면 N을 지정합니다.  속성 수는 애플리케이션에 대해 정의된 사용자정의 차원 수를 기반으로 합니다. 속성 이름이 변경되어 사용자정의 차원 별칭을 반영합니다. 예를 들어 속성 이름의 사용자정의가 사용자정의 차원 별칭으로 바뀝니다.
UseSecurityForEntities	애플리케이션의 엔티티가 보안으로 보호되는지 여부를 지정합니다. 엔티티에 보안이 적용되면 Y를 지정하고, 엔티티에 보안이 적용되지 않으면 N을 지정합니다.
UseSecurityForICP	애플리케이션의 ICP 멤버가 보안으로 보호되는지 여부를 지정합니다. ICP 멤버에 보안이 적용되면 Y를 지정하고, ICP 멤버에 보안이 적용되지 않으면 N을 지정합니다.
UseSecurityForScenarios	시나리오가 보안으로 보호되는지 여부를 지정합니다. 시나리오에 보안이 적용되면 Y를 지정하고, 시나리오에 보안이 적용되지 않으면 N을 지정합니다.
UseSubmissionPhase	애플리케이션에서 프로세스 관리의 단계별 제출이 사용되는지 여부를 지정합니다.  적합한 값은 Y 또는 N입니다. 기본값은 N입니다.
ValidationAccount	검증에 사용할 계정 이름을 지정합니다. 검증에 사용되는 계정은 기존 계정이어야 합니다.  프로세스 관리에서 검증 계정은 프로세스 단위를 다음 검토 레벨로 승인 요청하기 전에 값이 0인지 확인하는 데 사용됩니다. 검증 계정 1은 제출 단계 1에 사용되고, 검증 계정 2-9는 제출 단계 2-9에 사용됩니다.

## 기간별 조직

조직 구조는 인수, 매각, 합병 및 조직 개편을 비롯한 다양한 이유로 변경될 수 있습니다. 기간별 조직 기능을 사용하여 최신 통합 구조와 이전 구조를 같은 애플리케이션 내에서 함께 사용할 수 있습니다.

조직 변경을 지원하기 위해 Oracle Hyperion Financial Management에서는 활성 시스템 계정을 엔티티 계층의 필터로 사용합니다. 활성 계정은 데이터를 상위 레벨에 저장하고 ICP 차원을 사용하여 1차 하위 구성요소에 대한 정보를 저장하는 내부거래 계정입니다. 상위 멤버에 대한 하위 엔티티의 통합 상태가 활성인지 또는 비활성인지를 지정합니다.

상위 멤버의 하위 멤버에 해당하는 ICP 멤버의 경우 활성 계정은 하위 멤버가 현재 연도, 시나리오 및 기간에 대한 활성 통합 멤버로 간주될지 여부를 시스템에 나타냅니다. 활성 계정이 0인 ICP 멤버에 해당하는 하위 멤버는 비활성 하위 멤버로 간주되며 통합되지 않습니다. 활성 계정이 1인 ICP 멤버에 해당하는 하위 멤버는 활성 하위 멤버로 간주되며 통합됩니다. 활성 하위 데이터의 변경사항은 상위 멤버에 영향을 주지만 비활성 하위 데이터의 변경사항은 상위 멤버에 영향을 주지 않습니다. 데이터 그리드에서 활성 계정 값을 보거나 변경할 수 있습니다.

DefaultValueForActive 속성은 활성 계정이 비어 있는 1차 하위 구성요소의 상태를 제어합니다. 따라서 모든 상위 멤버-하위 멤버 교차를 활성 또는 비활성으로 지정하지 않아도 됩니다. 기본적으로 모든 1차 하위 구성요소는 달리 지정되지 않는 한 상위에 대해 활성 상태입니다.

## 통합 메소드 정의

통합 메소드 속성 테이블에 있는 속성을 사용하여 애플리케이션의 통합 메소드를 정의합니다.

표 4-8 통합 메소드 속성

속성	설명
ConsolMethod	<p>통합 메소드의 이름을 지정합니다. 이 속성은 필수입니다. 이름은 고유해야 하며 공백을 포함하여 최대 80자까지 사용할 수 있습니다.</p> <p>다음과 같은 문자는 이름에 사용하지 마십시오.</p> <ul style="list-style-type: none"> <li>• 별표(*)</li> <li>• At 기호(@)</li> <li>• 심표(.)</li> <li>• 중괄호({})</li> <li>• 큰따옴표 ""</li> <li>• 빼기 기호(-)</li> <li>• 숫자 기호(#)</li> <li>• 마침표(.)</li> <li>• 더하기 기호(+)</li> <li>• 세미콜론(;)</li> <li>• 슬래시 기호(/)</li> </ul>
Control	<p>계산 루틴에 사용될 통제 유형에 해당하는 임계값을 지정합니다. 이 속성에 대해 다음 값 중 하나를 지정합니다.</p> <ul style="list-style-type: none"> <li>• Blank</li> <li>• 아니요</li> <li>• 제한</li> <li>• 전체</li> </ul>
설명	<p>통합 메소드에 대한 설명을 지정합니다. 설명에는 공백을 포함하여 최대 80자까지 사용할 수 있습니다.</p>
IsHoldingMethod	<p>지주 회사에 통합 메소드를 사용할지 여부를 지정합니다. 이 속성은 선택 사항입니다. 지주 회사에 통합 메소드를 사용하려면 Y를 지정하고, 지주 회사에 다른 메소드를 사용하려면 N을 지정합니다.</p>

표 4-8 (계속) 통합 메소드 속성

속성	설명
PercentConsol	<p>소유권 계산 프로세스에 의해 적용되는 연결 지분을 지정합니다. 퍼센트 값(예: 100)이나 다음 키워드 중 하나를 지정합니다.</p> <ul style="list-style-type: none"> <li>• POWN</li> <li>• POWNMIN</li> </ul> <p>POWN 및 POWNMIN에 대한 자세한 내용은 <a href="#">POWN 또는 POWNMIN 통합 메소드 사용</a>을 참조하십시오.</p>
ToPercentControl	<p>PercentControl 범위의 상한을 지정하며 소유권 계산 루틴에 사용됩니다. 0에서 100 사이의 값을 지정합니다.</p> <p>메소드 레코드 중 하나는 값이 100이어야 합니다.</p>
ToPercentControlComp	<p>통제 지분 범위의 상한이 범위에 포함되는지 여부를 지정합니다. ToPercentControl 속성과 함께 소유권 계산 루틴에 사용됩니다. UsedByCalcRoutine 속성이 N인 경우 이 속성은 선택 사항입니다. 이 속성에 대해 &lt; 또는 &lt;=을 지정합니다.</p>
UsedByCalcRoutine	<p>메소드가 자동 소유권 계산에 사용되는지 여부를 지정합니다. 소유권 계산에 메소드를 사용하려면 Y를 지정하고, 소유권 계산에 메소드를 사용하지 않으려면 N을 지정합니다.</p>

## 통합 메소드 사용

통합 메소드는 통합 및 소유권 계산 프로세스 중에 사용됩니다.

메타데이터에서 통합 메소드를 정의하면 시작 통화 차원에 대한 [ConsolMethod] 시스템 목록이 자동으로 생성됩니다. 이 목록은 통합 메소드 섹션에서 정의된 모든 메소드로 구성됩니다.

통합 중에 사용하기 위해 엔티티에 통합 메소드를 지정하는 방법에는 두 가지가 있습니다.

- 이 메소드는 데이터 로드 또는 데이터 입력을 통해 수동으로 지정할 수 있습니다.
- 엔티티에 지정된 최종적인 통제 지분을 기반으로 하는 소유권 계산 프로세스에서 메소드를 자동으로 지정할 수 있습니다. 소유권 계산에 대한 자세한 내용은 *Oracle Hyperion Financial Management 사용자 가이드*를 참조하십시오.

## 수동으로 통합 메소드 지정

통합 메소드 정보를 수동으로 입력하려면 다음 정보를 사용하여 데이터 그리드를 생성할 수 있습니다.

POV: Scenario, Year, Period, View, Entity, Value, Account, Custom

Scenario: Applicable scenario

Year: Applicable year

Period: Applicable period

Entity: A parent entity

Value: [None]

Account: [Method] system-generated account

Custom: ToCurrency

Row: ICP entities. For parent entities, you use the ICP Entities system list, or a user-defined list of selected ICP entities.

Column: FromCurrency. You should use the ConsolMethods system-generated list.

메소드 지정 정보는 상위 엔티티의 데이터 파일에 대한 계정 메소드에 저장됩니다. 상위 엔티티의 각 1차 하위 구성요소에 대해, 시스템은 ICP 차원에 통합 메소드 지정을 저장합니다. 지정된 메소드는 1차 하위 구성요소가 상위 엔티티로 통합될 때 사용됩니다.

그리드의 교차의 경우, ICP 엔티티에 메소드가 지정됨을 나타내기 위해 1을 사용합니다. 예를 들어 상위 그룹에 두 개의 1차 하위 구성요소 A와 B가 있으며 A에 Global 메소드를 지정하고 B에 Equity 메소드를 지정할 경우 Global 메소드와 엔티티 A의 교차에 1을 입력하고 Equity 메소드와 엔티티 B의 교차에 1을 입력합니다.

## POWN 또는 POWNMIN 통합 메소드 사용

[소유권 계산] 프로세스에서는 통합 메소드 테이블의 설정을 사용하여 통제 지분 및 최종 소유 지분을 계산하고, 연결 지분을 자동으로 지정하고, 통합에 대한 메소드를 지정합니다.

EQUITY 프로세스에 해당하는 통합 메소드의 경우 통합 메소드 테이블에 있는 POWNMIN 키워드를 사용합니다. POWNMIN을 사용하는 경우 EQUITY 회사에 대해 지정되는 연결 지분은 단계별로 수행되는 통합 프로세스에서 사용된 백분율과 일치합니다.

**POWNMIN 계산**

$POWNMIN = POWN + \text{Sum of (Percent Minority of Entity Owners * Direct Percentage of Ownership in the Entity)}$

설명:

- Percent Minority = 연결 지분 - 소유 지분
- Entity Owners는 현재 상위 멤버의 하위 멤버 내에서 처리되는 엔티티의 주식을 소유하는 모든 엔티티입니다.
- Direct Percentage of Ownership in the Entity는 Shares%Owned 시스템 계정에서 검색됩니다.

예:

- A가 B 소유: 80%
- A가 C 소유: 70%
- B가 D 소유: 20%
- B가 C 소유: 20%

상위 엔티티 GROUP에는 엔티티 A, B, C 및 D가 하위 멤버로 포함되어 있습니다. A는 지주 회사입니다. 소유 지분은 다음과 같이 계산됩니다.

- A: 100%
- B: 80%
- C: 70%
- D: 30%

통합 메소드 테이블에서 D의 연결 지분이 POWNMIN이라고 가정합니다.

통합 프로세스가 단계별로 수행되는 경우 POWNMIN 프로세스는 다음과 같습니다.

1. 직접 소유 지분을 사용하여 D를 B에 통합: 20%
2. 직접 소유 지분을 사용하여 D를 C에 통합: 20%
3. 해당 백분율을 사용하여 B와 C를 A에 통합: (80% 및 70%)

다음과 같이 계산됩니다.

$\text{Entity D's Percent consolidation} = 30\% + (100\% - 80\%) * 20\% + (100\% - 70\%) * 20\% = 40\%$

이 단계별 통합 프로세스를 사용할 경우 엔티티 D는 총 백분율 40%를 사용하여 통합됩니다.

하위 지주 B와 C가 A에 통합될 때 10%에 해당하는 일부 소액 주주 지분이 엔티티 D의 주식에서 계산됩니다.

**POWN 계산**

그러나 플랫폼 계층을 사용하여 통합이 수행되는 경우에는 일반적으로 프로세스에서 최종 소유 지분(POWN)을 주식 회사의 연결 지분으로 사용합니다. 이 경우 Group에 대한 D의 연결 지분은 30%가 됩니다. 소액 주주 지분은 엔티티 D의 주식에서 계산되지 않습니다.

요약하면 다음과 같습니다.

- POWN을 사용하는 경우 엔티티 D에 지정되는 연결 지분은 30%입니다(최종 소유 지분).
- POWNMIN을 사용하는 경우 엔티티 D에 지정되는 연결 지분은 40%입니다(단계별 통합 프로세스 사용).

## 통화 정의

통화는 엔티티에 대해 환산된 값을 저장합니다. 각 애플리케이션에 통화 차원이 포함되어야 합니다. 통화 차원에는 엔티티 차원의 엔티티에 지정된 각 기본 통화에 대한 통화가 포함되어야 합니다. 통화 차원에 추가된 각 통화는 값 차원에 시스템 생성 멤버로 표시됩니다. 값 차원에서 통화를 선택하여 해당 통화로 환산된 데이터 값을 볼 수 있습니다.

통화 차원 유형으로 차원을 생성하고 통화 속성 테이블에 있는 속성을 사용하여 애플리케이션의 통화를 정의합니다. 애플리케이션에서 필요한 각 통화에 대해 통화 차원에 멤버를 생성합니다.

**표 4-9** 통화 속성

속성	설명
통화	<p>통화의 이름을 지정합니다. 이 속성은 필수입니다. 이름은 고유해야 하며 공백을 포함하여 최대 80자까지 사용할 수 있습니다.</p> <p>다음과 같은 문자는 통화 이름에 사용하지 마십시오.</p> <ul style="list-style-type: none"> <li>• 별표(*)</li> <li>• At 기호(@)</li> <li>• 심표(.)</li> <li>• 중괄호({})</li> <li>• 큰따옴표 ""</li> <li>• 빼기 기호(-)</li> <li>• 숫자 기호(#)</li> <li>• 마침표(.)</li> <li>• 더하기 기호(+)</li> <li>• 세미콜론(;)</li> <li>• 슬래시 기호(/)</li> </ul>
설명	<p>통화 설명을 지정합니다. 설명에는 공백을 포함하여 최대 80자까지 사용할 수 있습니다.</p>
DisplayInICT	<p>내부거래 트랜잭션 모듈의 드롭다운 목록에 통화가 표시되는지 여부를 지정합니다. 통화를 표시하려면 Y를 지정하고, 통화를 표시하지 않으려면 N을 지정합니다. 기본값은 Y입니다.</p>



표 4-9 (계속) 통화 속성

속성	설명
Scale	<p>소수점 위치를 식별하여 통화에서 금액이 표시되고 저장되는 단위를 지정합니다. 이 속성은 필수입니다. 환율의 입력 방식도 결정합니다. 예를 들어 데이터가 천 단위로 스케일되는 경우 데이터 양식에서 입력된 값 1은 데이터베이스에 1,000으로 저장됩니다. 스케일은 엔티티 속성이 아니라 통화 속성입니다. 이 속성에 대해 다음 값 중 하나를 지정합니다.</p> <ul style="list-style-type: none"> <li>• 비어 있음 = 없음</li> <li>• 0 = 단위</li> <li>• 1 = 십</li> <li>• 2 = 백</li> <li>• 3 = 천</li> <li>• 4 = 만</li> <li>• 5 = 십만</li> <li>• 6 = 백만</li> <li>• 7 = 천만</li> <li>• 8 = 억</li> <li>• 9 = 십억</li> </ul>
TranslationOperator	<p>내부거래 트랜잭션을 사용하는 경우 로컬 통화로 환율을 곱할 것인지 또는 나눌 것인지를 지정합니다. 기본적으로는 값이 비어 있습니다. 다음 값 중 하나를 지정합니다.</p> <ul style="list-style-type: none"> <li>• 트랜잭션 통화를 비율로 나누어 로컬 통화를 계산하려면 D</li> <li>• 트랜잭션 통화로 비율을 곱하여 로컬 통화를 나타내려면 M</li> <li>• 기본값인 D로 설정하려면 공백</li> </ul>

[Currencies] 시스템 멤버 목록은 시작 통화 및 대상 통화 차원에 사용할 수 있습니다. 애플리케이션에 추가한 통화는 [Currencies] 멤버 목록에 추가됩니다. [Currencies] 목록을 사용하면 통화 쌍에 대한 통화 환산율을 입력하고 비통화 멤버를 필터링할 수 있습니다.

## 셀 텍스트 레이블 정의

데이터 그리드 또는 양식에서 적합한 모든 셀에 대해 셀 텍스트를 추가할 수 있습니다. 경우에 따라 POV 교차에 대한 여러 유형의 정보를 저장하기 위해 셀 텍스트 항목이 여러 개 필요할 수도 있습니다. 셀 텍스트 항목을 여러 개 생성하고 셀 텍스트 레이블을 정의하면 정보 유형을 쉽게 확인할 수 있습니다.

지정된 POV 및 셀 텍스트 레이블의 셀 텍스트에 텍스트 문자열을 씁니다. 레이블은 메타데이터로 로드되며, 애플리케이션의 모든 계정 셀에 적용되며 사용자가 셀 텍스트 정보를 입력할 때 선택할 수 있습니다. *Oracle Hyperion Financial Management 사용자 가이드*를 참조하십시오.

셀 텍스트 레이블 정의에 대한 지침은 다음과 같습니다.

- 레이블 이름에는 최대 80자를 사용할 수 있습니다. 레이블 이름은 데이터 셀 내에서 고유해야 합니다.

- 레이블 이름에 공백을 사용할 수 있지만 공백으로 시작할 수는 없습니다. Oracle 데이터베이스를 사용하는 경우에는 레이블에 공백을 사용할 수 없습니다.
- 레이블 이름에 다음 문자를 사용할 수는 없습니다.
  - 앰퍼샌드(&)
  - 별표(\*)
  - At 기호(@)
  - 쉼표(,)
  - 중괄호({})
  - 큰따옴표""
  - 슬래시(/)
  - 보다 작음 기호(<)
  - 빼기 기호(-)
  - 숫자 기호(#)
  - 마침표(.)
  - 파이프 문자(|)
  - 더하기 기호(+)
  - 세미콜론(;)
  - 물결표 문자(~)

로드 파일의 [셀 텍스트 레이블] 섹션에 셀 텍스트 레이블을 메타데이터 로드的一部分로 로드할 수 있습니다. 다음 예에서는 셀 텍스트 레이블의 샘플 로드 파일 섹션을 보여 줍니다.

```
<MISC Name="CellTextLabel">
```

```
<MISCENTRY>
```

```
<LABEL>MaturityDate</LABEL>
```

```
</MISCENTRY>
```

```
<MISCENTRY>
```

```
<LABEL>CouponRate</LABEL>
```

```
</MISCENTRY>
```

```
<MISCENTRY>
```

```
<LABEL>ExchangeRate</LABEL>
```

```
</MISCENTRY>
```

```
<MISCENTRY>
```

```
<LABEL>Rating</LABEL>
```

```
</MISCENTRY>
```

```
<MISCENTRY>
```

```
<LABEL>InterestRate</LABEL>
```

```
</MISCENTRY>
```

```
</MISC>
```

## 시스템 생성 계정

애플리케이션을 생성하면 애플리케이션에 대한 통합 및 소유권에 사용되는 시스템 계정이 자동으로 생성됩니다.

### 주:

시스템 계정에 대한 설명, 보안 클래스 및 소수점 위치만 변경할 수 있으며 시스템 계정의 다른 모든 속성은 사전 정의된 것이므로 수정할 수 없습니다.

## 통합 계정

다음 통합 시스템 계정은 엔티티 차원의 각 상위에 필요하며 통합 프로세스에서 사용됩니다.

### 주:

활성 계정을 제외하고 통합에 사용되는 모든 시스템 계정은 BALANCE 계정입니다. 활성 계정은 BALANCERECURRING 계정입니다.

표 4-10 통합에 사용되는 시스템 계정

계정	설명
Active	1차 하위 구성요소의 상태가 상위 멤버로 통합되는 것을 말합니다. 하위 멤버가 해당 상위 멤버에 통합되면 Y이고, 하위 멤버가 해당 상위 멤버에 통합되지 않으면 N입니다.
[PCON]	연결 지분을 나타냅니다. 엔티티의 상위 멤버로 통합되는 엔티티 값의 백분율로, -100에서 100 사이(0 포함)의 양수 또는 음수여야 하며 기본값은 100입니다. 후속 기간의 경우 0이 지정되므로 모든 후속 기간에 백분율을 입력해야 합니다.
[POWN]	다른 엔티티가 소유하는 엔티티의 주식을 기반으로 하는 소유 지분으로, 0-100 사이의 양수여야 하며 기본값은 100입니다.
[DOWN]	직접 소유권의 백분율로, 0-100 사이의 양수여야 하며 기본값은 100입니다.
[PCTRL]	다른 엔티티가 소유하는 의결권부 주식을 기반으로 하는 통제 지분으로, 0-100 사이의 양수여야 하며 기본값은 100입니다.
Method	엔티티에 지정된 통합 메소드입니다. 없음 또는 사용 가능한 메소드 목록에서 선택합니다.
Consol1, Consol2, Consol3	통합 메소드로, 0-255 사이의 숫자입니다.

## 소유권 계정

이러한 소유권 시스템 계정은 소유권 계산에 사용됩니다.



**주:**

소유권 계산에 사용되는 모든 시스템 계정은 BALANCE 계정입니다.

**표 4-11** 소유권에 대한 시스템 계정

계정	설명
SharesOwned	총 소유 주식 수로, 양수 또는 0이어야 하며 기본값은 0입니다. 총 소유 주식 수는 기발행 총 주식 수보다 작거나 같아야 합니다.
VotingOwned	소유 의결권부 주식 수로, 양수 또는 0이어야 하며 기본값은 0입니다. 총 소유 의결권부 주식 수는 기발행 총 의결권부 주식 수보다 작거나 같아야 합니다.
SharesOutstanding	기발행 총 주식 수 또는 기발행 주식 백분율로, 양수 또는 0이어야 하며 기본값은 0입니다. 기발행 주식 수를 입력하거나 기발행 주식 수를 백분율로 입력합니다. 퍼센트에 대해 100을 입력합니다.
VotingOutstanding	기발행 의결권부 주식 수로, 양수 또는 0이어야 하며 기본값은 0입니다. 기발행 의결권부 주식 수를 입력하거나 기발행 의결권부 주식을 백분율로 입력합니다. 퍼센트에 대해 100을 입력합니다.
Shares%Owned	시스템에서 계산됩니다.
Voting%Owned	시스템에서 계산됩니다.

## 시스템 생성 계정 편집

애플리케이션을 생성하면 애플리케이션에 대한 시스템 계정 멤버가 자동으로 생성됩니다.



**주:**

시스템 계정 멤버에 대한 설명, 보안 클래스 및 소수점 위치만 편집할 수 있습니다. 다른 모든 속성은 사전 정의된 것이므로 수정할 수 없습니다.

시스템 계정을 편집하려면 다음을 수행합니다.

1. 시스템 생성 계정 멤버가 포함된 메타데이터 파일을 엽니다.

2. 목록 뷰 탭을 선택합니다.
3. 목록에서 시스템 계정을 선택하고 필요에 따라 설명, 보안 클래스 및 소수점 위치를 수정합니다.
4. 필요한 경우 3단계를 반복하여 다른 시스템 계정 멤버를 수정합니다.
5. 파일 저장을 누릅니다.



주:

변경사항을 적용하려면 업데이트된 메타데이터 파일을 애플리케이션에 로드해야 합니다.

## 내부거래 파트너 설정

내부거래 트랜잭션은 내부거래 파트너(ICP) 차원 내에서 관리됩니다. ICP 차원에는 계정에 대해 존재하는 모든 내부거래 잔액이 포함됩니다. ICP는 계정 차원 및 사용자정의 차원과 함께 사용되어 내부거래 트랜잭션 세부정보를 추적 및 제거하는 예약된 차원입니다.

내부거래 트랜잭션에 대해 애플리케이션을 설정하려면 다음 작업을 수행해야 합니다.

- 내부거래 트랜잭션을 수행하는 계정을 나타내며 각 내부거래 계정에 대한 플러그 계정을 나타냅니다(계정 메타데이터의 IsICP 및 PlugAcct 속성).
- 내부거래 트랜잭션을 수행하는 엔티티를 나타냅니다(엔티티 메타데이터의 IsICP 속성).

내부거래 트랜잭션을 생성할 때 각 그룹에 적어도 하나의 내부거래 계정 및 하나의 플러그 계정이 있어야 합니다. 계정의 IsICP 속성을 선택하여 계정을 내부거래 계정으로 지정합니다. 계정을 내부거래 계정으로 지정하고 내부거래 트랜잭션을 입력하면 통합 프로세스를 통해 [Elimination] 값 차원 멤버에 제거 또는 전환 항목이 생성됩니다.

플러그 계정은 제거가 완료될 때 Elimination 값 차원에 두 내부거래 계정 간의 차이를 저장하는 계정입니다. 플러그 계정은 ICP 계정으로 설정될 수 있습니다. 플러그 계정이 ICP에 의해 구체화되도록 하려면 IsICP 메타데이터 속성을 Y 또는 R로 설정하여 시스템이 해당 ICP 멤버에 제거 내용을 쓰도록 합니다. 플러그 계정이 ICP에 의해 구체화되지 않도록 하려면 IsICP 속성을 N으로 설정하여 시스템이 [ICP None]에 제거 내용을 쓰도록 합니다.

통합하는 동안 적합한 내부거래 엔티티 간의 트랜잭션이 제거됩니다. [엔티티 멤버 정의를](#) 참조하십시오.

다음 표에서는 시스템 생성 ICP 요소를 보여 줍니다.

**표 4-12 시스템 생성 ICP 요소**

ICP 요소	설명
[ICP Top]	최상위 내부거래 멤버를 지정합니다.
[ICP None]	내부거래 멤버가 사용되지 않음을 지정합니다.
[ICP Entities]	내부거래 트랜잭션에 대해 지정된 엔티티를 지정합니다.

## 시스템 생성 ICP 멤버 편집

애플리케이션을 생성하면 애플리케이션에 대한 ICP(내부거래 파트너) 멤버가 자동으로 생성됩니다. IsICP 속성이 선택된 각 엔티티 멤버에 대해 ICP 멤버가 생성됩니다.

### 주:

ICP 멤버에 대한 설명과 보안 클래스만 수정할 수 있으며 다른 모든 속성은 사전 정의된 것이므로 수정할 수 없습니다.

내부거래 멤버를 수정하려면 다음을 수행합니다.

1. 시스템 생성 내부거래 파트너 멤버가 포함된 메타데이터 파일을 엽니다.
2. 목록에서 ICP 멤버를 선택한 다음 필요에 따라 설명과 보안 클래스를 수정합니다.
3. 필요한 경우 2단계를 반복하여 다른 ICP 멤버를 수정합니다.
4. **파일 저장**을 누릅니다.

### 주:

변경사항을 적용하려면 업데이트된 메타데이터 파일을 애플리케이션에 로드해야 합니다.

## 시스템 생성 값 멤버 편집

애플리케이션을 생성할 때는 애플리케이션에 대해 값 멤버가 자동으로 생성됩니다.

### 주:

값 멤버에 대한 설명만 수정할 수 있으며 다른 모든 속성은 사전 정의된 것이므로 수정할 수 없습니다.

메타데이터를 로드하면 애플리케이션에서 각 통화에 대해 다음 세 가지 값 차원 멤버가 자동으로 생성됩니다.

- *CurrencyName*
- *CurrencyName Adjs*
- *CurrencyName Total*

여기서 *CurrencyName*은 통화 레이블입니다.

예를 들어 USD 통화의 경우 시스템에서 값 차원 멤버 USD, USD Adjs 및 USD Total을 생성합니다.

**주:**

메타데이터 파일에는 <Currency> 값 멤버에 대한 설명이 지정되어 있어야 합니다. 통화 설명이 메타데이터 파일에 지정되어 있지 않으면 메타데이터를 로드할 때 통화 설명이 표시되지 않습니다.

값 멤버 설명을 수정하려면 다음을 수행합니다.

1. 시스템 생성 값 멤버가 포함된 메타데이터 파일을 엽니다.
2. **멤버 속성** 탭의 계층에서 멤버를 선택하고 해당 설명 속성을 수정합니다.
3. 필요한 경우 2단계를 반복하여 다른 값 멤버에 대한 설명을 추가합니다.
4. **파일 저장**을 누릅니다.

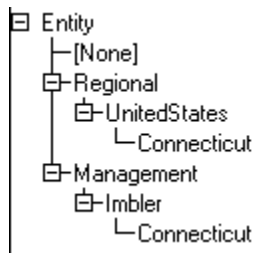
**주:**

변경사항을 적용하려면 업데이트된 메타데이터 파일을 애플리케이션에 로드해야 합니다.

## 보안을 기반으로 하는 메타데이터 필터링

보안을 기반으로 메타데이터를 필터링하면 사용자에게 액세스 권한이 있는 시나리오, 엔티티, ICP, 계정, 사용자정의 차원의 요소만 표시됩니다. `AppSettings EnableMetadataSecurityFiltering` 메타데이터 속성을 Y로 설정하여 애플리케이션 레벨에서 메타데이터 필터링을 설정합니다. 사용자가 계층에서 볼 수 있는 요소의 경우, 보안 클래스를 지정하고 사용자에게 보안 클래스에 대한 메타데이터 액세스 권한을 지정합니다.

사용자는 액세스 권한이 있는 멤버의 상위 및 상위 멤버에 대해 암시적인 액세스 권한을 갖습니다. 암시적 액세스 권한이 있으면 계층 트리 구조의 상위 및 상위 멤버를 볼 수 있지만 액세스할 수는 없습니다. 예를 들어 다음 트리 구조에서 상위(UnitedStates 및 Imbler) 및 상위 멤버(Management 및 Regional)가 트리에 표시되더라도 Connecticut에만 액세스할 수 있습니다.



## APP 형식의 메타데이터 파일 생성

APP 형식 메타데이터 파일을 사용하여 애플리케이션에 메타데이터를 추가할 수 있습니다. 메타데이터 파일 섹션은 임의의 순서로 정렬할 수 있지만 시스템에서 자동으로 섹션이 다음 순서로 처리됩니다.



- 통화
- 엔티티 차원
- 시나리오 차원
- 사용자정의 차원
- 계정 차원
- 값 차원
- 내부거래 파트너 차원
- 애플리케이션 설정
- 통합 메소드

각 차원 관련 메타데이터 유형에 대해서는 섹션이 다음 순서로 처리됩니다.

- 멤버
- 계층

 주:

사용자정의 차원의 경우 이 섹션에서 사용자정의 차원 별칭 이름을 사용해야 합니다.

엔티티, 시나리오, 계정 및 사용자정의 차원의 메타데이터는 멤버 및 계층 섹션에 배치됩니다. 사용자정의 차원에는 차원 섹션이 포함될 수 있습니다. 각 메타데이터 유형에 대한 섹션은 메타데이터 파일에 한 번만 포함될 수 있습니다.

다음과 같은 문자를 구분자로 사용할 수 있습니다.

, ~ @ \$ % ^ | : ; ? \

 주:

파일 이름이나 파일에서 다른 모든 방식으로 사용되지 않는 문자를 사용해야 합니다. 구분자는 파일 확장자가 APP인 ASCII 파일에만 필요합니다. XML(Extensible Markup Language) 파일에는 구분자가 필요하지 않습니다.

느낌표(!)로 시작하는 라인은 메타데이터 파일에서 새 섹션의 시작을 나타내며 뒤에 적합한 섹션 이름(예: 통화, 멤버 또는 계층)이 와야 합니다. True 또는 False 값은 True는 Y, False는 N으로 표시됩니다. 아포스트로피(')로 시작하는 라인은 주석 라인으로 간주되며 시스템에서 무시됩니다.

메타데이터 파일에서 다음과 같은 섹션을 사용할 수 있습니다.

- 파일 형식
- 버전
- 애플리케이션 설정
- 통화

- 차원
- 멤버
- 계층
- 통합 메소드

## 파일 형식

메타데이터 파일의 이 섹션에서는 파일 버전 번호를 나타냅니다. 버전 번호는 파일 형식을 변경하는 경우에만 변경됩니다. 파일 형식은 메타데이터를 추출할 때 자동으로 생성됩니다. 로드할 파일을 정의하는 경우 적합한 파일 형식을 포함해야 합니다. 다음 구문을 사용하여 파일 형식을 지정합니다.

```
!FILE_FORMAT = majorNumber.minorNumber
```

*majorNumber* 및 *minorNumber*는 한두 자리 숫자로 구성됩니다. *majorNumber*는 선행 0을 포함할 수 있고, *minorNumber*는 후행 0을 포함할 수 있습니다. *majorNumber*와 *minorNumber*를 포함하고 소수 구분자로 마침표( . )만 사용해야 합니다. 다음 예는 적합한 파일 형식 값을 나타냅니다.

```
!FILE_FORMAT = 11.12
```

```
!FILE_FORMAT = 11.120
```

```
!FILE_FORMAT = 011.120
```

```
!FILE_FORMAT = 011.12
```

## 버전

메타데이터 파일의 이 섹션에는 메타데이터를 추출하는 데 사용된 Oracle Hyperion Financial Management 버전이 표시됩니다. 버전 번호는 메타데이터를 추출할 때 자동으로 생성됩니다. 로드할 메타데이터 파일을 생성하는 경우 버전을 지정할 필요가 없습니다. 버전은 다음 구문으로 표현됩니다.

```
!VERSION = major version.minor version.build version
```

다음 예에서는 적합한 버전 값을 나타냅니다.

```
!VERSION = 11.1.4749
```

## 애플리케이션 설정

메타데이터 파일의 이 섹션에서는 전체 Oracle Hyperion Financial Management 애플리케이션에 적용되는 설정을 정의합니다. 애플리케이션 설정 속성에 대한 자세한 내용은 [애플리케이션 설정 정의](#)를 참조하십시오.

다음 예에서는 애플리케이션 설정 속성을 지정합니다.

```
!APPLICATION_SETTINGS
```

```
DefaultCurrency=USD
```

```
DefaultRateForBalanceAccounts=Rate1
```

```
DefaultRateForFlowAccounts=Rate2
```

```
UsePVAForBalanceAccounts=Y
```

```
UsePVAForFlowAccounts=Y
```

```
ICPEntitiesAggregationWeight=1
```

```
DefaultValueForActive=1
```

```
ConsolidationRules=N
```

```
OrgByPeriodApplication=N
```

```
NodeSecurity=Entity
```

```
UseSecurityForAccounts=N
```

```
UseSecurityForEntities=Y
```

```
UseSecurityForScenarios=Y
```

```
UseSecurityForFlows=Y
```

```
UseSecurityForMarket=Y
```

```
UseSecurityForRegion=N
```

```
UseSecurityForCostCenter=N
```

```
UseSecurityForICP=N
```

## 통화

메타데이터 파일의 이 섹션에서는 통화를 정의합니다. 다음 구문을 사용하여 통화를 지정합니다.

```
Label; Scale; Descriptions
```

[통화 정의](#)를 참조하십시오.

다음 예에서는 통화 속성을 지정합니다.

```
!CURRENCIES

EURO;0;English=European Euro

GBR;0;English=Great Britain Pounds

USD;0;English=United Stated Dollars
```

## 멤버

메타데이터 파일의 이 섹션에서는 차원의 멤버를 정의합니다. 구분자를 사용하여 누락된 값을 비어 있음으로 표시할 수 있습니다. 다음 구문을 사용하여 차원 멤버를 입력합니다.

```
!MEMBERS=Customers

'Label;IsCalculated;SwitchSignForFlow;SwitchTypeForFlow;
UserDefined1;UserDefined2;UserDefined3;SecurityClass;
DefaultParent;Descriptions

[None];N;N;N; ; ; ; ;DefaultParent=#root

AllCustomers;Y;N;N; ; ; ; ;DefaultParent=#root

Customer2;N;N;N; ; ; ; ;DefaultParent=AllCustomers

Customer3;N;N;N; ; ; ; ;DefaultParent=AllCustomers

Customer4;N;N;N; ; ; ; ;DefaultParent=AllCustomers

Customer5;N;N;N; ; ; ; ;DefaultParent=AllCustomers
```

**주:**

사용자정의 차원의 경우 이 섹션에서 사용자정의 차원 별칭 이름을 사용해야 합니다.

이러한 항목은 계정, 시나리오, 엔티티, 사용자정의, 값 및 ICP 차원의 멤버 섹션 형식을 보여 줍니다.

## 계정

계정 차원 멤버에 대한 구문:

```
'Label, AccountType, IsCalculated, IsConsolidated, IsICP, PlugAcct,
CustomTop, NumDecimalPlaces, UsesLineItems, EnableCustomAggr,
UserDefined1, UserDefined2, UserDefined3, XBRLTags, SecurityClass,
ICPTopMember, EnableDataAudit, DefaultParent, Descriptions
```

[계정 정의](#)를 참조하십시오.

`Custom_Order` 키워드는 모든 새 메타데이터 파일에서 사용자정의 차원의 열을 정의하는 데 필요합니다. 이 섹션은 애플리케이션 프로파일(.per) 파일의 사용자정의 순서와 일치해야 합니다.

예:

```
!Custom_Order=Product;Customers;Channel;UnitsFlows
```

다음 예에서는 두 계정 `AdminExpenses`와 `CapitalStock`의 속성을 지정합니다.

```
!MEMBERS=Account
```

```
AdminExpenses;EXPENSE;N;Y;Y; ;AllCustom3;AllCustom1;AllMarket;AllFlows;2;N;Y;Y;Y;Y; ;
; ; ; ; ;N;DefaultParent=NetIncome
```

```
CapitalStock;LIABILITY;N;Y;N; AllCustom3;
[None];AllMarket;AllFlows;6;N;Y;Y;Y;Y;
; ; ; ; ; ;N;DefaultParent=TotalEquity;English=Capital Stock
```

## 시나리오

시나리오 차원 멤버에 대한 구문:

```
'Label, DefaultFreq, DefaultView, ZeroViewForNonadj, ZeroViewForAdj,
ConsolidateYTD, UserDefined1, UserDefined2, UserDefined3,
SupportsProcessManagement, SecurityClass, MaximumReviewLevel,
UsesLineItems, EnableDataAudit, EnableJournalsAutoLabel,
DefFreqForPostingFlowTrans, DefaultParent, Descriptions
```

시나리오 멤버 정의를 참조하십시오.

다음 예에서는 두 시나리오 실제와 예산의 속성을 지정합니다.

```
!MEMBERS=Scenario
```

```
Actual;MTD;Periodic;Periodic;Periodic;N; ; ; ;N; ;10;Y;N;N;MTD ;DefaultParent=#root
```

```
Budget;MTD;Periodic;Periodic;Periodic;N; ; ; ;Y; ;10;Y;N;N;MTD ;DefaultParent=#root
```

## 엔티티

엔티티 차원 멤버에 대한 구문:

```
'Label, DefCurrency, AllowAdjs, IsICP, AllowAdjFromChildren, SecurityClass,
UserDefined1, UserDefined2, UserDefined3, HoldingCompany, SecurityAsPartner,
DefaultParent, Descriptions
```

엔티티 멤버 정의를 참조하십시오.

다음 예에서는 세 개 엔티티 California, Canada 및 Connecticut의 속성을 지정합니다.

```
!MEMBERS=Entity
```

```
California;USD;Y;Y;Y;US;;;;;DefaultParent=Imbler;English=State of
California;French=California
```

```
Canada;USD;Y;N;N;;;;;DefaultParent=Regional
```

```
Connecticut;USD;Y;Y;N;US;Northeast;;;;;DefaultParent=Imbler
```

## 사용자정의

사용자정의 차원 멤버에 대한 구문:

```
'Label, IsCalculated, SwitchSignForFlow, SwitchTypeForFlow, UserDefined1,
UserDefined2, UserDefined3, SecurityClass, DefaultParent, Descriptions
```

사용자정의 멤버 정의를 참조하십시오.



예를 들어 통화 USD 및 EUR을 각각 "US Dollars" 및 "Euro"라는 설명과 함께 정의한다고 가정합니다. 또한 로드된 메타데이터 파일에서 다음과 같은 값 멤버 설명을 정의한다고 가정합니다.

```
[None];English=ValueNone
```

```
<Currency Total>;English=Total
```

```
<Currency Adjs>;English=Adjs
```

```
<Currency>;English=Base
```

다음 표에서는 USD 및 유로 통화용으로 시스템에서 생성하는 값 차원 멤버 3쌍에 대해 설명합니다.

**표 4-13** 값 차원 설명

값 멤버	설명
USD Total	US Dollars Total
USD Adjs	US Dollars Adjs
USD	US Dollars Base
EUR Total	Euro Total
EUR Adjs	Euro Adjs
EUR	Euro Base

 **주:**

메타데이터 파일에는 값 멤버 <Currency>에 대한 설명이 지정되어 있어야 합니다. 통화 설명이 메타데이터 파일에 지정되어 있지 않으면 메타데이터를 로드할 때 통화 설명이 표시되지 않습니다.

## 내부거래 파트너

멤버 섹션을 사용하여 내부거래 파트너 차원의 시스템 생성 멤버에 대한 보안 클래스와 설명을 정의할 수 있습니다.

- [ICP Top]
- [ICP None]
- [ICP Entities]

내부거래 파트너 멤버에 대한 구문:

```
Label;SecurityClass;Descriptions
```



다음 예에서는 보안 클래스를 지정하지 않고 [ICP Top], [ICP None] 및 [ICP Entities]에 대한 설명을 정의하는 방법을 보여 줍니다.

```
[ICP Top];;English=Top ICP
```

```
[ICP None];;English=No ICP
```

```
[ICP Entities];;English=Entities ICP
```

## 통합 메소드

메타데이터 파일의 이 섹션에서는 통합 메소드를 정의합니다.

통합 메소드에 대한 구문:

```
Label;UsedByCalcRoutine;IsHoldingMethod;ToPercentControlComp;  
ToPercentControl;PercentConsol;Control;Descriptions
```

[통합 메소드 정의를 참조하십시오.](#)

다음 예에서는 통합 메소드의 속성을 지정합니다.

```
!CONSOLIDATION_METHODS
```

```
M2;Y;N;<=;20;0;No
```

```
M3;Y;N;<;50;POWN;Limited
```

```
M4;Y;N;<=;50;50;Limited
```

```
M1;Y;Y;<=;100;100;Full
```

```
M5;Y;N;<=;100;100;Full
```

## 계층

이 메타데이터 파일 섹션에서는 상위-1차 하위 구성요소 관계를 정의합니다. 상위-1차 하위 구성요소 관계를 노드라고 합니다. 노드에는 고유한 속성 값 세트를 지정할 수 있습니다.

노드 레코드는 구분된 목록입니다. 목록에서 각 라인의 처음 두 항목은 상위와 1차 하위 구성요소를 식별합니다. 구분자를 사용하여 누락된 속성 값을 비어 있음으로 표시할 수 있습니다. 계층의 모든 최상위 레벨 멤버는 빈 상위의 1차 하위 구성요소로 표시해야 합니다.

 **팁:**

사용자정의 차원의 노드 레코드에는 세 번째 속성이 있습니다. [사용자정의 계층](#)을 참조하십시오.

계층 섹션을 시작하려면 다음 라인을 입력하고 <> 문자를 차원 이름으로 바꿉니다.

```
!HIERARCHIES=<>
```

사용자정의 차원에 대한 섹션을 시작할 때는 공백을 포함하지 마십시오. 예를 들어 FLOW 차원의 계층 섹션은 다음 라인으로 시작합니다.

```
!HIERARCHIES=Customers
```

```
;[None];1
```

```
;AllCustomers;0
```

```
AllCustomers;Customer2;1
```

```
AllCustomers;Customer3;1
```

```
AllCustomers;Customer4;1
```

```
AllCustomers;Customer5;1
```

 **주:**

사용자정의 차원의 경우 이 섹션에서 사용자정의 차원 별칭 이름을 사용해야 합니다.

이러한 항목은 계정, 시나리오, 엔티티 및 사용자정의 차원의 계층 섹션 형식을 보여 줍니다.

## 계정 계층

계정 차원 계층에 대한 구문:

```
parentmemberlabel;childmemberlabel
```

다음 예에서는 계정 차원 계층을 지정합니다.

```
!HIERARCHIES=Account

; [None]

;ExchangeRates

ExchangeRates;Rate1

ExchangeRates;Rate2

;Plug

;NetProfit

NetProfit;NetIncome

NetIncome;GrossMargin

GrossMargin;Sales

GrossMargin;TotalCosts

TotalCosts;Purchases

TotalCosts;Salaries

TotalCosts;OtherCosts

NetIncome;AdminExpenses

NetIncome;InterestCharges

NetProfit;Taxes
```

## 시나리오 계층

시나리오 차원 계층에 대한 구문:

```
parentmemberlabel;childmemberlabel
```

다음 예에서는 시나리오 차원 계층을 지정합니다.

```
!HIERARCHIES=Scenario
```

```
;Actual
```

```
;Budget
```

## 엔티티 계층

엔티티 차원 계층에 대한 구문:

```
parentmemberlabel;childmemberlabel
```

다음 예에서는 엔티티 차원 계층을 지정합니다.

```
!HIERARCHIES=Entity
```

```
:[None]
```

```
;Regional
```

```
Regional;UnitedStates
```

```
UnitedStates;California
```

```
California;Sunnyvale
```

```
California;FosterCity
```

## 사용자정의 계층

사용자정의 차원 계층에 대한 구문:

```
parentmemberlabel;childmemberlabel;AggregationWeight
```

다음 예에서는 사용자정의 차원 계층을 지정합니다.

```
!HIERARCHIES=Products
```

```
; [None]; 1
```

```
; AllProducts; 0
```

```
AllProducts; Golf; 1
```

```
Golf; GolfBalls; 1
```

```
Golf; GolfShoes; 1
```

```
Golf; GolfTees; 1
```

```
Golf; GolfClubs; 1
```

## 차원이 메타데이터 파일에 없음

연도, 기간 및 뷰 차원이 메타데이터 파일에 포함되어 있지 않습니다. 이러한 차원은 애플리케이션을 정의할 때 지정한 애플리케이션 프로파일에서 정의합니다.

값 및 내부거래 파트너 차원 멤버는 대체로 시스템에서 정의됩니다. 하지만 값 멤버의 설명과 일부 내부거래 파트너 차원 멤버의 보안 클래스 및 설명을 정의할 수 있습니다.

- 값 - 표준 멤버는 자동으로 생성됩니다. 이외에도 메타데이터를 로드하면 로드된 각 통화에 대해 *CurrencyName*, *CurrencyName Adjs* 및 *CurrencyName Total*의 값 차원 멤버 3쌍이 자동으로 생성됩니다. 여기서 *CurrencyName*은 통화 레이블입니다. 예를 들어 USD 통화의 경우 Oracle Hyperion Financial Management는 값 차원 멤버 USD, USD Adjs 및 USD Total을 생성합니다. 시스템 생성 멤버의 설명 및 시스템에서 사용자 정의 통화에 대해 생성하는 값 멤버에 추가되는 설명을 정의할 수 있습니다.
- 내부거래 파트너 - 이 차원은 자동으로 생성됩니다. 내부거래 파트너 차원 멤버는 IsICP 속성이 TRUE로 설정된 각 엔티티 차원 멤버에 대해 생성됩니다. [내부거래 파트너](#)에 설명된 일부 내부거래 파트너 멤버에 대해 보안 클래스와 설명을 정의할 수 있습니다.

## 메타데이터 참조 무결성

애플리케이션에서 발생하는 참조 무결성 문제를 방지하려면 메타데이터 변경사항을 수락하기 전에 Oracle Hyperion Financial Management에서 이 변경사항이 현재 상태의 애플리케이션에서 적합한지 확인합니다.

메타데이터를 로드할 때 시스템에서는 메타데이터 로드 파일과 애플리케이션의 메타데이터 요소를 비교합니다. 모든 변경사항이 기록되고 일부 변경사항은 기존 데이터에 대해 확인됩니다. 참조 무결성 문제가 발생하는 수정 내용은 허용되지 않습니다.

 주:

데이터를 잠가서 저장된 데이터가 변경되지 않도록 방지합니다. 계층이 변경되는 경우 집계된 데이터가 변경되지 않도록 방지하지 않습니다. 기존 집계에 영향을 주지 않고 계층 전체를 변경하려면 기존 계층을 변경하지 말고 새 계층을 추가하여 이전 계층을 유지해야 합니다.

## 메타데이터 참조 무결성 검사

로그 파일에서는 시스템이 로드되는 메타데이터 파일과 관련해서 기존 데이터를 확인하도록 하는 메타데이터 속성의 특정 변경사항에 대한 정보를 제공합니다.

또한 시스템은 로드 파일과 애플리케이션의 메타데이터 간에 부적합한 POV가 있는지 확인합니다. 차원 멤버가 로드 파일에 없고 애플리케이션의 분개에 있는 경우 메타데이터 로드가 금지됩니다.

## 메타데이터 로그 파일 참조 무결성 오류

메타데이터 로그 파일에서 다음 섹션 아래에 참조 무결성 오류가 표시됩니다.

```
Metadata referential integrity check started at
```

참조 무결성 검사 섹션의 각 라인은 로그 파일의 메타데이터 무결성 오류를 나타냅니다. 무결성 검사 중에 확인된 오류는 다음 형식으로 표시됩니다.

```
Journals::SINGLECA1 Scenario::Actual Year::2014
Value::[Contribution Adjs]
Period::January has 1 occurrences of
Changed::[SCENARIO::Actual::ZeroViewForAdj: Periodic to YTD]
```

다음 예에서는 POV: 시나리오 실제, 연도 2014, 값 [Contribution Adjs], 기간 1월 SINGLECA1 분개에서 메타데이터 무결성 오류가 발생하는 경우를 보여 줍니다. 오류는 실제 시나리오에 대한 ZeroViewForAdj 속성이 [주기적]에서 YTD로 변경되었다는 것입니다. 실제 시나리오에 대한 분개가 있으므로 이러한 변경은 허용되지 않습니다.

## 메타데이터 병합 유틸리티 사용

기존 릴리스에서 새 릴리스로 업그레이드하는 경우 메타데이터 병합 유틸리티를 사용하여 기존 메타데이터 파일과 최신 버전의 파일을 병합할 수 있습니다. 이렇게 하면 기존 메타데이터 파일의 수정 사항을 유지할 수 있습니다. 이 유틸리티는 Financial Management\Utilities 폴더에 있습니다.

 주:

메타데이터 병합 유틸리티는 XML 형식의 메타데이터에서만 작동하고 APP 형식에서는 작동하지 않습니다.

## 사용법

```
MetadataMerge.bat -b<Base File>[i<Ignore descriptions>]-l <Latest
file>-m <Modified file>[-o <Output File>]
```

### 설명:

**-b, -base <기본 파일>** = 파일 이름과 확장자를 포함하는 기본 버전의 메타데이터 파일 전체 경로

**-l, -latest <최신 파일>** = 파일 이름과 확장자를 포함하는 릴리스 버전의 메타데이터 파일 전체 경로

**-m, -modified <수정된 파일>** = 파일 이름과 확장자를 포함하는 수정된 메타데이터 파일 전체 경로

**-o, -output <출력 파일>** = 업데이트된 메타데이터 및 메타데이터 차이 보고서를 저장할 출력 메타데이터 파일 경로

**-i, -ignoredesc <설명 무시>** = 멤버 설명 변경 무시

### 예

```
-b c:\temp\MetadataMerge\Comma_V1_B.xml
```

```
-l c:\temp\MetadataMerge\Comma_V2_R.xml
```

```
-m c:\temp\MetadataMerge\Comma_Customer_M.xml (User-modified metadata
based on Comma_V1_B.xml file)
```

```
-i true
```

```
명령: MetadataMerge.bat -b c:\temp\MetadataMerge\Comma_Metadata_B.xml -l
c:\temp\MetadataMerge\Comma_Metadata_R.xml -m
c:\temp\MetadataMerge\Comma_Metadata_M.xml -i true
```

메타데이터 병합 유틸리티를 사용하려면 다음을 수행합니다.

1. 파일 탐색기 또는 명령행에서 MetadataMerge.bat를 실행합니다.
2. 병합 프로세스 중 메타데이터 충돌이 있을 경우 경고 메시지가 표시됩니다. 다음 옵션 중 하나를 선택합니다.
  - Y - 예. 시스템이 최신 릴리스 파일의 변경사항을 적용하고 기존 파일에 병합합니다.
  - N - 아니요. 시스템이 최신 릴리스 파일의 변경사항을 적용하지 않습니다.
  - MA - 모두 병합. 최신 릴리스 파일의 모든 변경사항이 적용됩니다. 시스템이 추가 충돌에 대한 메시지를 표시하지 않습니다.
  - MN - 병합 안 함. 변경사항이 적용되지 않습니다. 시스템이 추가 충돌에 대한 메시지를 표시하지 않습니다.
3. Images 폴더에 있는 두 이미지를 Metadata Differences Report.html 파일이 생성된 경로에 복사합니다. 해당 이미지는 메타데이터 차이 보고서의 트리에 대한 확장 및 축소 아이콘입니다.

## 메타데이터 로드

메타데이터 파일을 로드하면 Oracle Hyperion Financial Management에서 메타데이터를 로드 파일의 새 메타데이터로 바꿉니다. 바꾸기는 계정 추가와 같은 메타데이터의 사소한 변경에 유용합니다. 예를 들어 애플리케이션에 North America 엔티티가 있고 메타데이터 파일에서 항목을 로드하는 경우 파일의 North America 항목 속성이 애플리케이션의 North America 항목 속성을 바꿉니다.

### 주:

메타데이터 파일에 앰퍼샌드 문자(&)를 사용하지 마십시오. 사용하면 오류가 발생합니다.

메타데이터 파일을 로드하면 시스템이 파일 로드를 진행하기 전에 통합, 데이터 입력 또는 기타 로드 프로세스와 같은 다른 태스크가 완료되기를 기다립니다. 장기 실행 통합 등의 작업을 수행하는 동안이 아니라 서버 클러스터에서 가벼운 작업이 수행되는 기간 동안 메타데이터를 로드하는 것이 좋습니다. [실행 중인 태스크] 페이지를 검사하여 진행 중인 통합이나 데이터 로드 등을 확인할 수 있습니다.

큰 메타데이터 파일을 로드하면 프록시 시간 초과 오류가 발생할 수 있습니다. 오류가 발생하면 웹 프록시 시간 초과 설정을 늘립니다.

메타데이터 파일을 애플리케이션에 로드하면 해당 애플리케이션의 사용자는 시스템이 변경되었으며 애플리케이션에서 로그오프한 후 다시 로그인해야 한다는 통지를 받습니다.

### 주의:

고아 멤버는 로드하기 전에 삭제해야 합니다. 고아 멤버를 삭제하지 않으면 메타데이터가 업데이트되지 않습니다.

[병합] 또는 [바꾸기] 로드 옵션을 선택해야 합니다. 새 메타데이터를 로드하기 전에 메타데이터를 모두 지우고 데이터 무결성을 확인할 수도 있습니다.

**표 4-14 메타데이터 로드 옵션**

로드 옵션	설명
병합	차원 멤버가 로드 파일과 애플리케이션 데이터베이스에 모두 있는 경우 데이터베이스의 멤버가 로드 파일의 멤버로 바뀝니다. 데이터베이스에 로드 파일에서 참조되지 않는 다른 차원 멤버가 있는 경우 데이터베이스의 해당 멤버는 변경되지 않습니다. 예를 들어 데이터베이스에 엔티티 CT, MA 및 CA가 있습니다. 병합 메소드를 사용하여 CA에 대한 새 정보만 포함된 메타데이터 파일을 로드합니다. 이 경우 데이터베이스의 CA는 새 정보로 업데이트되지만 MA와 CT는 변경되지 않고 데이터베이스에 그대로 유지됩니다.



표 4-14 (계속) 메타데이터 로드 옵션

로드 옵션	설명
바꾸기	<p>애플리케이션 데이터베이스에 있는 차원 멤버가 모두 삭제되고 로드 파일의 멤버가 데이터베이스에 삽입됩니다.</p> <p>예를 들어 데이터베이스에 엔티티 CT, MA 및 CA가 있습니다. 바꾸기 메소드를 사용하여 CA에 대한 새 정보만 포함된 메타데이터 파일을 로드합니다. 데이터베이스에서 CT와 MA는 삭제되고 로드 파일의 새 정보가 포함된 CA 엔티티만 유지됩니다.</p>
로드하기 전에 모든 메타데이터를 지웁니다.	<p>애플리케이션 데이터베이스에 있는 모든 차원 멤버와 해당 데이터, 분개 및 내부거래 트랜잭션이 삭제됩니다.</p> <p>이 옵션을 선택하면 병합 및 바꾸기 메소드의 기능이 재정의됩니다.</p>
무결성 확인	<p>데이터에 대해 메타데이터를 검사하여 무결성을 확인합니다. <a href="#">메타데이터 참조 무결성</a>을 참조하십시오.</p>

 주:

웹 팝업 차단剂的 예외에 Financial Management를 추가하는 것이 좋습니다. 데이터 로드와 같은 일부 태스크를 수행하면 태스크 상태를 표시하는 상태 창이 팝업됩니다. 컴퓨터에 팝업 차단剂이 설정되어 있으면 상태 창이 표시되지 않습니다.

메타데이터를 로드하려면 다음을 수행합니다.

1. 애플리케이션을 엽니다.
2. 통합, 로드, 애플리케이션 요소 순으로 선택합니다.
3. 메타데이터 섹션에서 로드할 파일 이름을 입력하거나 **찾아보기**를 눌러 파일을 찾습니다.

 주:

기본적으로 메타데이터 파일은 XML 또는 APP 파일 확장자를 사용합니다. 로드 프로세스에서도 TXT 파일 확장자가 허용되지만 XML 또는 APP 파일 확장자를 사용하는 것이 좋습니다.

4. 선택 사항: 메타데이터 및 데이터 지우기를 선택합니다.

 주:

이 옵션을 선택하면 [메타데이터 옵션] 섹션에서 요소를 선택할 수 없습니다.

- 선택 사항: 무결성 확인**을 선택하여 현재 애플리케이션의 데이터에 대해 메타데이터 파일을 확인합니다.

이 옵션은 애플리케이션이 로드 파일의 메타데이터에 의해 부정적인 영향을 받지 않도록 하므로 선택하는 것이 좋습니다.

 **주:**

무결성 오류가 발생할 경우 메타데이터 로그 파일에 기록되며 파일의 어떤 부분도 애플리케이션에 로드되지 않습니다.

이 절차를 계속하기 전에 오류를 수정해야 합니다. [메타데이터 로그 파일 참조 무결성 오류](#)를 참조하십시오.

- 로드 옵션** 섹션에서 다음 로드 방법 중 하나를 선택합니다.

- 병합
- 바꾸기

- 구분자**에서 파일의 메타데이터를 구분하는 데 사용되는 문자를 선택합니다.

구분자는 APP 파일 확장자를 사용하는 ASCII 파일에만 필요하고 XML 파일에는 필요하지 않습니다. 적합한 문자는 다음과 같습니다.

, ~ @ \$ % ^ | : ; ? \

 **주:**

파일 이름이나 파일에서 다른 모든 방식으로 사용되지 않는 문자를 사용합니다. 예를 들어 엔티티 설명에 심표를 사용하는 경우에는 심표를 구분자로 사용할 수 없습니다.

- 필터** 섹션에서 로드할 메타데이터 유형을 선택합니다.

 **팁:**

필터 선택을 재설정하려면 **재설정**을 누릅니다.

- 선택 사항: 스캔**을 눌러 파일 형식이 정확한지 확인합니다.

- 로드**를 누릅니다.

- 선택 사항:** 로그 파일을 다운로드하려면 **로그 다운로드**를 누릅니다. **열기**를 눌러 로그 파일을 표시하거나 **저장**을 누르고 로컬에서 파일을 저장할 위치를 선택합니다.

## 메타데이터 로드 변경사항 보기

메타데이터 파일이 성공적으로 로드된 후 로드 프로세스 중 수행된 메타데이터 변경사항에 대한 자세한 정보를 볼 수 있습니다. 예를 들어 계정 차원에서 속성이 변경된 경우 이전 속성 값과 새 속성 값이 표시됩니다. [태스크 감사] 모듈에서 이 정보의 보고서를 볼 수 있습니다.

메타데이터 로드 변경사항 보고서에는 다음 세부정보가 포함되어 있습니다.

- 추가된 멤버
- 삭제된 멤버
- 멤버 속성 변경사항(설명 변경사항은 포함되지 않음)
- 새 상위 멤버/하위 멤버 관계
- 삭제된 상위 멤버/하위 멤버 관계
- 사용자정의 차원의 상위 멤버-하위 멤버 관계에 대한 합산 가중치의 변경사항

#### 샘플 메타데이터 로드 보고서 출력

\*\* Entity \*\*

Added member WestRegion.

Changed IsICP for China from Y to N.

Added Parent/Child Regional/Asia.

Removed Parent/Child Europe/Bulgaria.

\*\* Account \*\*


Renamed member SalesIC to SalesInterco.

Deleted member EastSales.

\*\* Scenario \*\*

Added member Forecast.

메타데이터 로드 변경사항을 보려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리, 감사, 태스크** 순으로 선택합니다.
3. 태스크 감사 **활동** 열에서 보려는 메타데이터 로드 프로세스 옆에 있는 링크 아이콘()을 누릅니다.
4. **열기**를 누르고 아무 텍스트 편집기에서나 보고서를 봅니다.

## 메타데이터 추출

메타데이터를 추출하여 보거나 수정할 수 있습니다. 메타데이터를 추출하는 경우 파일을 XML 또는 APP 파일로 저장하고 파일 이름과 위치를 지정합니다. 메타데이터를 수정한 후 변경사항을 적용하려면 수정된 파일을 애플리케이션에 재로드해야 합니다.

값 차원과 같은 시스템 정의 차원의 멤버는 추출할 수 없습니다. 또한 애플리케이션 프로파일에 정의된 연도 및 기간과 같은 차원의 멤버는 추출할 수 없습니다.

메타데이터를 추출하려면 다음을 수행합니다.

1. 애플리케이션을 엽니다.
2. 통합, 추출, 애플리케이션 요소 순으로 선택합니다.
3. 메타데이터 섹션의 구분자에서 파일의 메타데이터를 구분하는 데 사용되는 문자를 선택합니다.

구분자는 .app 파일 확장자를 사용하는 ASCII 파일에만 필요하고 XML 파일에는 필요하지 않습니다. 적합한 문자는 다음과 같습니다.

, ~ @ \$ % ^ | : ; ? \

### 주:

파일 이름이나 파일에서 다른 모든 방식으로 사용되지 않는 문자를 사용합니다. 예를 들어 엔티티 설명에 심표를 사용하는 경우에는 심표를 구분자로 사용하지 않습니다.

4. 파일 형식을 선택합니다.
  - **Financial Management(.app)**
  - **Financial Management(.xml)**
5. 필터에서 추출할 메타데이터 유형을 선택합니다.
6. 추출을 누릅니다.
7. 브라우저에 표시되는 다운로드 지침에 따라 추출된 파일을 다운로드합니다.
 

지침은 사용 중인 웹 브라우저에 따라 달라집니다. 설정한 웹 디렉토리에 파일을 저장해야 합니다.
8. **선택사항:** 로그 파일을 다운로드하려면 **로그 다운로드**를 누릅니다. 열기를 눌러 로그 파일을 표시하거나 **저장**을 누르고 로컬에서 파일을 저장할 위치를 선택합니다.

# 5

## 멤버 목록 관리

멤버 목록을 사용하면 차원 내에서 멤버 서브세트를 지정할 수 있으며, 멤버 계층을 찾는 데 걸리는 시간을 줄일 수 있습니다. 예를 들어 Italy, France 및 UK가 엔티티 차원의 POV에서 자주 사용되는 경우 자주 사용되는 멤버가 포함된 European Region이라는 멤버 목록을 생성할 수 있습니다. 그러면 European 멤버를 전체 엔티티 차원 계층에서 찾지 않고 이 목록에서 선택할 수 있습니다.

규칙에 멤버 목록을 사용할 수도 있습니다. 예를 들어 멤버 목록의 모든 멤버를 계산하는 규칙을 작성할 수 있습니다.

Oracle Hyperion Financial Management에서는 다음 두 가지 유형의 멤버 목록을 사용합니다.

- 시스템 정의 멤버 목록
- 사용자 정의 멤버 목록

애플리케이션이 생성되고 메타데이터가 로드되면 시스템 정의 멤버 목록이 생성됩니다. 이러한 멤버 목록은 지정한 상위 멤버의 모든 하위 멤버나 지정한 차원의 모든 기본 멤버와 같은 공용 등록정보에 따라 멤버를 그룹화합니다. 시스템 정의 멤버 목록 이름은 대괄호로 묶여 있습니다(예: [Descendants]).

사용자 정의 멤버 목록에는 사용자가 지정하는 멤버가 포함되어 있습니다. 모든 Financial Management 차원에 대해 멤버 목록을 생성할 수 있습니다.

정적 멤버 목록과 동적 멤버 목록을 생성할 수 있습니다.

정적 멤버 목록에는 사용자 지정 차원 멤버가 포함됩니다. 예를 들어 Sales, Purchases, Salaries, OtherCosts 및 TotalCosts 계정만 포함된 ProfitAndLoss라는 정적 계정 멤버 목록을 생성할 수 있습니다. 정적 목록에 멤버를 추가하려면 수동으로 멤버를 추가해야 합니다.

동적 멤버 목록에는 런타임에 지정한 기준을 충족하는 멤버가 포함됩니다. 목록이 동적으로 빌드되기 때문에 검색 시 업데이트됩니다. 시나리오, 연도, 기간 및 엔티티 차원의 경우 POV의 현재 멤버를 목록의 시작 지점으로 사용할 수 있습니다.

### 주:

같은 파일에 정적 멤버 목록과 동적 멤버 목록을 모두 포함할 수 있습니다.

Financial Management용 샘플 애플리케이션을 설치하면 샘플 멤버 목록 파일이 포함됩니다. 이러한 파일은 Financial Management를 설치한 디렉토리의 Sample Applications 폴더에 있습니다.

## 멤버 목록 파일 생성

Notepad ++ 등의 텍스트 편집기를 사용하여 멤버 목록 파일을 생성한 다음 파일을 애플리케이션에 로드할 수 있습니다.

Microsoft Visual Basic 스크립트 구문을 사용하여 각 정적 및 동적 목록에 멤버를 추가할 수 있습니다.

 주:

멤버 목록 이름에 따옴표("") 또는 앰퍼샌드(&)를 사용할 수 없습니다.

기본적으로 멤버 목록 파일은 LST 파일 확장자를 사용합니다.

다음과 같은 서브루틴을 사용하여 멤버 목록 파일을 생성합니다.

- Sub `EnumMemberLists` (): 멤버 목록이 포함되는 차원을 지정하고 각 차원의 멤버 목록을 정의합니다. 각 차원의 목록 수와 해당 차원 내의 각 멤버 목록 이름을 정의합니다.
- Sub `EnumMembersInList` (): 각 멤버 목록 내의 멤버를 정의합니다.

 주:

`HS.Dimension` 함수는 차원의 차원 별칭을 반환합니다. 차원 이름과 차원 별칭(긴 이름)을 지정할 수 있는 사용자정의 차원의 경우 `HS.Dimension` 함수에서 차원 별칭(긴 이름)을 반환합니다.

스크립트에 동적 멤버 목록을 추가하는 방법에 대한 자세한 내용은 [동적 멤버 목록을 참조하십시오](#).

## EnumMemberLists

각 멤버 목록 파일에는 멤버 목록이 있는 차원, 각 차원에 대한 목록 수 및 각 멤버 목록의 이름을 지정하는 `EnumMemberLists` 서브루틴이 포함되어야 합니다. `EnumMemberLists` () 서브루틴 내에서 다음 표의 구문과 함수를 사용하여 멤버 목록을 정의합니다.

구문	설명
<code>Dim ElementLists(n)</code>	<p>각 명령문에 대한 목록 수를 지정합니다.</p> <p>여기서 <i>Element</i>는 멤버 목록이 생성되는 차원 이름이고, <i>n</i>은 해당 차원에 대해 정의되는 멤버 목록의 총수입니다. 예를 들어 파일에 엔티티 차원에 대한 3개의 멤버 목록이 포함된 경우 구문은 다음과 같습니다.</p> <pre>Dim EntityLists(3)</pre>
<code>HS.Dimension = "<i>Element</i>"</code>	<p>여기서 <i>Element</i>는 차원 이름입니다. 예:</p> <pre>If HS.Dimension = "Entity" Then</pre>

구문	설명
<code>ElementLists(n) = "ListName"</code>	<p>각 목록의 이름과 숫자 ID를 지정합니다.</p> <p>여기서 <i>Element</i>는 차원 이름이고, <i>n</i>은 멤버 목록의 ID 번호이고, <i>ListName</i>은 목록의 이름입니다.</p> <p>@POV 키워드를 사용하여 현재 POV에 설정된 멤버 목록에 따라 동적 목록을 생성할 수 있습니다. 엔티티 목록에 표시되는 엔티티는 보고서의 POV에서 선택된 시나리오, 연도, 기간 및 엔티티를 기반으로 할 수 있습니다.</p> <p>예:</p> <pre>EntityLists(1) = "NewEngland"  EntityLists(2) = "Alloc"  EntityLists(3) = "AllEntities(@POV)"</pre>
<code>HS.SetMemberLists ElementLists</code>	<p>지정된 이름과 숫자 ID를 설정합니다.</p> <p>여기서 <i>Element</i>는 차원 이름입니다. 예:</p> <pre>HS.SetMemberLists EntityLists</pre>

## EnumMembersInList

EnumMembersInList 서브루틴을 사용하여 목록에 멤버를 추가합니다. 정적 멤버 목록의 경우 스크립트에서 목록의 모든 멤버를 나열합니다. EnumMembersInList () 서브루틴 내에서 다음 표의 구문과 함수를 사용하여 각 멤버 목록의 멤버를 정의합니다.

엔티티를 제외한 모든 차원의 경우 멤버를 지정해야 하는 HS.AddMemberToList 문이 사용됩니다. 엔티티 차원의 경우 멤버 및 상위 멤버를 지정해야 하는 HS.AddEntityToList 문이 사용됩니다.

**표 5-1 EnumMembersInList 구문**

구문	설명
<code>HS.Dimension = "Element"</code>	<p>여기서 <i>Element</i>는 차원 이름입니다. 예:</p> <pre>If HS.Dimension = "Entity" Then</pre>

표 5-1 (계속) EnumMembersInList 구문

구문	설명
HS.MemberListID= <i>n</i>	숫자 ID로 멤버 목록을 지정합니다. 여기서 <i>n</i> 은 EnumMemberLists 서브루틴에서 멤버 목록에 지정된 숫자 ID입니다.  HS.MemberListID = 1
HS.AddEntityToList <i>Member</i> HS.AddMemberToList <i>Member</i>	엔티티가 아닌 차원의 목록에 멤버를 추가합니다. 여기서 <i>Member</i> 는 멤버 이름입니다.  HS.AddEntityToList "UnitedStates, "Maine"  HS.AddMemberToList "July"
HS.AddEntityToList <i>Parent, Member</i>	엔티티 차원의 목록에 멤버를 추가합니다. 여기서 <i>Parent</i> 는 추가되는 멤버의 상위 멤버이고, <i>Member</i> 는 엔티티 차원의 멤버입니다.  HS.AddEntityToList "UnitedStates", "California"
HS.Entity.List	엔티티 차원의 목록에 멤버를 추가합니다.  HS.Entity.List (" "[Base] )"
HS.MemberListEntity HS.MemberListScenario HS.MemberListYear HS.MemberListPeriod	동적 멤버 목록을 지정하는 데 사용됩니다.



이 섹션에서는 파일의 EnumMembersInList 섹션 샘플을 보여 줍니다. 이 예에서는 엔티티 목록 세 개의 엔티티가 정의됩니다. 계정 목록의 멤버도 정의됩니다.

```
Sub EnumMembersInList()  
  
If HS.Dimension = "Entity" Then  
  
    If HS.MemberListID = 1 Then  
  
        HS.AddEntityToList "United States", "Massachusetts"  
  
        HS.AddEntityToList "United States", "Rhode Island"  
  
        HS.AddEntityToList "United States", "Maine"  
  
    ElseIf HS.MemberListID = 2 Then  
  
        HS.AddEntityToList ,"United States" "Connecticut"  
  
    ElseIf HS.MemberListID = 3 Then  
  
        HS.AddEntityToList "United States", "California"  
  
    End If  
  
    ElseIf HS.Dimension = "Account" Then  
  
        If HS.MemberListID = 1 Then  
  
            HS.AddMemberToList "Sales"  
  
            HS.AddMemberToList "Purchases"  
  
            HS.AddMemberToList "Salaries"  
  
            HS.AddMemberToList "OtherCosts"  
  
            HS.AddMemberToList "TotalCosts"  
  
            HS.AddMemberToList "GrossMargin"
```

 주:

각 차원에 대해 필요한 개수만큼 멤버 목록을 보유할 수 있으며 모든 차원에 대해 멤버 목록을 생성할 필요는 없습니다.

## 동적 멤버 목록

동적 멤버 목록의 경우 멤버 목록의 모든 멤버를 나열하는 대신 지정된 기준을 충족하는 멤버를 선택하는 규칙을 입력합니다. 기준은 통화, 계정 유형 등의 멤버 등록정보입니다. 사용자가 목록에 액세스할 때마다 동적으로 목록이 생성됩니다.

Oracle Hyperion Financial Management 함수와 인수를 사용하여 멤버 목록을 빌드할 수 있습니다.

다음 구문에서는 모든 USD 엔티티를 가져오는 동적 멤버 목록을 생성합니다.

```
If HS.Dimension = "Entity" Then

    If HS.MemberListID=1 Then

        ELi=HS.Entity.List("", "")

        'Entities are read into an array.

        For i=Lbound(ELi) to Ubound(ELi)

            'Loops through all entities.

            If (StrComp(HS.Entity.DefaultCurrency(ELi(i)),
                "USD",vbTextCompare)=0) Then

                HS.AddEntityToList "",ELi(i)

            'String compares default currency for entity to USD. If there is a match,
            the entity is added to the member list.

            End If

        Next

    End If

End If
```

이 예에서는 엔티티 목록이 배열에 수신됩니다. 배열의 각 엔티티에 대해 `DefaultCurrency` 등록정보의 값이 기본 값 `USD`와 비교됩니다. 값이 `USD`와 같으면 엔티티가 목록에 추가되고 시스템에서 배열의 다음 엔티티를 처리합니다.

## 동적 POV 멤버 목록

동적 POV 멤버 목록은 하나 이상 차원의 현재 POV 멤버를 기준으로 동적으로 생성됩니다.

이 샘플 멤버 목록 파일의 굵은꼴 섹션에서는 동적 POV 섹션을 보여 줍니다.

```
Sub EnumMemberLists ()

Dim EntityLists (5)

If HS.Dimension = "Entity"Then

    EntityLists (1) = "AllEntities"

    EntityLists (2) = "AppCur"

    EntityLists (3) = "NoappCur"

    EntityLists (4) = "Global (@POV) "

        EntityLists (5) = "POWN (@POV) "

    HS.SetMemberLists EntityLists

End If

End Sub

Sub EnumMembersInList ()

If HS.Dimension = "Entity" Then

    If HS.MemberListID = 1 Then

        HS.AddEntityToList "", "Corp_Ops"

        HS.AddEntityToList "", "China"

        HS.AddEntityToList , "", "Germany"

        HS.AddEntityToList "", "UK"
```

```
YearPOV=HS.MemberListYear
```

```
PerPOV=HS.MemberListPeriod
```

```
EntPOV=HS.MemberListEntity
```

```
If HS.MemberListID = 4 Or HS.MemberListID = 5 Then

    If ( EntPOV <> "" ) Then

        EntList=HS.Node.List("E#" & EntPOV,"[Descendants]", "S#" & ScenPOV &
        "..Y#" & YearPOV & ".P#" & PerPOV)

        If IsArray(EntList) Then

            For each Ent in EntList

                If Ent <> "[None]" Then

                    If HS.Node.Method("S#" & ScenPOV & ".Y#" & YearPOV & ".P#" &
                    PerPOV & ".E#" & EntPOV & "." & Ent)="GLOBAL" Then

                        If HS.MemberListID = 4 Then HS.AddEntityToList "",Ent

                    End If

                    If HS.Node.POwn("S#" & ScenPOV & ".Y#" & YearPOV & ".P#" &
                    PerPOV & ".E#" & EntPOV & "." & Ent) > 0.5 Then

                        If HS.MemberListID = 5 Then HS.AddEntityToList "",Ent

                    End If

                End If

            End If

        End If

    End If

Next
```

## 멤버 목록 로드

멤버 목록 스크립트 파일을 생성한 후 애플리케이션에 로드합니다. 파일을 로드하기 전에 스캔하여 정확한 형식이 지정되었는지 확인할 수 있습니다. 파일을 수정하는 경우 애플리케이션에 재로드해야 합니다.

멤버 목록 파일을 로드하면 시스템이 파일 로드를 진행하기 전에 통합, 데이터 입력 또는 기타 로드 프로세스와 같은 다른 태스크가 완료되기를 기다립니다. 장기 실행 통합 등의 작업을 수행하는 동안이 아니라 서버 클러스터에서 가벼운 작업이 수행되는 기간 동안 멤버 목록을 로드하는 것이 좋습니다. [실행 중인 태스크] 페이지를 검사하여 진행 중인 통합이나 데이터 로드 등을 확인할 수 있습니다.

멤버 목록 파일을 애플리케이션에 로드하면 해당 애플리케이션의 사용자는 시스템이 변경되었으며 애플리케이션에서 로그오프한 후 다시 로그온해야 한다는 통지를 받습니다. 계산 상태도 OK SC(시스템 변경)로 변경됩니다.

### 주:

웹 팝업 차단기의 예외에 Oracle Hyperion Financial Management를 추가하는 것이 좋습니다. 데이터 로드와 같은 일부 태스크를 수행하면 태스크 상태를 표시하는 상태 창이 팝업됩니다. 컴퓨터에 팝업 차단이 설정되어 있으면 상태 창이 표시되지 않습니다.

멤버 목록을 로드하려면 다음을 수행합니다.

1. 애플리케이션을 엽니다.
2. 통합, 로드, 애플리케이션 요소 순으로 선택합니다.
3. 멤버 목록 섹션에서 로드할 파일 이름을 입력하거나 **찾아보기**를 눌러 파일을 찾습니다.

### 주:

기본적으로 멤버 목록 파일은 LST 파일 확장자를 사용합니다. 로드 프로세스에서도 TXT 파일 확장자가 허용되지만 LST 파일 확장자를 사용하는 것이 좋습니다.

4. **선택 사항:** 스캔을 눌러 파일 형식이 정확한지 확인합니다.
5. 로드를 누릅니다.

 주:

부적합한 멤버 목록을 로드하면 사용자가 데이터 그리드에 액세스할 수 없습니다. 다음 오류 메시지가 표시되는 경우 멤버 목록을 수정한 후 파일을 로드해야 할 수도 있습니다.

EPMHFM-102: Error executing VBScript,  
An unexpected error has occurred. Check server log or contact your system administrator for more information.

6. **선택사항:** 로그 파일을 다운로드하려면 **로그 다운로드**를 누릅니다. 열기를 눌러 로그 파일을 표시하거나 **저장**을 누르고 로컬에서 파일을 저장할 위치를 선택합니다.

## 멤버 목록 추출

애플리케이션에서 멤버 목록을 추출할 수 있습니다. 추출한 멤버 목록은 ASCII 파일로 저장됩니다. 기본적으로 멤버 목록 파일은 LST 파일 확장자를 사용합니다. 멤버 목록을 추출한 후 텍스트 편집기에서 멤버 목록 정보를 보고 수정할 수 있습니다.

멤버 목록을 추출하려면 다음을 수행합니다.

1. 애플리케이션을 엽니다.
2. **통합, 추출, 애플리케이션 요소** 순으로 선택합니다.
3. **멤버 목록** 섹션에서 **추출**을 누릅니다.
4. 브라우저에 표시되는 다운로드 지침에 따라 추출된 파일을 다운로드합니다.

지침은 사용 중인 웹 브라우저에 따라 달라집니다. 설정한 웹 디렉토리에 파일을 저장해야 합니다.

5. **선택사항:** 로그 파일을 다운로드하려면 **로그 다운로드**를 누릅니다. 열기를 눌러 로그 파일을 표시하거나 **저장**을 누르고 로컬에서 파일을 저장할 위치를 선택합니다.

## 차원별 시스템 목록

다음 표에서는 시스템 생성 목록의 이름과 해당 목록을 사용할 수 있는 차원을 보여 줍니다.

표 5-2 차원별 시스템 목록

시스템 목록	시나리오	엔티티	계정	ICP	사용자정의 값	연도	기간		
[Hierarchy]		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Descendants]		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Children]			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>

표 5-2 (계속) 차원별 시스템 목록

시스템 목록	시나리오	엔티티	계정	ICP	사용자정의	값	연도	기간
[Base]			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
[Parents]			<input checked="" type="checkbox"/>					
[Ancestors]			<input checked="" type="checkbox"/>					
[System]				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
[Currencies]						<input checked="" type="checkbox"/>	C1, C2	
[ConsolMethod]						<input checked="" type="checkbox"/>	C1	
[Inputs]							<input checked="" type="checkbox"/>	
[Adjustments]							<input checked="" type="checkbox"/>	
[Totals]							<input checked="" type="checkbox"/>	
[Default Currencies]							<input checked="" type="checkbox"/>	
[First Generation]								<input checked="" type="checkbox"/>
[Second Generation]								<input checked="" type="checkbox"/>
[Third Generation]								<input checked="" type="checkbox"/>
[Fourth Generation]								<input checked="" type="checkbox"/>
[Fifth Generation]								<input checked="" type="checkbox"/>
[Sixth Generation]								<input checked="" type="checkbox"/>



# 6

## 분개 관리

대부분의 외부 일반 원장 시스템에서는 분개 정보가 포함된 ASCII 텍스트 파일을 생성할 수 있으며, 이 ASCII 파일을 Oracle Hyperion Financial Management 애플리케이션에 로드할 수 있습니다. 필요한 경우 ASCII 파일을 로드하기 전에 텍스트 편집기에서 편집할 수 있습니다.

Financial Management용 샘플 애플리케이션을 설치하면 샘플 분개가 포함됩니다. 이러한 파일은 Financial Management를 설치한 디렉토리의 Sample Applications 폴더에 있습니다.

## 분개 파일 생성

멀티바이트 문자 세트(MBCS)를 지원하는 ASCII 형식이나, Little Endian 바이트 순서를 사용하여 유니코드 형식으로 인코딩된 파일로 분개 파일을 생성할 수 있습니다. 기본적으로 분개 파일은 JLF 파일 확장자를 사용합니다.

보안 정보 파일은 다음 섹션으로 이루어질 수 있습니다.

- 파일 형식
- 버전
- 분개 그룹
- 표준
- 반복
- 머리글 - 시나리오, 연도, 기간

느낌표(!)로 시작하는 라인은 분개 파일에서 새 섹션의 시작을 나타내며 뒤에 적합한 섹션 이름(예: 연도)이 와야 합니다. 아포스트로피(')로 시작하는 라인은 주석 라인으로 간주되며 시스템에서 무시됩니다.

문자가 파일에서 다른 방식으로 사용되고 있지 않으면 다음과 같은 특수 문자를 사용하여 파일 내에서 정보를 구분할 수 있습니다.

문자	설명
&	앰퍼샌드
@	At 기호
\	백슬래시
^	캐럿
:	콜론
,	쉼표
\$	달러 기호
#	숫자 기호
%	퍼센트 기호
	파이프 기호
?	물음표

문자	설명
;	세미콜론
~	물결표

 주:

파일 전체에서 동일한 구분자를 사용해야 합니다. 한 파일 내에서 여러 구분자를 사용하면 파일을 로드할 때 오류가 발생합니다.

## 파일 형식 섹션

이 파일 섹션에는 파일 버전 번호가 포함되어 있습니다. 이 숫자는 파일 형식을 변경할 때만 변경됩니다. 분개를 추출하면 파일 형식이 자동으로 생성됩니다.

 주:

이 섹션은 필수가 아닙니다.

다음 예에서는 파일 형식을 지정합니다.

```
!File_Format = 1.0
```

## 버전 섹션

이 파일 섹션에는 분개를 추출하는 데 사용한 Oracle Hyperion Financial Management 버전이 포함되어 있습니다. 버전 번호는 분개를 추출할 때 자동으로 생성됩니다.

 주:

이 섹션은 필수가 아닙니다.

다음 예에서는 버전을 지정합니다.

```
!Version = 11.1
```

## 분개 그룹 섹션

이 파일 섹션에서는 이 구문을 사용하여 분개 그룹을 정의합니다.

```
!GROUP=<journal group>;<journal group description>
```

예를 들어 이 예에서는 다음 두 개의 분개 그룹을 정의합니다.

```
!GROUP=Allocations;Allocations Journals Group
!GROUP=Tax;Tax Journals Group
```

## 표준 섹션

표준 템플리트는 애플리케이션의 모든 시나리오, 연도 및 기간에 적용되며 시나리오, 기간 및 연도의 특정 조합에 종속되지 않습니다.

다음 구문에서는 표준 템플리트를 지정합니다.

```
!STANDARD = <label>, <balancing attribute>, <type>, <journal group>,
<securityclass>, <SingleParent.SingleEntity>
```

```
!DESC=<journal description>
```

```
<parent.entity>, <account>, <ICP>, <CustomDimensionName>, <amount type>,
<amount>, <line item desc>
```

## 반복 섹션

반복 템플리트는 애플리케이션의 모든 시나리오, 연도 및 기간에 적용되며 시나리오, 기간 및 연도의 특정 조합에는 종속되지 않지만 값 조정에 종속됩니다.



**주:**

자동 전환 템플리트에 대해서는 반복 템플리트를 생성할 수 없습니다. 유형 속성 값은 일반을 나타내는 **R**이어야 합니다.

다음 구문에서는 반복 템플리트를 지정합니다.

```
!RECURRING = <label>, <balancing attribute>, <type>, <value>, <journal
group>, <securityclass>, <SingleParent.SingleEntity>
```

```
!DESC=<journal description>
```

```
<parent.entity>, <account>, <ICP>, <CustomDimensionName>, <amount type>,
<amount>, <line item desc>
```

## 머리글 섹션

이 파일 섹션에는 시나리오, 연도 및 기간 정보가 포함되어 있습니다. 머리글 섹션 뒤에는 분개 유형 머리글 정보 및 해당 세부정보 라인이 있습니다. 특정 시나리오, 연도 및 기간에 따라 분개 정보가 구성됩니다. 분개 파일에 여러 개의 머리글 섹션을 지정할 수 있습니다.

다음 구문에서는 시나리오, 연도 및 기간을 지정합니다.

```
!SCENARIO= Actual
```

```
!YEAR = 2014
```

```
!PERIOD = January
```

머리글 섹션 뒤에는 2014년 1월 실제에 대한 분개 세부정보가 있습니다.

[머리글] 섹션의 [분개] 하위 섹션에서는 지정된 시나리오, 기간 및 연도에 대한 분개 세부정보를 제공합니다.

**테이블 1**에서는 분개 하위 섹션에서 사용되는 속성을 설명합니다. 이러한 속성은 일반 분개와 반복 및 표준 템플릿에 사용됩니다.

**표 6-1 분개 속성 설명**

속성	값
<label>	분개에 대한 사용자 정의 레이블로, 최대 20자까지 사용할 수 있습니다.
<balancing attribute>	<ul style="list-style-type: none"> <li>U = 대차불일치</li> <li>B = 합계의 대차가 일치함</li> <li>E = 엔티티별 대차가 일치함</li> </ul>
<type>	<ul style="list-style-type: none"> <li>R = 일반 분개</li> <li>A = 자동 차기 역분개</li> <li>V = 자동 당기 역분개</li> </ul> <p>시스템 생성 자동 전환 분개는 로드할 수는 없지만 추출할 수 있습니다.</p>
<status>	<ul style="list-style-type: none"> <li>W = 작업 중</li> <li>S = 제출됨</li> <li>A = 승인됨</li> <li>P = 게시됨</li> <li>R = 거부됨</li> </ul>
<value dimension>	<ul style="list-style-type: none"> <li>[Contribution Adjs]</li> <li>[Parent Adjs]</li> <li>&lt;Entity Curr Adjs&gt;</li> <li>&lt;Parent Curr Adjs&gt;</li> </ul>
<journal group>	<p>선택 사항: 사용자 정의 매개변수로, 최대 30자까지 사용할 수 있습니다.</p> <p>그룹을 사전 로드해야 합니다.</p>
<security class>	<p>선택 사항: 분개와 연결된 적합한 보안 클래스입니다.</p> <p>보안 클래스를 지정하지 않으면 분개에서 기본 보안 클래스를 가정합니다. 이 보안 클래스에 대한 액세스 권한이 있는 사용자만 분개에 액세스할 수 있습니다.</p>

표 6-1 (계속) 분개 속성 설명

속성	값
<singleparent.singleentity>	선택 사항: 분개의 모든 라인 항목에 사용되는 적합한 상위/엔티티 쌍입니다. 전체 분개에 대해 하나의 상위/엔티티 쌍을 지정하는 경우에는 parent.entity 속성이 사용되지 않습니다.
<journal description>	분개 설명으로, 최대 255자까지 사용할 수 있습니다. 각 라인이 !DESC= 구문으로 시작되는 경우 여러 라인으로 구성된 설명을 로드할 수 있습니다.
<parent.entity>	엔티티 차원의 적합한 멤버입니다. 상위는 값 차원의 기여금액 조정, 상위 조정 및 상위 통화 조정 멤버에만 필요합니다. 이 속성은 Single Parent.Single Entity 속성이 사용되지 않는 경우에만 사용됩니다.
<account>	계정 차원의 적합한 멤버입니다. 일반 분개의 경우 입력 계정이어야 하며 계정 유형은 REVENUE, EXPENSE, ASSET, LIABILITY, FLOW 또는 BALANCE여야 합니다.
<ICP>	선택 사항: 내부거래 파트너 차원의 적합한 멤버입니다. 이 속성은 선택 사항이지만 ICP None을 하나 이상 지정해야 합니다.
<custom>	선택 사항: 사용자정의 차원의 적합한 멤버입니다.
<amount type>	<ul style="list-style-type: none"> <li>• D = 차변</li> <li>• C = 대변</li> </ul>
<amount>	금액 유형이 차변인지 또는 대변인지에 관계없이 양수 금액입니다.
<line item description>	선택 사항: 특정 분개 세부정보에 대한 설명으로, 최대 50자까지 사용할 수 있습니다.

다음 구문에서는 일반 분개를 지정합니다.

```
!JOURNAL = <label>, <balancing attribute>, <type>, <status>, <value dimension>, <journal group>, <SecurityClass>, <SingleParent.SingleEntity>
```

```
!DESC=<journal description>
```

```
<parent.entity>, <account>, <ICP>, <CustomDimensionName>, <amount type>, <amount>, <line item desc>
```

## 분개 로드

표준 및 반복 분개 템플릿은 물론 작업, 거부됨, 제출됨, 승인됨 및 게시됨 상태의 분개를 로드할 수 있습니다. 자동화된 통합 분개는 통합 프로세스에서 생성되었으므로 로드할 수 없습니다.

분개를 로드하려면 먼저 분개를 로드할 기간을 열어야 합니다. *Oracle Hyperion Financial Management 사용자 가이드*의 "기간 관리"를 참조하십시오.

분개는 [바꾸기] 모드를 사용하여 로드됩니다. [바꾸기] 모드에서는 새 분개 데이터를 로드하기 전에 특정 분개 레이블의 데이터를 모두 지웁니다. 게시된 분개는 열려 있는 기간에 로드해야 합니다. 게시된 자동 전환 분개를 로드하는 경우 승인된 전환 분개가 다음 기간에 자동으로 생성되며 생성된 전환 분개를 게시해야 합니다.

기본적으로 분개 파일은 JLF 파일 확장자를 사용합니다. 로드 프로세스에서는 TXT 또는 CSV와 같은 다른 파일 확장자가 허용되지만 JLF 파일 확장자를 사용하는 것이 좋습니다.

기본 로드 옵션을 변경하면 모든 행에 대한 옵션이 업데이트됩니다. [재정의] 옵션을 사용하여 특정 행을 활성화하고 이 행에 대한 옵션을 업데이트할 수 있습니다.

로드 프로세스가 완료되면 오류를 볼 수 있도록 로그에 대한 링크가 표시됩니다.

#### 주:

웹 팝업 차단기의 예외에 Oracle Hyperion Financial Management를 추가하는 것이 좋습니다. 데이터 로드와 같은 일부 태스크를 수행하면 태스크 상태를 표시하는 상태 창이 팝업됩니다. 컴퓨터에 팝업 차단이 설정되어 있으면 상태 창이 표시되지 않습니다.

분개를 로드하려면 다음을 수행합니다.

1. **통합, 로드, 분개** 순으로 선택합니다.
2. **구분자**에 파일에서 분개 데이터를 구분하는 데 사용되는 구분자를 입력합니다. 적합한 문자는 다음과 같습니다.

, ~ @ \$ % & ^ | : ; ? \

#### 주:

파일 이름이나 파일에서 다른 모든 방식으로 사용되지 않는 문자를 사용합니다. 예를 들어 엔티티 설명에 심표를 사용하는 경우에는 심표를 구분자로 사용할 수 없습니다.

3. **선택 사항:** Firefox를 브라우저로 사용하는 경우 여러 파일을 로드할 수 있습니다. 아직 선택되지 않은 경우 **복수 선택**을 선택하고 **찾아보기**를 눌러 파일을 찾습니다.

한 번에 최대 10개 파일을 선택할 수 있습니다. 그러면 세부정보 행이 선택한 파일로 채워집니다.

파일 선택 콘텐츠 영역으로 파일을 끌어 놓을 수 있습니다.

4. **선택 사항:** **추가**를 눌러 분개를 로드할 행을 더 추가합니다.

#### 팁:

행을 삭제하려면 행을 선택하고 **삭제**를 누릅니다.

5. **선택 사항:** 행의 기본 파일 설정을 재정의하려면 **재정의**를 누릅니다.

 **팁:**

로드 옵션을 기본값으로 재설정하려면 **재설정**을 누릅니다.

6. **로드**를 누릅니다.

 **주:**

기존 파일을 재로드하는 경우 해당 파일을 다시 사용할지를 묻는 경고 프롬프트가 표시됩니다. 다시 사용하려면 **예**를 누릅니다.

## 분개 추출

애플리케이션에서 분개, 반복 분개 및 분개 템플릿을 추출할 수 있습니다. 분개를 추출할 시나리오, 연도, 기간, 엔티티 및 값 차원을 선택할 수 있습니다. 추출할 분개 상태, 분개 유형 및 분개 대차 계정 유형을 선택할 수 있습니다.

분개를 추출하면 멀티바이트 문자 세트(MBCS)를 지원하는 ASCII 파일에 저장됩니다. 기본적으로 분개 파일은 JLF 파일 확장자를 사용합니다. 분개를 추출한 후 텍스트 편집기에서 분개 정보를 보고 수정할 수 있습니다.

자동화된 통합 분개를 외부 시스템으로 추출할 수 있지만 Oracle Hyperion Financial Management로 다시 임포트할 수는 없습니다. 이러한 분개를 추출할 때 자동화된 통합 분개의 대차 계정 유형은 비어 있습니다.

추출 프로세스가 완료되면 오류를 볼 수 있도록 로그에 대한 링크가 표시됩니다.

분개를 추출하려면 다음을 수행합니다.

1. **통합, 추출, 분개** 순으로 선택합니다.
2. 분개를 추출할 시나리오 및 연도를 지정합니다.
3. **선택 사항:** 분개를 추출할 기간을 선택합니다.
4. **선택 사항:** 추출할 엔티티 및 값 차원을 하나 이상 선택합니다.

자동화된 통합 분개를 추출하려면 [Proportion] 및/또는 [Elimination]을 값 멤버로 선택해야 합니다.

 **주:**

기간, 엔티티 및 값 차원의 경우 특정 멤버를 선택하지 않으면 차원의 모든 멤버를 추출하는 것으로 가정합니다. 하지만 구체적으로 멤버를 선택하면 차원 옆에 더하기 기호(+)가 표시되어 복수 선택을 나타냅니다.

5. 구분자에 파일에서 분개 데이터를 구분하는 데 사용되는 구분자를 입력합니다. 적합한 문자는 다음과 같습니다.

, ~ @ \$ % & ^ | : ; ? \

 주:

파일 이름이나 파일에서 다른 모든 방식으로 사용되지 않는 문자를 사용합니다. 예를 들어 엔티티 설명에 심표를 사용하는 경우에는 심표를 구분자로 사용할 수 없습니다.

6. 추출할 분개 유형을 선택합니다.
- 템플릿
  - 반복 템플릿
  - 분개
7. 선택 사항: 추출할 분개의 [레이블]과 [그룹]을 입력합니다.  
퍼센트 기호(%)를 와일드카드로 사용할 수 있습니다.  
자동화된 통합 분개의 경우 [레이블] 필드에서 통합 규칙에 지정한 감사 트랜잭션의 [특성] 값을 쿼리할 수 있습니다. 숫자는 레이블에 사용할 수 없습니다.
8. 추출할 분개 상태, 분개 유형 및 잔액 유형을 선택하거나 모두를 선택합니다.
9. 추출을 누릅니다.
10. 브라우저에 표시되는 다운로드 지침을 따릅니다.  
지침은 사용 중인 웹 브라우저에 따라 달라집니다. 설정한 웹 디렉토리에 파일을 저장해야 합니다.



# 7

## 데이터 양식 관리

데이터 양식은 사용자가 특정 데이터를 보고 입력할 수 있도록 설정된 사전 정의된 행과 열이 포함된 템플릿입니다. 사용자는 데이터 양식에 행과 열을 추가하거나 행과 열을 제거할 수 없습니다. 양식을 설정할 때 지정된 차원에 대해서만 POV 선택을 변경할 수 있습니다. 한 양식에서 다른 양식으로 연결되는 링크를 정의하여 사용자가 연결된 양식에서 데이터를 보기 위해 드릴스루하도록 할 수 있습니다.

데이터 양식을 생성하려면 [관리자] 또는 [데이터 입력 양식 관리] 보안 역할이 지정되어야 합니다. 데이터 양식을 로드, 추출 및 삭제하려면 [데이터 입력 양식 관리] 역할에 지정되어야 합니다.

다음 두 가지 방법으로 데이터 양식을 생성할 수 있습니다.

- 양식 디자이너의 데이터 양식 옵션 사용
- 스크립트 뷰에서 스크립트 작성

기본적으로 데이터 양식은 WDF 파일 확장자를 사용합니다.

[디자이너] 뷰와 [스크립트] 뷰를 쉽게 전환할 수 있습니다. 스크립트에 오류가 있으면 [스크립트] 뷰에 표시됩니다. [스크립트] 뷰에서 데이터 양식을 인쇄할 수도 있습니다.

데이터 양식의 [디자이너] 뷰에서 양식 열기로 전환할 수 있습니다. 양식을 저장하라는 메시지가 표시된 후 데이터 양식이 표시됩니다.

## 양식 디자이너에서 데이터 양식 생성

양식 디자이너를 사용하여 POV를 설정하고 양식 세부정보, 행, 열 및 머리글 옵션을 지정합니다.

데이터 양식을 생성한 후 스캔하여 유효성을 확인할 수 있습니다. 양식 스크립트가 생성되고 검증됩니다. 오류가 있으면 [스크립트] 뷰에 표시됩니다. [데이터 양식 스크립트 사용](#)을 참조하십시오.

양식 세부정보 설정에 대한 자세한 내용은 다음을 참조하십시오.

- [POV 설정](#)
- [양식 세부정보 지정](#)
- [양식 행 및 열 옵션 지정](#)
- [양식 머리글 지정](#)

데이터 양식을 생성하려면 다음을 수행합니다.

1. **통합, 문서** 순으로 선택합니다.
2. **새로 작성, 데이터 양식** 순으로 누르거나 **작업, 새로 작성, 데이터 양식** 순으로 선택합니다.

기본적으로 양식 디자이너가 표시됩니다.

 **팁:**

[디자이너] 페이지가 자동으로 표시되지 않는 경우 **디자이너**를 누르거나 **작업, 디자이너** 순으로 선택합니다.

3. 데이터 양식의 POV를 설정합니다.
4. 양식 세부정보를 지정합니다.
5. 행 및 열 옵션을 지정합니다.
6. **선택사항:** 행 또는 열을 추가하려면 **새 행 추가**, **새 열 추가**를 누르거나 **작업, 새 행 추가** 또는 **새 열 추가** 순으로 선택합니다.

 **팁:**

행 또는 열을 삭제하려면 **열/행 삭제**를 누르거나 **작업, 열/행 삭제** 순으로 선택합니다.

7. 머리글 옵션을 지정합니다.

 **팁:**

양식을 기본값이나 저장된 최종 정의로 재설정하려면 **재설정**을 누릅니다.

8. **선택 사항:** 스캔을 눌러 양식의 유효성을 확인합니다.
9. **저장**을 눌러 양식을 저장하고 양식에 대한 정보를 입력한 다음 **저장**을 누릅니다.

## POV 설정

POV 막대를 사용하여 배경 POV 및 선택 가능 POV를 정의할 수 있습니다. 양식의 백그라운드 POV는 각 차원에 대해 데이터 양식에 표시되는 초기 값을 지정합니다. 양식의 선택 가능 POV는 사용자가 멤버를 선택할 수 있는 차원으로 구성됩니다.

POV 막대의 임의 차원에 대해 멤버를 선택하여 배경 POV 멤버를 지정합니다. 멤버 목록을 선택하는 경우 이 목록은 데이터 양식 사용자가 멤버를 선택할 수 있는 선택 가능 POV로 사용됩니다. 3개 이상의 항목과 2개 이상의 멤버 또는 멤버 목록을 선택할 수 없습니다.

동적 POV 멤버 목록을 정의한 경우 이 목록을 사용하여 선택한 차원에 적합한 멤버를 빠르게 선택할 수 있습니다. [멤버 선택] 대화상자의 맨아래에, 선택한 차원에 적합한 동적 POV 멤버 목록이 @POV 접미어와 함께 표시됩니다.

POV 사용에 적용되는 규칙은 다음과 같습니다.

- 멤버나 멤버 목록을 선택하지 않고 차원이 행이나 열에 사용되지 않는 경우 차원의 초기 값으로 사용자 POV의 멤버가 사용됩니다.
- 차원의 백그라운드 POV와 선택 가능 POV에 값을 지정했는데 백그라운드 POV의 멤버가 선택 가능 POV의 멤버 목록에 없는 경우 선택 가능 POV의 멤버 목록에서 첫 번째 멤버가 백그라운드 POV 멤버로 사용됩니다.

- 사용자가 계층의 모든 멤버에 대해 작업할 수 있으려면 멤버 계층에 대한 전체 액세스 권한이 있어야 합니다. 예를 들어 사용자가 사용자정의 4 멤버에 모두 액세스할 수 있도록 하려면 상위 엔티티(이 경우 CustomTop)에 대한 액세스를 활성화해야 합니다.

양식 디자이너 그리드에는 행과 열이 포함됩니다. 기본적으로 새 양식에서 시나리오 차원은 그리드 열에 사용되고 기간 차원은 그리드 행에 사용됩니다. 양식에 열 또는 행을 추가한 다음 POV 막대의 차원을 그리드 행과 열로 끌어 놓아 데이터 양식 레이아웃을 정의할 수 있습니다. 동일한 행 또는 열에 여러 차원을 끌어 놓는 경우 차원 순서를 조정할 수 있습니다. 차원을 누르는 경우 [멤버 선택] 페이지에서 그리드 또는 POV에 포함할 멤버를 선택할 수 있습니다.

그리드에서 행 또는 열을 선택하는 경우 해당 행 및 열 옵션이 [양식 세부정보] 등록정보 창에 표시됩니다. 행 또는 열에 계산 표현식을 포함할 수 있습니다. 이 경우 행 또는 열에 차원을 끌어 놓을 수 없으며 등록정보 창을 사용하여 행 또는 열에 계산 표현식을 입력해야 합니다. 행에 사용되는 차원은 열에 사용할 수 없고 그 반대의 경우도 마찬가지입니다.

## 양식 세부정보 지정

[양식 세부정보] 섹션에서는 데이터 양식에 대한 그리드, 인쇄, 표시, 제외 옵션 등의 양식 등록정보를 지정합니다. 데이터 양식의 사용자에게 대한 지침을 포함하고 데이터 양식에 사용할 수 있는 주문형 규칙을 지정할 수 있습니다.

### 주:

[양식 세부정보] 섹션의 항목은 기본 설정을 변경하는 경우에만 스크립트에서 업데이트됩니다. 기본 설정을 변경하지 않고 그대로 두면 이 항목의 키워드가 스크립트에 표시되지 않습니다.

각 옵션에 적합한 값은 [데이터 양식 스크립트 사용](#)을 참조하십시오.

## 데이터 양식에 대해 주문형 규칙 지정

데이터 양식에서 실행할 수 있는 주문형 규칙을 생성할 수 있습니다. 주문형 규칙은 계산 서브세트만 실행하여 데이터 양식에서 결과를 빠르게 보려는 경우에 유용합니다. 예를 들어 특정 주문형 규칙을 실행하도록 설정된 데이터 양식에서 작업하는 경우 데이터를 입력한 다음 주문형 규칙을 실행하여 계산 결과를 빠르게 확인할 수 있습니다.

애플리케이션 규칙 파일에서 주문형 규칙을 생성합니다. 규칙은 새 서버루틴에서 생성되며 OnDemand 접두어로 식별됩니다(예: OnDemand\_Calculation). [규칙 파일 생성](#)을 참조하십시오.

데이터 양식에 대해 주문형 규칙을 지정하려면 다음을 수행합니다.

1. **통합, 문서** 순으로 선택합니다.
2. **새로 작성, 데이터 양식** 순으로 누르거나 **작업, 새로 작성, 데이터 양식** 순으로 선택합니다. 기본적으로 양식 디자이너가 표시됩니다.

### 팁:

[디자이너] 페이지가 자동으로 표시되지 않는 경우 **디자이너**를 누르거나 **작업, 디자이너** 순으로 선택합니다.

3. 양식 세부정보 패널의 주문형 규칙에서 다음 방법 중 하나를 사용합니다.
  - 규칙 이름을 알고 있는 경우 텍스트 상자에 심표로 구분된 목록으로 입력합니다.
  - 사용가능 규칙을 검색하려면 텍스트 상자 옆에 있는 **편집** 버튼을 누른 다음 **사용가능 규칙** 팝업 대화상자에서 양식에 사용할 규칙을 하나 이상 선택하고 **확인**을 누릅니다.
4. **확인**을 누릅니다.
5. 데이터 양식을 저장합니다.

## 양식 행 및 열 옵션 지정

행과 열에 대해 지정한 차원 요소는 배경 POV 또는 선택 가능 POV에 설정된 요소를 재정의합니다. 행 또는 열당 하나의 멤버 목록만 지정할 수 있습니다.

각 옵션에 적합한 값은 [데이터 양식 스크립트 사용](#)을 참조하십시오.

양식에 100개 이상의 행이 필요한 경우 [스크립트] 탭을 사용하여 추가 행을 지정할 수 있습니다. 사용할 구문에 대한 자세한 내용은 [Rn](#)을 참조하십시오.

양식에 24개 이상의 열이 필요한 경우 [스크립트] 탭을 사용하여 추가 열을 지정할 수 있습니다. 사용할 구문에 대한 자세한 내용은 [Cn](#)을 참조하십시오.

또한 행과 열에 대해 계산된 행 또는 열과 계산 표현식을 지정할 수 있습니다. [계산된 행] 옵션을 선택하면 선택된 행 또는 열이 계산된 행과 열이 되며 계산 표현식을 입력해야 합니다.

### 주:

[기타]를 사용하여 [옵션] 섹션에 표시되지 않는 Blank, ReadOnly 등의 행 또는 열 키워드에 대한 구문을 지정할 수 있습니다. 또한 스크립트의 행 또는 열 정의에 사용되지 않거나 부적합한 구문이 포함되어 있는 경우 해당 구문이 [기타]에 표시됩니다.

## 양식 머리글 지정

[머리글] 섹션에서는 각 차원에 대한 머리글 등록정보(예: 레이블 표시, 설명 표시 또는 둘 다), 스타일, 멤버 및 멤버 목록에 대한 레이블 또는 설명의 최대 길이 및 길이 고정 여부를 지정할 수 있습니다. 각 차원에 대해 다른 길이를 지정할 수 있습니다.

[머리글] 섹션에는 다음 두 가지 용도로 사용되는 [기타] 텍스트 상자도 있습니다.

- [HeaderOption](#) 키워드에 대해 구문을 추가한 경우 [기타]에서 해당 구문을 지정할 수 있습니다.
- 차원의 [HeaderOption](#) 키워드에 부적합한 구문이 양식에 포함되어 있는 경우 [기타] 텍스트 상자에 해당 구문이 표시됩니다.

각 옵션에 적합한 값은 [데이터 양식 스크립트 사용](#)을 참조하십시오.

## 데이터 양식 스크립트 사용

스크립트 기능을 사용하여 스크립트를 보고 수정하고 인쇄할 수 있습니다. [스크립트] 페이지의 텍스트 상자에 구문을 입력합니다.

데이터 양식 스크립트를 생성하는 경우 키워드, 값 및 옵션의 세 가지 구문 요소 유형을 사용할 수 있습니다. 키워드는 스크립트의 개별 라인에 있으며 등호 왼쪽에 놓입니다. 값은 등호 바로 뒤에 놓여 라인을 완성합니다. 옵션은 스크립트의 한 라인에 추가할 수 있으며 각각 심표로 구분됩니다.



주:

값이 필요한 경우 옵션 앞에 지정해야 합니다. 옵션은 필수가 아니며 순서에 관계없이 배치할 수 있습니다.

샘플 데이터 양식 스크립트는 Oracle Hyperion Financial Management용 샘플 애플리케이션과 함께 설치됩니다. 이러한 파일은 Financial Management를 설치한 디렉토리의 Sample Applications 폴더에 있습니다.



주:

데이터 입력 양식 스크립트 요소는 대/소문자를 구분하지 않습니다.

표 7-1 데이터 양식 스크립트 구문

스크립트 구문	설명
AddMember	행 정의에 사용하여 사용자가 이전에 데이터가 없었거나 0을 포함하여 양식에서 제외된 멤버의 데이터를 추가할 수 있도록 합니다. 이 옵션은 양식에 아이콘을 추가하며, 아이콘을 누르면 사용자가 양식에 추가할 멤버를 선택할 수 있습니다.
BackgroundPOV	양식의 배경 차원 멤버를 지정하는 데 사용합니다.
Blank	비어 있는 행, 열 또는 셀을 양식에 삽입하는 데 사용합니다.
Cn	양식의 각 행을 정의하는 데 사용합니다.
CalcByRow	셀에 교차하는 열 계산이 있을 경우 행 계산을 사용할지 여부를 지정하는 데 사용합니다.
Cell_Link	행 정의에 <a href="#">Link</a> 와 함께 사용하여 다른 데이터 입력 양식에 링크합니다.
CellText	행 또는 열이 셀 텍스트 입력을 허용하는지 여부를 지정하는 데 사용됩니다.

표 7-1 (계속) 데이터 양식 스크립트 구문

스크립트 구문	설명
CustomHeader	<p>멤버 레이블 또는 설명 대신 표시할 사용자정의 머리글 텍스트를 지정하는 데 사용됩니다. 행이나 열 정의에 사용합니다.</p> <p>다음과 같은 키워드는 사용자정의 머리글에 사용할 수 없습니다.</p> <ul style="list-style-type: none"> <li>• &lt;pre&gt;</li> <li>• &lt;textarea&gt;</li> <li>• &lt;script&gt;</li> <li>• &lt;javascript&gt;</li> <li>• &lt;jscript&gt;</li> <li>• &lt;vbs&gt;</li> <li>• &lt;vbscript&gt;</li> <li>• &lt;XonX=X&gt; 등의 문자열. 여기서 X = 임의의 문자열입니다.</li> </ul>
CustomHeaderStyle	<p>행이나 열 머리글에 사용자정의 스타일 속성을 지정하는 데 사용합니다.</p>
DynamicPOV	<p>더 이상 제공되지 않습니다. 사용하지 마십시오.</p>
FormInputBoxLength	<p>양식의 입력 상자 너비를 지정하는 데 사용합니다.</p>
FormNumDecimals	<p>양식의 소수 자릿수를 지정하는 데 사용합니다. 이 키워드는 셀 통화의 소수 자릿수 설정을 재정의합니다. 행, 열 또는 셀에 대해 이 설정을 대체하려면 <b>NumDecimals</b>를 사용합니다.</p>
FormRowHeight	<p>양식의 모든 행 높이를 지정하는 데 사용합니다.</p>
FormScale	<p>양식의 스케일을 지정하는 데 사용합니다.</p>
HeaderOption	<p>양식에서 차원 머리글이 표시되는 방식을 지정하는 데 사용합니다. 레이블 및/또는 설명을 표시하거나, 스타일 속성을 설정하거나, 최대 또는 고정 너비를 설정합니다.</p>
Instructions	<p>HTML 형식의 텍스트와 링크로 지침을 생성하는 데 사용합니다.</p>
LineItemDetailSinglePeriod	<p>라인 항목 세부정보를 선택한 셀에 대해서만 표시할지 또는 모든 기간에 대해 표시할지를 지정하는 데 사용합니다.</p>
Link	<p><b>Cell_Link</b>와 함께 사용하여 다른 데이터 입력 양식에 링크합니다.</p>
MaxCells	<p>데이터 양식의 최대 셀 수를 지정하는 데 사용합니다.</p>
MaxColsForSparseRetrievalMethod	<p>최소 양식의 성능을 최적화하는 데 사용합니다. 포함된 열 수가 10개를 초과하는 양식에 사용합니다.</p>
NoSuppress	<p>하나 이상의 행이나 열에 대해 제외를 해제하는 데 사용합니다. 이 설정은 <b>SuppressInvalidRows</b>, <b>SuppressNoDataRows</b>, <b>SuppressZeroRows</b>, <b>SuppressInvalidCols</b>, <b>SuppressNoDataCols</b>, <b>SuppressZeroCols</b> 등 양식의 다른 숨김 설정을 대체합니다.</p>

표 7-1 (계속) 데이터 양식 스크립트 구문

스크립트 구문	설명
NumDecimals	행, 열 또는 셀의 소수 자릿수를 지정하는 데 사용합니다. 이 키워드는 셀 통화의 소수 자릿수 설정과 양식의 소수 자릿수 설정인 <a href="#">FormNumDecimals</a> 를 대체합니다.
OnDemandRules	데이터 양식에 사용할 수 있는 주문형 규칙을 지정하는 데 사용합니다.
Override	하나 이상의 행이나 열에 대해 다른 POV 또는 계산을 지정하거나, 스타일 속성을 추가하거나 스케일을 설정하는 데 사용합니다. 행이나 열 정의에 사용합니다.
HideInPov	POV에서 차원을 숨길지 여부를 지정하는 데 사용합니다.
POVOrder	POV에서 차원 이름의 순서를 지정하는 데 사용합니다.
PrintNumDataColsPerPage	각 페이지에 인쇄할 열 수를 지정하는 데 사용합니다.
PrintNumRowsPerPage	각 페이지에 인쇄할 행 수를 지정하는 데 사용합니다.
PrintRepeatHeadersonAllPages	각 페이지에 머리글을 인쇄하는 데 사용합니다.
Rn	양식의 각 행을 정의하는 데 사용합니다.
ReadOnly	읽기 전용 행, 열 또는 셀을 지정하는 데 사용합니다.
ReportDescription	양식에 대한 설명을 지정하는 데 사용합니다. 앰퍼샌드(&)는 설명에 사용할 수 없습니다.
ReportLabel	양식의 레이블을 지정하는 데 사용합니다. 다음과 같은 문자는 데이터 양식 레이블에 지원되지 않습니다. 앰퍼샌드(&), 별표(*), 백슬래시(\), 콜론(:), 침표(.), 중괄호({ }), 큰따옴표(" "), 슬래시(/), 보다 작음 및 보다 큼(< >), 숫자 기호(#), 괄호(), 마침표(.), 파이프( ), 더하기(+), 물음표(?), 세미콜론(; ) 및 후행 밑줄(_).
ReportSecurityClass	양식의 보안 클래스를 지정하는 데 사용합니다.
ReportType	양식 유형을 설정하는 데 사용합니다. 값을 <a href="#">WebForm</a> 으로 설정해야 합니다.
RowHeaderPct	양식의 총 너비를 참조하여 행 머리글 너비의 크기를 조정하는 데 사용됩니다.
SCalc	행, 열 또는 셀에 대해 서버측 계산을 지정하는 데 사용합니다.
Scale	행, 열 또는 셀의 스케일을 지정하는 데 사용합니다. 적합한 값은 -12에서 12 사이입니다. 이 설정은 양식 스케일 설정을 재정의합니다. <a href="#">FormScale</a> 을 참조하십시오.
SelectablePOVList	양식의 선택 가능 차원 멤버를 지정하는 데 사용합니다.
ShowDescriptions	차원 멤버에 대한 설명을 표시하는 데 사용합니다.

표 7-1 (계속) 데이터 양식 스크립트 구문

스크립트 구문	설명
ShowLabels	차원 멤버의 레이블을 표시하는 데 사용합니다.
String	열, 행 또는 셀에 텍스트 문자열을 추가하는 데 사용합니다.
Style	행, 열, 셀 또는 차원 머리글의 스타일 속성을 지정하는 데 사용합니다.
SuppressColHeaderRepeats	반복된 열 머리글이 표시되지 않도록 하는 데 사용합니다.
SuppressInvalidCols	부적합한 셀이 열에 표시되지 않도록 하는 데 사용합니다.
SuppressInvalidRows	부적합한 셀이 행에 표시되지 않도록 하는 데 사용합니다.
SuppressNoDataCols	데이터가 없는 열이 표시되지 않도록 하는 데 사용합니다.
SuppressNoDataRows	데이터가 없는 행이 표시되지 않도록 하는 데 사용합니다.
SuppressRowHeaderRepeats	반복된 행 머리글이 표시되지 않도록 하는 데 사용합니다.
SuppressZeroCols	0이 포함된 열이 표시되지 않도록 하는 데 사용합니다.
SuppressZeroRows	0이 포함된 행이 표시되지 않도록 하는 데 사용합니다.

## AddMember

행 정의 내에서 이 옵션을 사용하여 양식에 아이콘을 추가합니다. 이 아이콘을 누르면 사용자가 양식에 추가할 멤버를 선택할 수 있습니다. 새 멤버가 양식에 추가되면 사용자가 해당 멤버에 대한 데이터를 입력할 수 있습니다.

AddMember 옵션은 계정, ICP 및 사용자정의 차원에 사용할 수 있습니다.



주:

**SuppressNoDataRows** 키워드를 사용하거나 양식에서 해당 확인란을 선택하여 NoData 셀을 숨길 수 있습니다.

### 구문

```
AddMember:MemberList
```

*MemberList*를 사용자가 양식에 멤버를 추가하는 데 사용할 수 있는 목록 이름으로 바꿉니다.



**▲ 주의:**

AddMember 라인당 하나의 멤버 목록만 참조할 수 있습니다.

이 옵션을 사용하려면 다음 두 개의 행 정의가 필요합니다.

- 요약 행에는 멤버 목록의 합계가 표시되고 AddMember 옵션이 있습니다.

**✎ 주:**

요약 행은 제외할 수 없습니다.

- 목록 행에는 요약 행에 지정된 것과 동일한 POV가 있지만 멤버가 추가되는 차원 목록이 있습니다. 이 목록에는 요약 행 목록의 멤버가 하나 이상 포함되어야 합니다.

**✎ 주:**

목록 행은 요약 행 앞이나 뒤에 배치할 수 있습니다.

**예**

```
R1=A#SalesInterco.I{ [Base] }
```

```
R2=A#SalesInterco.I#[ICP Entities],  
AddMember:I{ [Base] }
```

이 예에서 행 1은 내부거래 파트너 차원에 대한 [Base] 멤버 목록을 지정하고, 행 2는 AddMember 아이콘에 사용할 멤버 목록과 멤버 목록의 합계를 지정합니다.

**✎ 주:**

양식의 여러 행에서 AddMember 옵션을 사용할 수 있지만 각 요약 행에 고유한 목록 행이 있어야 합니다.

## BackgroundPOV

이 키워드를 사용하여 양식의 백그라운드 차원 멤버를 지정합니다. 백그라운드 POV에 지정되지 않은 차원은 동적으로 간주되며 양식을 열 때 사용자 POV에서 가져옵니다.

사용자가 계층의 모든 멤버에 대해 작업할 수 있으려면 멤버 계층에 대한 전체 보안 액세스 권한이 있어야 합니다. 예를 들어 사용자가 사용자정의 4 멤버에 모두 액세스할 수 있도록 하려면 상위 엔티티(이 경우 CustomTop)에 대한 액세스를 활성화해야 합니다.

 주:

차원의 **SelectablePOVList**를 기준으로 백그라운드 POV의 멤버가 부적합한 경우 기본적으로 목록의 첫번째 멤버가 설정됩니다.

구문

BackgroundPOV=S#Scenario.W#View.E#Parent.Entity.V#Value.A#Account.I#ICP.  
C1#Custom1.C2#Custom2.C3#Custom3.C4#Custom4

표 7-2 BackgroundPOV 키워드 구문

매개변수	설명
Scenario	적합한 시나리오 멤버의 이름입니다.
View	적합한 뷰입니다.
Parent	적합한 상위 멤버의 이름입니다. 이 매개변수는 선택 사항입니다.
Entity	적합한 엔티티 멤버의 이름입니다.
Value	적합한 값 멤버의 이름입니다.
Account	적합한 계정 멤버의 이름입니다.
ICP	적합한 ICP 멤버의 이름입니다.
Custom	적합한 사용자정의 멤버의 이름입니다.

 주:

백그라운드 POV의 차원 멤버를 모두 지정할 필요는 없습니다. 멤버를 지정하지 않은 차원에는 사용자 POV의 차원 멤버가 사용됩니다.

## 예

```
BackgroundPOV=S#Actual.Y#2014.P#January.W#Periodic.V#<Entity
Currency>.A#Sales.I#[ICPNone].C4#[None]
```

## Blank

이 옵션을 사용하여 양식에 빈 행, 열 또는 셀을 지정합니다. 행 또는 열 정의나 셀 재정의 정의에 이 옵션을 사용합니다. 빈 행, 열 또는 셀이 비어 있고 읽기 전용이며 셀에 데이터나 POV가 없기 때문에 셀에서 컨텍스트 메뉴가 비활성화되어 있습니다. Blank 옵션은 서버측 계산으로 간주되므로 SCalc가 사용되는 모든 곳에서 사용할 수 있습니다.

## 예

```
C4=Blank
```

## Cn

이 키워드를 사용하여 양식의 열을 정의합니다. C1, C2, C3 등의 키워드는 지정된 순서대로 각 열의 정의를 제공합니다. 열 ID는 1부터 시작하여 순서대로 지정되어야 합니다.

열에 지정된 차원 요소는 [BackgroundPOV](#) 또는 [SelectablePOVList](#)에 설정된 요소를 대체합니다. 열 정의에 멤버 목록과 시스템 목록을 사용할 수 있습니다.



## 주:

열당 하나의 목록을 사용할 수 있습니다.

열 정의 내에서 다음과 같은 값과 옵션을 사용할 수 있습니다.

- [Blank](#)
- [Calc1](#)
- [CellText](#)
- [CustomHeader](#)
- [NumDecimals](#)
- [Override](#)
- [ReadOnly](#)
- [SCalc](#)
- [Scale](#)
- [String](#)
- [Style](#)

## 구문

*Cn=CalcExpression*

*Cn=POVExpression*

**표 7-3 Columns 키워드 구문**

매개변수	설명
n	열 번호입니다.
CalcExpression	S <i>Calc</i> , Blank 또는 String을 사용합니다. S <i>Calc</i> , Blank 및 String을 참조하십시오.
POVExpression	적합한 차원 교차나 멤버 목록입니다.

## 예

C1=S#Actual.P#July,CustomHeader:Actual\_July

C2=S#Actual.P#August, (Override 2,3, P#July)

C3=S#Budget.P#September

C4=S*Calc*(col(1)+col(3)),numdecimals:4,scale:1,readonly

C5=Blank

C6=S#Budget.P#October,Style:font-style:bold

C7=C1{TotalProducts.[Hierarchy]}

## Calc1

이 값은 사용되지 않습니다. 이 키워드를 사용하는 양식이 있는 경우 S*Calc*를 사용하도록 수동으로 편집해야 합니다.

## CalcByRow

기본 계산 순서(열 먼저)를 행 먼저로 변경하는 데 사용되는 키워드입니다. 이 키워드는 전체 양식에 적용됩니다.

### 구문

```
CalcByRow=Boolean
```

행 계산이 사용되는 경우 *Boolean*이 *True*이고, 열 계산이 사용되는 경우 *False*입니다.

## 예

```
ReportType=WebForm
```

```
ReportLabel=CalcByCol
```

```
ReportDescription=Demonstrate CalcByRow
```

```
BackgroundPOV=S#Actual.Y#2014.P#January.w#<Scenario  
View>.E#UnitedStates.Connecticut.V#USD.A#Sales.I#[ICP  
NONE].C1#Golfballs.C2#Customer2.C3#[None].C4#[None]
```

```
C1=S#Actual
```

```
C2=S#Budget
```

```
C3=SCalc(Col(2)-Col(1)),CustomHeader:Variance
```

```
C4=SCalc(Col(3)/Col(1)),CustomHeader:Variance %
```

```
R1=A#Sales
```

```
R2=A#Purchases
```

```
R3=A#OtherCosts
```

```
R4=SCalc(Row(1)+Row(2)+Row(3)),CustomHeader:Total
```

```
ShowLabels=True
```

```
CalcByRow=False
```

## Cell\_Link

이 옵션을 사용하여 다른 데이터 양식에 대한 링크를 지정합니다. 링크는 행 정의의 참조입니다. [Link](#)와 함께 사용합니다. 최대 64개의 링크(Link1 - Link64)를 지정할 수 있습니다. 링크에 순차적으로 번호를 매길 필요는 없습니다.

**주:**

링크된 양식 이름은 대소문자를 구분합니다.

### 구문

`Cell_Linkx``Linkx=FormName`

x를 링크에 지정할 번호로 바꾸고 *FormName*을 링크할 양식 이름으로 바꿉니다.

### 예

`R1=A#Salaries, Cell_Link1``Link1=Dynamic`

## CellText

이 옵션을 사용하여 행, 열 또는 셀에 대해 셀 텍스트를 표시합니다. 셀 텍스트 항목의 처음 69자는 항목의 제목으로 간주됩니다.

**주:**

셀 텍스트가 포함된 데이터 양식을 추출하는 경우 셀 텍스트 항목의 제목(처음 69자)만 추출됩니다.

### 구문

`CellText:<cell text label>`

### 예

`R1=A#Salaries, CellText:[Default]``R2=A#Salaries, CellText:Validation`

## CustomHeader

이 옵션을 사용하여 열 또는 행의 사용자정의 머리글을 지정합니다. 이 옵션은 계산된 열이나 행이 있으며 특정 공식을 숨기고 차이 등의 설명으로 바꾸려는 경우에 유용합니다. 행이나 열에 차원을 중첩한 경우 차원뿐 아니라 행이나 열의 전체 머리글에 사용자정의 머리글이 적용됩니다.

세미콜론을 구분자로 사용하여 후속 셀의 사용자정의 머리글을 지정할 수 있습니다. 예를 들어 다음 구문은 행에 있는 세 개의 차원 머리글을 사용자정의 머리글인 Scenario, Year 및 Month로 바꿉니다.

```
R2=S#Actual.Y#2014.P#January,CustomHeader:Scenario;Year;Month
```

일부 머리글 셀만 바꾸고 다른 셀은 바꾸지 않으려면 마침표(.)를 사용하여 원래 머리글이 표시되어야 함을 나타냅니다. 기간을 제외하여 머리글을 숨길 수도 있습니다. 예를 들어 다음 구문은 첫 번째 차원에 원래 머리글을 표시하고, 두 번째 차원에서 머리글을 숨기고, 세 번째 차원에 원래 머리글을 표시합니다.

```
R2=S#Actual.Y#2014.P#January,CustomHeader:.;;
```

### ▲ 주의:

계산된 행이나 열에 CustomHeader 옵션을 사용하는 경우 머리글에서 첫 번째 셀의 텍스트만 설정할 수 있습니다.

## 구문

```
CustomHeader:HeaderName
```

HeaderName을 사용할 머리글로 바꿉니다.

### ✎ 주:

십표(.), 콜론(:) 또는 앰퍼샌드(&)는 사용자정의 머리글에 사용할 수 없습니다.

## 예

다음 예에서는 열 3의 사용자정의 머리글을 Variance로 설정합니다.

```
C3=Scalc(col(1)-col(2)),customheader:Variance
```

## CustomHeaderStyle

이 옵션을 사용하여 행 또는 열 머리글에 사용자정의 스타일 속성을 지정합니다. 이 옵션은 모든 머리글의 차원 하나가 아니라 행 또는 열 머리글 전체에 적용된다는 점에서 HeaderOption 키워드의 Style 옵션과는 다릅니다. 충돌이 발생할 경우 CustomHeaderStyle 옵션이 HeaderOption:Style 키워드보다 우선합니다. CustomHeaderStyle을 사용하는 대신 두 스타일을 혼합하려면 아래 예와 같이 CustomHeaderStyle 옵션 앞에 세미콜론을 삽입합니다. [Style](#)을 참조하십시오.



### 구문

이 옵션은 W3C에서 지원되는 표준을 사용합니다.

```
CustomHeaderStyle:Property:Value
```

### 예

```
C1=S#Actual,CustomHeaderStyle:font-style:italic;font-family:arial;font-size:12px;font-color:red
```

## DynamicPOV

이 키워드는 더 이상 제공되지 않습니다.

DynamicPOV 키워드가 포함된 양식이 있는 경우 이 사용 중단을 고려하려면 다음 단계를 사용합니다.

1. BackgroundPOV 키워드에서 DynamicPOV 키워드로 지정된 차원을 모두 제거합니다.
2. DynamicPOV 키워드를 삭제합니다.

## FormInputBoxLength

이 키워드를 사용하여 입력 상자 너비를 지정할 수 있습니다. 기본값은 20자 너비입니다.



주:

이 키워드는 입력 상자에 입력할 수 있는 문자 수를 결정하지 않습니다.

### 구문

```
FormInputBoxLength=InputLength
```

InputLength를 입력 상자 너비의 문자 수로 바꿉니다.

### 예

```
FormInputBoxLength=20
```

## FormNumDecimals

이 키워드를 사용하여 전체 양식의 소수 자릿수를 지정합니다. 이 키워드를 지정하면 셀에 설정된 소수 자릿수가 재정의됩니다. 이 키워드를 지정하지 않으면 셀의 소수 자릿수가 사용됩니다.

NumDecimals를 사용하여 열, 행 또는 셀의 소수 자릿수 설정을 대체할 수 있습니다.

## 구문

```
FormNumDecimals=Decimals
```

*Decimals*를 0에서 9 사이의 값으로 바꿉니다.



### 주:

값 9를 사용하면 최대 정밀도가 사용되므로 실제로 소수점 뒤에 최대 14자리가 포함됩니다.

## 예

```
FormNumDecimals=0
```

## FormRowHeight

이 키워드를 사용하여 양식의 모든 행 높이를 지정합니다. 기본값은 16px입니다.

## 구문

```
FormRowHeight=Pixels px
```

*Pixels*를 행 높이의 픽셀 수로 바꿉니다.



### 주의:

*Pixels* 뒤에 *px*를 포함해야 합니다. *px*를 생략하면 양식을 인쇄할 때 행과 열이 정확하게 정렬되지 않을 수 있습니다.

## 예

```
FormRowHeight=16 px
```

## FormScale

이 키워드를 사용하여 양식의 기본 스케일링을 지정합니다.

양식에 이 키워드를 지정하면 엔티티 통화의 스케일이 재정의됩니다. 그렇지 않으면 엔티티에 지정된 통화에 대해 정의된 스케일링이 사용됩니다.

행 또는 열 정의나 셀 대체에 **Scale** 옵션을 사용하여 양식 스케일 설정을 대체할 수 있습니다.

## 구문

```
FormScale=n
```

*n*을 -12에서 12 사이의 값으로 바꿉니다.

## 예

```
FormScale=0
```

## HeaderOption

이 키워드를 사용하여 각 차원의 머리글 기본값을 지정할 수 있습니다. 현재 지원되는 옵션은 다음과 같습니다.

- **Length** - 행 머리글의 최대 길이입니다. 숫자이거나, 최대 길이와 최소 길이를 동일하게 설정하려는 경우 "Fixed"란 단어일 수 있습니다.



주:

행 머리글이 지정된 길이보다 길면 줄임표로 잘립니다.

- **ShowDescription** - 멤버 설명을 표시합니다.
- **ShowLabel** - 멤버 레이블을 표시합니다.
- **Style**

## 구문

```
HeaderOptionDimension=Length:n
```

```
HeaderOptionDimension=ShowDescription
```

```
HeaderOptionDimension=ShowLabel
```

```
HeaderOptionDimension=Style:Property:Value;Property:Value...
```

*Dimension*을 행 차원 이름으로, *n*을 머리글 길이나 "Fixed"로, *Property*를 스타일 등록정보로, *Value*를 등록정보 값으로 바꿉니다. [Style](#)을 참조하십시오.

**예**

```
HeaderOptionPeriod=Length:4
```

```
HeaderOptionScenario=ShowDescription
```

```
HeaderOptionsAccount=ShowLabel,Style:font-style:italic
```

## HideInPov

이 키워드를 사용하여 각 차원에 대해 HideInPOV 옵션을 지정합니다. 이 값이 있으면 데이터 양식의 POV 막대에 차원이 표시되지 않습니다. 이 값을 찾을 수 없는 경우 POV 막대에 차원이 표시됩니다.

**구문**

```
HideinPOV=Dimension
```

차원 이름은 단축 이름이나 긴 이름일 수 있으며 심표로 구분된 목록에 나열됩니다.

**주:**

차원이 선택 가능한 POV의 일부인 경우 선택 가능한 모든 차원이 POV 막대에 표시되므로 차원이 무시됩니다.

**예**

다음 예에서 시나리오, 연도 및 기간은 데이터 양식을 열 때 POV 막대에서 숨겨집니다.

```
HideinPov=S,Y,P
```

## Instructions

이 키워드를 사용하여 양식 사용자에게 대한 지침을 지정할 수 있습니다. 이 키워드를 지정하지 않으면 지침 창이 열릴 때 "이 양식에 정의된 세부 지침이 없습니다"라는 메시지가 표시됩니다.

**구문**

```
지침=HTMLInstructions
```

*HTMLInstructions*를 HTML 형식 텍스트와 링크로 바꿉니다.

**예**

```
Instructions=Please enter your cost center budgets for the year.Any questions, please contact the Budget Administrator.
```

## LineItemDetailSinglePeriod

이 키워드를 사용하여 라인 항목 세부정보를 선택한 셀에 대해서만 표시할지 또는 모든 입력 기간에 대해 표시할지를 지정합니다. 기본값은 `True`이며, 선택한 셀에 대해서만 세부정보가 표시됩니다.

### 구문

```
LineItemDetailSinglePeriod=Boolean
```

선택한 셀에 대한 라인 항목 세부정보를 표시하려면 `Boolean`을 `True`로 바꾸고, 모든 입력 기간에 대한 라인 항목 세부정보를 표시하려면 `False`로 바꿉니다.

### 예

```
LineItemDetailSinglePeriod=True
```

## Link

이 키워드를 사용하여 다른 데이터 양식에 대한 링크를 지정합니다. 링크는 행 정의의 참조입니다. [Cell\\_Link](#)와 함께 사용합니다. 최대 64개의 링크(Link1 - Link64)를 지정할 수 있습니다. 링크에 순차적으로 번호를 매길 필요는 없습니다.

### 주:

링크된 양식 이름은 대소문자를 구분합니다.

### 구문

```
Cell_Linkx
```

```
Linkx=FormName
```

*x*를 링크에 지정할 번호로 바꾸고 *FormName*을 링크할 양식 이름으로 바꿉니다.

### 예

```
R1=A#Salaries, Cell_Link1
```

```
Link1=Dynamic
```

## MaxCells

이 키워드는 데이터 양식에 허용되는 셀의 최대 개수를 지정합니다. 양식의 셀 수가 `MaxCells` 값을 초과하면 오류가 발생합니다. 기본값은 `25000`입니다.

## 구문

```
MaxCells=n
```

$n$ 을 양식의 셀 수로 바꿉니다.

### 주:

값은 계산된 셀을 비롯하여 양식에 표시되는 셀 수입니다. 제외된 셀은 포함되지 않습니다.

## 예

```
MaxCells=500
```

## MaxColsForSparseRetrievalMethod

### 주:

이 키워드는 사용되지 않습니다. 이 키워드를 사용하는 스크립트가 있는 경우 양식은 정확하게 로드되지만 값을 편집할 수 없습니다.

이 키워드를 사용하여 희소 데이터 양식의 성능을 최적화하도록 양식의 열 수를 지정합니다. 열 수가 10개를 초과하는 양식에 이 키워드를 지정합니다. 양식의 열 수가 10개 이하이면 자동으로 최적화됩니다.

## 구문

```
MaxColsForSparseRetrievalMethod=n
```

$n$ 을 양식의 열 수로 바꿉니다.

## 예

```
MaxColsForSparseRetrievalMethod=11
```

## NoSuppress

이 옵션을 사용하여 하나 이상의 행이나 열에 대해 제외를 해제합니다. 따라서 양식 제외 옵션에 관계없이 행이나 열이 표시됩니다. 행 또는 열 정의에 이 옵션을 사용합니다.

## 예

```
R4=A#Inventory,NoSuppress
```

## NumDecimals

이 옵션을 사용하여 계산되었거나 계산되지 않은 행 또는 열에 대해 표시하거나 셀 재정의에 표시할 소수 자릿수를 지정합니다. 이 키워드를 지정하면 셀에 설정되었거나 `FormNumDecimals`로 설정된 소수 자릿수가 대체됩니다.

### 구문

```
NumDecimals:n
```

*n*을 0에서 9 사이의 값으로 바꿉니다.



#### 주:

값 9를 사용하면 최대 정밀도가 사용되므로 실제로 소수점 뒤에 최대 14자리가 포함됩니다.

### 예

```
C4=A#Inventory,NumDecimals:1
```

## OnDemandRules

데이터 양식에 사용할 수 있는 주문형 규칙을 지정하려면 이 옵션을 사용합니다. 심표로 구분된 규칙 이름 목록을 사용합니다. [데이터 양식에 대해 주문형 규칙 지정](#)을 참조하십시오.

### 예

```
OnDemandRules=Calculation,Tax,Tax2
```

## Override

이 옵션을 사용하여 하나 이상의 연속된 열이나 행에 대해 다른 POV 차원 멤버, 공식 계산 또는 텍스트를 지정하거나 스타일을 변경합니다. 행 또는 열 정의에 이 옵션을 사용합니다.



#### 주:

연속하지 않는 셀을 재정의하려면 양식 옵션의 [기타] 필드에 재정의의 입력할 수 있습니다. 각 재정의의 심표로 구분하여 재정의의 여러 개 입력할 수 있습니다. 다음 예에서는 세 개의 개별 셀을 재정의합니다.

```
Override(1,1,string("455")),Override(3,3,string("23")),  
Override(5,5,string("2234"))
```

[재정의] 옵션에 멤버 목록을 사용할 수는 없습니다. 한 셀에 여러 개의 재정의의 적용할 수 있으며, 충돌이 있을 경우 마지막 값이 사용됩니다. 대부분의 경우 한 셀의 재정의의 혼합할 수도 있습니다. 예를 들어 행의 재정의에서는 NumDecimals 값을 지정하고, 열의 재정의에서는 Scale 값을 지정하는 경우 동일한 차원이 사용되는 경우를 제외하고 충돌이 발생하지 않습니다. 선행 세미콜론을 사용하는 경우 Style 값을 혼합할 수 있습니다.

Override 키워드 뒤에 세미콜론을 포함하여 교차하는 위치에서 재정의의 혼합할 수도 있습니다. 세미콜론이 없으면 양식에서 충돌할 경우 행 값이 열 값보다 우선하므로 행에 대해 정의된 스타일이 사용됩니다. 행과 열 정의의 재정의의 혼합하려면 행 키워드에 세미콜론을 포함해야 합니다.

## 구문

```
Override(StartCell,EndCell,Override)
```

매개변수	설명
StartCell	재정의의 시작 지점을 나타내는 정수 값입니다. 행에 대해 재정의가 정의된 경우 이 매개변수는 재정의가 적용되는 시작 열을 나타냅니다. 다음 예에서는 재정의가 열 2에서 시작됩니다.  R2=A#Sales, Override(2,3,A#SalesTP)
EndCell	재정의의 종료 지점을 나타내는 정수 값입니다. 위의 예에서는 재정의가 열 3에서 끝납니다.
Override	POV입니다. 예를 들어 다른 계정으로 재정의하려면 A#newacct를 지정합니다. 다른 시나리오와 계정으로 재정의하려면 A#newacct.S#newscenario를 지정합니다. POV 재정의와 함께 또는 단독으로 다음 값이나 옵션을 사용할 수도 있습니다. <ul style="list-style-type: none"> <li>Blank</li> <li>CellText</li> <li>NumDecimals</li> <li>ReadOnly</li> <li>SCalc</li> <li>Scale</li> <li>String</li> <li>Style</li> </ul>

## 예

다음 재정의의 예에서는 기간으로 1월을 사용하고 해당 월의 판매 금액 대신 PriorSales 금액을 사용하여 행 2에 대해 열 2와 3을 재정의합니다.

```
R2=A#Sales, Override(2,3,Y#2014.P#January.A#PriorSales)
```



다음 재정의 예에서는 Sales1, Sales2 및 Sales3의 평균을 공식으로 계산하여 행 2에 대해 열 2와 3을 재정의합니다.

```
R2=A#Sales, Override (2,3,SCalc((A#Sales1+A#Sales2+A#Sales3)/3))
```

다음 재정의 예에서는 읽기 전용 옵션을 사용하여 행 2에 대해 열 2와 3을 재정의합니다.

```
R2=A#Sales, Override (2,3,readonly)
```

## POVOrder

이 키워드를 사용하여 POV에서 차원 이름의 순서를 지정합니다. 차원 이름은 단축 이름이나 긴 이름일 수 있으며 표시되어야 하는 순서대로 심표로 구분된 목록에 나열됩니다. 목록에 지정되지 않은 모든 차원은 목록의 끝에 차원 번호순으로 추가됩니다.

### 예

다음 예에서 POV 막대의 차원 순서는 계정, 기간, 연도, 시나리오입니다.



### 주:

계정 차원이 숨김으로 지정된 경우 기간이 POV 막대에 첫 번째로 표시됩니다.

```
POVOrder=A, P, Y, S
```

## PrintNumDataColsPerPage

이 키워드를 사용하여 각 페이지에 인쇄할 열 수를 지정합니다. 기본값은 6입니다.

사용자는 양식의 프린터 버전에서 이 설정을 재정의할 수 있습니다.



### 주:

이 키워드의 값을 설정하는 경우 행 높이, 사용된 프린터 및 프린터 설정(예: 해상도 및 방향)과 같은 양식의 세부 사항도 고려해야 합니다.

### 구문

```
PrintNumDataColsPerPage=n
```

*n*을 페이지당 인쇄할 열 수로 바꿉니다.

### 예

```
PrintNumColsPerPage=6
```

## PrintNumRowsPerPage

이 키워드를 사용하여 각 페이지에 인쇄할 행 수를 지정합니다. 기본값은 20입니다.  
사용자는 양식의 프린터 버전에서 이 설정을 재정의할 수 있습니다.

### 주:

이 키워드의 값을 설정하는 경우 행 높이, 사용된 프린터 및 프린터 설정(예: 해상도 및 방향)과 같은 양식의 세부 사항도 고려해야 합니다.

### 구문

```
PrintNumRowsPerPage=n
```

*n*을 페이지당 인쇄할 행 수로 바꿉니다.

### 예

```
PrintNumRowsPerPage=20
```

## PrintRepeatHeadersonAllPages

이 키워드를 사용하여 모든 페이지에 머리글을 인쇄할지 여부를 지정합니다. 기본값은 True입니다.

사용자는 양식의 프린터 버전에서 이 설정을 재정의할 수 있습니다.

### 구문

```
PrintRepeatHeadersonAllPages=Boolean
```

*Boolean*을 True 또는 False로 바꿉니다.

### 예

```
PrintRepeatHeadersonAllPages=True
```

## Rn

이 키워드를 사용하여 양식의 행을 정의합니다. R1, R2, R3 등의 키워드는 지정된 순서대로 각 행의 정의를 제공합니다. 행 ID는 1부터 시작하여 순서대로 지정되어야 합니다. 행 정의에 멤버 목록과 시스템 목록을 사용할 수 있습니다.

**주:**

한 행에 여러 목록을 사용할 수 있습니다.

행에 지정된 차원 요소는 [BackgroundPOV](#) 또는 [SelectablePOVList](#)에 설정된 요소를 대체합니다. 행 정의 내에서 다음과 같은 값과 옵션을 사용할 수 있습니다.

- [AddMember](#)
- [Blank](#)
- [Cell\\_Link](#)
- [CellText](#)
- [CustomHeader](#)
- [NoSuppress](#)
- [NumDecimals](#)
- [Override](#)
- [ReadOnly](#)
- [SCalc](#)
- [Scale](#)
- [String](#)
- [Style](#)

### 구문

*Rn=CalcExpression**Rn=POVExpression***표 7-4 Rows 키워드 구문**

매개변수	설명
n	행 번호입니다.
CalcExpression	SCalc, Blank 또는 String을 사용합니다. <a href="#">SCalc</a> , <a href="#">Blank</a> 및 <a href="#">String</a> 을 참조하십시오.
POVExpression	적합한 차원 교차나 멤버 목록입니다.

**예**

```
R1=A#Sales.I#[ICP Entities],AddMember:I{[Base]}
```

```
R2=A#Sales.I{[Base]}
```

```
R3=A#HeadCount.I#[ICP None],NoSuppress
```

```
R4=A#Purchases.I#[ICP None],CustomHeader:ABC
```

```
R5=S#Calc(Row(2)*100),numdecimals:1,scale:0
```

```
R6=A{OperatingIncome.[Descendants]}
```

## ReadOnly

이 옵션을 사용하여 양식에 읽기 전용 행, 열 또는 셀을 지정합니다. 행 또는 열 정의나 셀 재정의 정의에 이 옵션을 사용합니다. 읽기 전용 셀은 다른 모든 셀과 유사하지만 셀 내용을 편집할 수 없습니다. 읽기 전용 행, 열 또는 셀이 편집 가능한 행, 열 및 셀과 구분되도록 스타일을 수정할 수 있습니다. [Style](#)을 참조하십시오.

**주:**

Excel로 내보내는 경우 읽기 전용 서식이 유지됩니다.

**예**

```
C4=S#Actual.Y#2014,ReadOnly
```

## ReportDescription

이 키워드를 사용하여 양식 설명을 지정합니다.

**구문**

```
ReportDescription=Description
```

*Description*을 양식에 대한 설명으로 바꿉니다. 설명에는 최대 255자를 사용할 수 있습니다.

**예**

```
ReportDescription=Intercompany Detail
```

## ReportLabel

이 키워드를 사용하여 양식 이름을 지정합니다. 이 키워드는 필수입니다.

### 구문

```
ReportLabel=Label
```

*Label*을 양식 이름으로 바꿉니다. 이름에는 최대 40자를 사용할 수 있습니다. 다음과 같은 문자는 이름에 사용할 수 없습니다.

- 별표(\*)
- At 기호(@)
- 백슬래시(\)
- 콜론(:)
- 쉼표(,)
- 중괄호({})
- 슬래시(/)
- 작거나 큼 기호(<)
- 숫자 기호(#)
- 괄호( )
- 마침표(.)
- 파이프(|)
- 더하기 기호(+)
- 물음표(?)
- 따옴표 ""
- 세미콜론(;

### 주:

보고서 레이블에 밑줄(\_)을 사용할 수 있지만 두 문자 사이에 사용해야 합니다. 레이블 이름으로 밑줄만 사용할 수는 없으며 레이블 이름의 끝에 사용할 수도 없습니다.

### 예

```
ReportLabel=ICP Detail
```

## ReportSecurityClass

이 키워드를 사용하여 양식에 지정할 보안 클래스를 지정합니다. 기본값은 [Default]입니다.

## 구문

```
ReportSecurityClass=SecurityClass
```

`SecurityClass`를 적합한 보안 클래스의 이름으로 바꿉니다.

## 예

```
ReportSecurityClass=Corporate
```

## ReportType

이 키워드를 사용하여 보고서 유형을 지정합니다. 이 키워드는 필수이며, 파일을 데이터 양식 스크립트로 로드하려면 값을 `WebForm`으로 설정해야 합니다.

## 구문

```
ReportType=WebForm
```

## RowHeaderPct

이 옵션을 사용하면 양식의 총 너비를 참조하여 행 머리글 너비의 크기를 조정할 수 있습니다. 이 옵션은 행 머리글의 실제 너비가 지정된 백분율을 초과하는 경우에만 적용할 수 있습니다.

예를 들어 실제 행 머리글 너비가 전체 양식 너비의 25%이고 지정한 값이 40%이면 콘텐츠를 25%로 표시할 수 있도록 행 머리글이 40%까지 증가하지 않습니다. 하지만 지정한 값이 10%이면 스크롤 막대를 사용하여 지정한 10% 너비에 들어가도록 행 머리글 너비가 실제 너비인 25%에서 감소합니다. 행 머리글에 여러 열이 있는 경우 시스템에서 모든 열의 너비를 더하여 전체 양식 너비와 비교합니다.

## 구문

```
RowHeaderPct:n
```

`n`을 1에서 100 사이의 값으로 바꿉니다.

## 예

```
RowHeaderPct=30
```

## Scalc

`Scalc` 함수를 사용하여 열과 행에 표준 수치 연산자를 사용하는 사용자정의 공식을 생성할 수 있습니다. 이 값은 행 또는 열 정의나 셀 재정의 정의 내에서 사용됩니다. 지정한 계산은 애플리케이션 서버에서 수행됩니다. 다음 예에서는 열 1에서 열 2를 뺍니다.

```
Scalc(col(1)-col(2))
```

이 값을 사용하여 양식 내에서 텍스트를 생성할 수도 있습니다.

## 구문

`S Calc (<expression> [<operator> <expression>])`



주:

S Calc 계산에 [<operator> <expression>] 쌍을 여러 개 포함할 수 있습니다.

매개변수	설명
operator	계산의 수치 연산자입니다. 지원되는 연산자는 다음과 같습니다. + - * /
expression	계산의 값입니다. 숫자 값 외에도 셀 참조, 행 참조, 열 참조 및 기타 여러 유형의 항목을 포함할 수 있습니다.

S Calc 계산에서는 다음 유형의 항목을 사용할 수 있습니다.

- 차원 멤버에 대한 참조. 다음 예에서는 계정 차원 멤버 Purchases 및 OtherCosts를 참조합니다.  $R6 = S Calc ((A \# Purchases) - (A \# OtherCosts)) * 100$
- Cell(*rowIndex*, *columnIndex*) 구문을 사용하는 셀 참조. 다음 예에서는 양식에서 두 번째 열의 네 번째 행에 있는 셀을 참조합니다.  $R1 = S Calc (Cell (4, 2))$
- Row(*rowIndex*) 구문을 사용하는 행 참조. 다음 예에서는 행 4를 행 2로 나눕니다.  $R3 = S Calc (Row (4) / Row (2))$



주:

멤버 목록이 포함된 행이나 열의 경우 목록의 멤버 합계에 대해 계산이 수행됩니다.

- Col(*columnIndex*) 구문을 사용하는 열 참조. 다음 예에서는 열 1과 열 3을 추가합니다.  $C4 = S Calc (Col (1) + Col (3))$
- 괄호를 사용하여 중첩하는 중첩된 공식
- S Calc 계산이 포함된 셀에 대한 전달 참조



주:

S Calc 계산은 데이터가 저장된 후에만 수행되며 양식을 새로 고쳐야 계산된 결과가 양식에 표시됩니다.

S Calc 행이나 열은 계산 시 다른 S Calc 행이나 열을 참조할 수 있습니다. 하지만 S Calc 행이나 열의 참조를 다른 S Calc 행이나 열에 대해 전달할 수는 없습니다. 예를 들어 다음과 같은 S Calc 전달 참조를 사용할 수 있습니다.

```
C1=A#Sales
C2=A#COGS
C3=S Calc (Col (1) -Col (2))
C4=S Calc (Col (3) /Col (1) *100)
```

## Scale

이 옵션을 사용하여 계산되지 않은 열, 행 또는 셀의 스케일을 지정합니다.

### 구문

```
Scale:n
```

*n*을 -12에서 12 사이의 값으로 바꿉니다.

### 예

```
C4=A#Inventory,Scale:2
```

## SelectablePOVList

이 키워드를 사용하여 사용자가 선택할 수 있는 차원의 멤버를 지정합니다.

선택할 수 있는 차원의 초기 값은 차원에 대해 지정된 경우 백그라운드 POV에서 제공됩니다. 그렇지 않으면 사용자 POV에서 초기 값이 제공됩니다. 초기 값이 목록에 부적합한 경우 기본적으로 목록의 첫 번째 멤버가 지정됩니다.

사용자가 새 멤버를 선택하면 선택한 멤버는 사용자 POV의 일부가 됩니다.

### 구문

```
SelectablePOVList=Dimension{MemberList}
```

```
SelectablePOVList=Dimension{[SystemList]}
```

```
SelectablePOVList=Dimension{Parent.[SystemList]}
```



표 7-5 SelectablePOVList 키워드 구문

매개변수	설명
Dimension	<p>선택할 수 있는 차원을 나타내는 다음 문자 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• S(시나리오)</li> <li>• W(부)</li> <li>• E(엔티티)</li> <li>• V(값)</li> <li>• P(기간)</li> <li>• A(계정)</li> <li>• I(ICP)</li> <li>• C(사용자정의)</li> </ul> <p>차원의 적합한 상위입니다.</p>
Parent	적합한 멤버 목록의 이름입니다.
MemberList	적합한 시스템 목록의 이름입니다.
SystemList	

## 예

```
SelectablePOVList=S{ActualBudget}.Y{Years}.P{Months}.E{[Hierarchy]}.
C1{ProductsAllocate}.C2{Customers}.C3{AllChannels}
```

## ShowDescriptions

이 키워드를 사용하여 양식의 차원 요소에 대해 설명을 표시할지 여부를 지정합니다. 기본값은 `False`입니다.



## 주:

설명이 없으면 레이블이 표시됩니다. 엔티티에 대한 설명이 없으면 엔티티 레이블만 표시되고 상위 레이블은 포함되지 않습니다.

## 구문

```
ShowDescriptions=Boolean
```

*Boolean*을 True 또는 False로 바꿉니다.

예

```
ShowDescriptions=True
```

## ShowLabels

이 키워드를 사용하여 양식의 차원 요소에 대해 레이블을 표시할지 여부를 지정합니다. 기본값은 True입니다.

**ShowLabels**와 **ShowDescriptions**를 [True]로 설정하면 레이블과 설명이 하이픈으로 구분됩니다. 둘 다 False로 설정하면 레이블이 표시되지 않습니다.

구문

```
ShowLabels=Boolean
```

*Boolean*을 True 또는 False로 바꿉니다.

예

```
ShowLabels=False
```

## String

이 옵션을 사용하여 열, 행 또는 셀에 텍스트 문자열을 지정합니다. 행 또는 열 정의나 셀 재정의 정의에 이 옵션을 사용합니다. **String** 옵션은 서버측 계산으로 간주되므로 SCalc가 사용되는 모든 곳에서 사용할 수 있습니다.

예

```
C4=String ("Show this read-only string")
```

## Style

이 옵션을 사용하여 행, 열, 셀, 사용자정의 머리글 또는 차원 머리글의 스타일 속성을 지정합니다. 행, 열, 셀 재정의, 사용자정의 머리글 또는 머리글 옵션 정의 내에서 이 옵션을 사용합니다.

### ▲ 주의:

양식에 스타일 속성을 적용할 때는 주의하십시오. **Style** 옵션은 강력한 기능이며, 부정확하게 사용할 경우 양식의 모양이 완전히 변경될 수 있습니다.

**참고:** Excel로 익스포트하는 경우 스타일 서식이 유지됩니다.

## 구문

각 스타일 등록정보는 등록정보 이름, 콜론 및 값으로 구성됩니다. 값은 공백으로 구분된 하나 이상의 단어일 수 있습니다. 예:

```
Style:color:red;text-align:right
```

Style 옵션은 Oracle Hyperion Financial Management가 아니라 브라우저에서 해석되므로 브라우저가 지원하는 기능으로만 제한됩니다. Style 옵션은 W3C(World Wide Web Consortium)에서 지원되는 표준을 사용합니다.

Style 옵션 키워드는 한 라인에 한 번 사용할 수 있으며 등록정보:값 쌍을 여러 개 추가할 수 있습니다. 예:

```
C3=S#Actual.Y#@CUR(-1),Style:font-color:blue;background-color:red;font-weight:bold
```

Style 예 표에서는 사용할 수 있는 일부 등록정보와 잠재적 값을 보여 줍니다. 전체 목록은 W3C의 등록정보 인덱스를 참조하십시오.

**표 7-6 Style 예**

등록정보	값
Color(전경) Background-color	색상 이름이나 표준 16진수 RGB 표기법입니다. 예: Style: color: red Style: color: rgb(255,0,0) Style: background-color: #ff0000 Style: background-color: yellow
Font-family	글꼴 이름입니다. 예: Style: font-family: Arial
Font	글꼴 스타일입니다. 예: Style: font-style: italic Style 값에 최대 6개의 글꼴 등록정보를 조합할 수 있습니다. 예: Style: font-style:italic;font-family:Arial;font-size:12px
Font-weight	글꼴 두께입니다. 값에는 demi-bold, demi-light, extra-bold, light, extra-light, demi-light 등이 포함됩니다. 예: Style: font-weight: extra-bold
Font-size	글꼴 포인트 크기입니다. 예: Style: font-size: 12px
Alignment	텍스트 맞춤입니다. 값에는 left, right, center, justify 등이 포함됩니다. 예: Style: text-align: center

[스타일] 텍스트 상자 또는 [사용자 정의 머리글 스타일] 텍스트 상자에 행에 대한 첫 문자로 세미콜론을 포함하여 행과 열 교차에서 스타일을 조합할 수 있습니다. Style 키워드 뒤에 세미콜론이 없으면 양식에서 충돌할 경우 행 값이 열 값보다 우선하므로 행에 대해 정의된 스타일이 사용됩니다.

그리드의 각 셀에는 오른쪽 및 맨아래 테두리만 표시되므로 한 픽셀이 인접하는 셀을 구분합니다. Style을 사용하여 오른쪽 또는 맨아래 테두리의 색상을 변경하거나 라인을 파선으로 설정할 수 있습니다. 하지만 맨위 또는 왼쪽 테두리를 활성화하면 각각 위쪽 셀의 맨아래 테두리와 왼쪽 셀의 오른쪽 테두리에 추가되므로 두 개의 테두리가 생성됩니다. 인접한 테두리를 해제하면 이중 테두리를 방지할 수 있습니다. 인접한 테두리를 해제하면 이중 테두리를 방지할 수 있습니다.

또한 머리글이 그리드와 정렬되도록 축의 CustomHeaderStyle을 설정해야 합니다.

## 예

다음 예에서는 행 정의의 Style 키워드 뒤에 세미콜론을 추가하여 행 1(분홍색 배경)과 열 1(분홍색 배경 및 굵은 녹색 텍스트)의 스타일을 혼합합니다. SuppressColHeaderRepeats 예에서는 이 구문이 사용된 경우의 샘플 데이터 입력 양식을 보여줍니다.

```
R1=Blank, CustomheaderStyle: font-weight: bold, CustomHeader: Balance sheet accounts, Style:; Background-color: pink
```

```
C1=S#Actual.Y#@CUR, Style: Background-color: pink; color: green; font-weight: bold
```

	<i>Actual</i>	<i>Budget</i>
<b>Balance sheet accounts</b>		
Cash		1,746,137.00
Short Term Receivables		9,951.32
Inventories		-10,746.57
Short Term Investments		2,831.47
Total Short Term Assets		1,748,173.21
Computers	<b>33,333.00</b>	13,382.68
Buildings		4,351.48
Office Equipment	<b>50,000.00</b>	-948.88
Transportation	<b>80,000.00</b>	4,719.12
Fixtures	<b>800,000.00</b>	931.97
Tangible Assets	<b>963,333.00</b>	22,436.37
Accumulated Depreciation	<b>22,222.00</b>	-87,002.34

## SuppressColHeaderRepeats

이 키워드를 사용하여 반복된 열 머리글을 제외할지 여부를 지정합니다. 기본값은 True입니다.

다음 예에서는 반복된 열 머리글이 제외되는 데이터 양식을 보여 줍니다. 제외를 해제하면 열 머리글 Actual이 August 위에도 표시됩니다.

	Actual	
	July	August
Sales	500,000.00	600,000.00
Purchases	250,000.00	300,000.00

### 구문

```
SuppressColHeaderRepeats=Boolean
```

*Boolean*을 True 또는 False로 바꿉니다.

### 예

```
SuppressColHeaderRepeats=False
```

## SuppressInvalidCols

이 키워드를 사용하여 부적합한 셀이 포함된 열을 양식에서 제외할지 여부를 지정합니다. 기본값은 False입니다.



### 주:

이 설정에 관계없이 관리자가 아닌 사용자에게 대해서는 부적합한 열이 제외됩니다.

### 구문

```
SuppressInvalidCols=Boolean
```

부적합한 열을 제외하려면 *Boolean*을 True로 바꾸고, 부적합한 열을 표시하려면 False로 바꿉니다.

### 예

```
SuppressInvalidCols=False
```

## SuppressInvalidRows

이 키워드를 사용하여 부적합한 셀이 포함된 행을 양식에서 제외할지 여부를 지정합니다. 기본값은 False입니다.



### 주:

이 설정에 관계없이 관리자가 아닌 사용자에게 대해서는 부적합한 행이 제외됩니다.

## 구문

```
SuppressInvalidRows=Boolean
```

부적합한 행을 제외하려면 *Boolean*을 True로 바꾸고, 부적합한 행을 표시하려면 False로 바꿉니다.

## 예

```
SuppressInvalidRows=False
```

## SuppressNoDataCols

이 키워드를 사용하여 데이터 없음 셀이 포함된 열을 양식에서 제외할지 여부를 지정합니다. 기본값은 False입니다.

## 구문

```
SuppressNoDataCols=Boolean
```

데이터가 없는 열을 제외하려면 *Boolean*을 True로 바꾸고, 데이터가 없는 열을 표시하려면 False로 바꿉니다.

## 예

```
SuppressNoDataCols=False
```

## SuppressNoDataRows

이 키워드를 사용하여 데이터 없음 셀이 포함된 행을 양식에서 제외할지 여부를 지정합니다. 기본값은 False입니다.

## 구문

```
SuppressNoDataRows=Boolean
```

데이터가 없는 행을 제외하려면 *Boolean*을 True로 바꾸고, 데이터가 없는 행을 표시하려면 False로 바꿉니다.

## 예

```
SuppressNoDataRows=False
```

## SuppressRowHeaderRepeats

이 키워드를 사용하여 반복된 행 머리글을 제외할지 여부를 지정합니다. 기본값은 True입니다.

### 구문

```
SuppressRowHeaderRepeats=Boolean
```

*Boolean*을 True 또는 False로 바꿉니다.

### 예

```
SuppressRowHeaderRepeats=False
```

## SuppressZeroCols

이 키워드를 사용하여 0이 포함된 열을 양식에서 제외할지 여부를 지정합니다. 기본값은 False입니다.

### 구문

```
SuppressZeroCols=Boolean
```

0이 포함된 열을 제외하려면 *Boolean*을 True로 바꾸고, 0이 포함된 열을 표시하려면 False로 바꿉니다.

### 예

```
SuppressZeroCols=False
```

## SuppressZeroRows

이 키워드를 사용하여 0이 포함된 행을 양식에서 제외할지 여부를 지정합니다. 기본값은 False입니다.

### 구문

```
SuppressZeroRows=Boolean
```

0이 포함된 행을 제외하려면 *Boolean*을 True로 바꾸고, 0이 포함된 행을 표시하려면 False로 바꿉니다.

### 예

```
SuppressZeroRows=False
```

## 상대 기간 사용

행과 열에 대해 다음 키워드를 사용하여 현재 POV에 상대적인 기간을 선택할 수 있습니다.

- @Cur
- @First

- @Last
- @Next
- @Prev

 주:

연도 및 기간 값은 대/소문자를 구분하지 않습니다.

예를 들어 POV에서 선택된 기간이 1월인 경우 다음 공식을 사용하여 생성된 열 또는 행에 2월 데이터가 표시됩니다.

P#CUR+1

기간이 현재 연도를 벗어날 때 시스템에서 상대 연도를 적용하려는 경우 또는 이전 연도를 참조해야 하는 경우 행 또는 열 정의의 일부로 Y#@Cur을 지정해야 합니다.

예를 들어 현재 기간이 마지막 기간인 경우 정확한 연도를 반환하려면 C1=Y#@Cur.P#@Cur(+1)을 지정해야 합니다. 현재 POV가 2013년 12월인 경우 C1=Y#@Cur.P#@Cur(+1)은 2014년 1월을 반환합니다.

현재 POV가 2013년 12월인 경우 C2=P#@Cur(+1)은 2014년 1월을 반환합니다(1월이 연도의 첫 번째 기간이고 12월이 마지막 기간인 경우). 따라서 시스템에서 연도 정보를 정확하게 반영하도록 하려면 열 정의에 연도를 지정합니다 (예: C2=Y#@Cur.P#@Cur(+1)).

다른 차원에 @CUR 함수를 사용하여 현재 POV를 검색할 수 있습니다. 하지만 다른 차원에 @CUR을 사용하는 경우 다른 차원은 시간순이 아니기 때문에 +2와 같은 오프셋 사용이 지원되지 않습니다. 예를 들어 현재 POV에 실제 시나리오가 있는 경우 열에 적합한 예는 C1=S#@CUR을 사용하여 "Actual"을 반환하는 것입니다. POV 시나리오를 예산으로 변경한 경우 동일한 열 정의에서 "Budget"이 반환됩니다.

## 충돌하는 속성의 우선 순위

여러 가지 방법으로 데이터 양식의 소수 자릿수, 스케일 및 POV를 설정할 수 있습니다. 예를 들어 다음과 같은 방법으로 소수 자릿수를 설정할 수 있습니다.

- 양식 - FormNumDecimals=3
- 열 - C4=A#Inventory,NumDecimals:1
- 행 - R4=A#Sales,NumDecimals:2
- 셀 대체 - R2=S#Actual.P#August, Override(2,7,P#July, NumDecimals:4)

경우에 따라 이러한 속성 중 하나의 설정이 동일한 속성의 충돌하는 설정과 교차할 수 있습니다. 예를 들어 열의 스케일 설정이 1인데 교차하는 행의 스케일 설정이 2일 수 있습니다. 소수 자릿수 및 스케일 속성에 대해 적용되는 값은 다음 우선 순위를 기반으로 합니다.

- 셀 대체
- 행
- 열



- 양식
- 기본값 - 소수 자릿수에 대한 기본값은 계정에서 제공됩니다. 스케일에 대한 기본값은 엔티티 통화에서 제공됩니다.

 주:

POV에서 각 차원의 값은 독립적으로 해결됩니다. 예를 들어 계정 차원은 양식 레벨에서 설정되고 시나리오 차원은 행 레벨에서 설정될 수 있습니다.

충돌하는 속성은 우선 순위에 따라 해결됩니다. 예를 들어 소수 자릿수 속성이 셀 재정의에 정의된 경우 양식, 행, 열 또는 기본값 대신 이 값이 사용됩니다.

마찬가지로, 셀에 충돌하는 데이터와 계산이 지정되어 있는 경우 다음 우선 순위가 적용됩니다.

- SCalc
- 데이터

 주:

양식의 SCalc 공식은 왼쪽에서 오른쪽, 맨위에서 맨아래 순으로 계산됩니다. 예를 들어 행 1의 모든 열이 순서대로 계산된 다음 행 2의 모든 열이 계산됩니다. SCalc 공식은 계산 순서에서 앞에 있는 다른 SCalc 셀을 참조할 수 있습니다.

## 데이터 양식 편집

양식 디자이너를 사용하여 데이터 양식을 편집할 수 있습니다. 한 번에 하나의 데이터 양식만 편집할 수 있습니다.

 주:

데이터 양식을 편집한 후 변경사항을 확인하려면 업데이트된 스크립트를 애플리케이션에 재로드해야 합니다.

데이터 양식을 편집하려면 다음을 수행합니다.

1. 통합, 문서 순으로 선택합니다.
2. 데이터 양식을 엽니다.
3. 편집을 누르거나 작업, 편집 순으로 선택합니다.
4. 필요에 따라 양식을 편집합니다.
5. 저장을 누릅니다.

## 데이터 양식 로드

데이터 양식 스크립트를 생성한 후 애플리케이션에 로드합니다.

 주:

웹 팝업 차단기의 예외에 Oracle Hyperion Financial Management를 추가하는 것이 좋습니다. 데이터 로드와 같은 일부 태스크를 수행하면 태스크 상태를 표시하는 상태 창이 팝업됩니다. 컴퓨터에 팝업 차단이 설정되어 있으면 상태 창이 표시되지 않습니다.

데이터 양식을 로드하려면 다음을 수행합니다.

1. **통합, 로드, 문서** 순으로 선택합니다.
2. **문서 유형**에서 **데이터 양식**을 선택합니다.
3. 파일 이름을 입력하거나 **찾아보기**를 눌러 파일을 찾습니다.

 주:

기본적으로 데이터 양식 스크립트는 WDF 파일 확장자를 사용합니다. 로드 프로세스에서는 TXT 또는 CSV와 같은 기타 파일 확장자가 허용되지만 WDF 파일 확장자를 사용하는 것이 좋습니다.

4. **선택 사항:** 다음 단계 중 하나를 수행합니다.
  - 로드되는 파일에 지정된 보안 클래스를 대체하려면 **보안 클래스 대체**를 선택한 다음 **보안 클래스**에서 보안 클래스를 선택합니다.
  - 이 정보를 자신만 사용할 수 있도록 하려면 **개인**을 선택합니다. 이 옵션을 선택하면 **대체 옵션**과 **보안 클래스** 목록을 사용할 수 없습니다.
5. 로드할 양식이 모두 추가될 때까지 이 단계를 반복합니다.
6. **선택 사항:** 데이터 양식을 업데이트하는 경우 **기존 문서를 덮어씁니다**를 선택합니다.
7. **로드**를 누릅니다.

 주:

로드하는 모든 데이터 양식에 적합한 ReportLabel이 있어야 합니다. 여러 양식을 로드하도록 선택했는데 양식 중 하나에 부적합한 ReportLabel이 있는 경우 선택한 양식이 하나도 로드되지 않습니다.

## 데이터 양식 추출

애플리케이션에서 데이터 양식 스크립트를 추출할 수 있습니다. 스크립트를 추출해도 폴더나 애플리케이션에서 스크립트가 삭제되지는 않습니다. 스크립트의 내용만 선택한 위치로 추출됩니다.

데이터 양식 스크립트를 추출한 후 수정하여 애플리케이션에 재로드할 수 있습니다. 추출한 스크립트를 새 스크립트의 모델로 사용할 수도 있습니다.

 주:

애플리케이션에 스크립트를 재로드하는 경우 [기존 문서를 덮어씁니다.] 옵션을 선택하여 애플리케이션의 이전 파일을 바꾸어야 합니다.

데이터 양식을 추출하려면 다음을 수행합니다.

1. 애플리케이션을 엽니다.
2. **통합, 문서** 순으로 선택합니다.
3. **문서 추출**을 누르거나 **작업, 문서 추출** 순으로 선택합니다.
4. 파일 이름을 입력하거나 **찾아보기**를 눌러 파일을 찾습니다.
5. **추출**을 누릅니다.
6. **저장**을 누르고 파일을 저장할 위치를 지정합니다.

## 데이터 양식 삭제

데이터 양식을 삭제하려면 [데이터 입력 양식 관리] 보안 역할을 가진 관리자여야 합니다. 폴더를 삭제하려는 경우 먼저 폴더에 포함된 데이터 양식을 삭제해야 합니다.

데이터 양식을 삭제하려면 다음을 수행합니다.

1. 애플리케이션을 엽니다.
2. **통합, 문서, 데이터 양식** 순으로 선택합니다.
3. 데이터 양식을 선택하고 **삭제**를 누르거나 **작업, 삭제** 순으로 선택합니다.

# 8

## 데이터를 데이터베이스로 추출

이 장의 절차를 사용하여 데이터를 데이터베이스로 추출합니다.



주:

데이터를 플랫폼 파일로 추출하려면 *Oracle Hyperion Financial Management 사용자 가이드*에서 데이터 추출 섹션을 참조하십시오.

데이터를 추출하고 Oracle Essbase 데이터베이스를 사용하여 데이터를 분석하고 보고서를 작성할 수 있습니다. 스타 스키마를 사용하여 데이터를 Essbase 데이터베이스로 보냅니다. 타사 제품에 스타 스키마를 사용할 수도 있습니다.



주:

데이터를 스타 스키마로 내보낸 후 데이터베이스를 생성합니다. 데이터베이스를 생성하려면 해당 릴리스의 데이터베이스 설명서를 참조하십시오.

데이터를 데이터베이스로 추출하기 전에 스타 스키마를 저장할 데이터베이스에 대한 DSN(데이터 소스 이름)을 설정해야 합니다. 스타 스키마를 여러 데이터베이스에 저장하려면 각 데이터베이스에 대해 DSN을 생성할 수 있습니다. 지침은 *Oracle Enterprise Performance Management System 설치 및 구성 가이드*를 참조하십시오.

데이터를 데이터베이스로 추출하려면 관리자 또는 확장 분석 보안 역할이 지정되어야 합니다.

확장 분석 보안 역할이 지정되지 않은 경우 데이터 추출 시 다음과 같은 제한이 적용됩니다.

- ICP 및 모든 사용자정의 차원의 기본 레벨 데이터만 추출할 수 있습니다.
- 시나리오 차원 하나와 연도 차원 하나만 선택할 수 있습니다.
- 시나리오의 기본 빈도에 있는 기간만 선택할 수 있습니다.

## DSN(데이터 소스 이름) 구성

데이터를 데이터베이스로 추출 기능을 사용하려면 DSN(데이터 소스 이름)을 구성하여 데이터베이스 서버와 애플리케이션 서버 간의 연결을 제공해야 합니다. DSN은 데이터베이스 서버 이름과 데이터베이스 서버에 대한 전체 액세스 권한이 있는 사용자의 사용자 이름 및 비밀번호와 같은 기타 데이터베이스 관련 정보를 지정합니다.

DSN을 생성하려면 다음을 수행합니다.

1. 탐색, 관리, 통합 관리 순으로 선택합니다.
2. 관리 태스크에서 **DSN 구성**을 두 번 누릅니다.
3. 데이터 소스 생성을 누릅니다.

4. 대상 데이터베이스의 데이터 소스 이름을 입력합니다.
5. **데이터베이스 유형** 드롭다운 목록에서 데이터베이스 유형(**Oracle, SQL Server** 또는 **DB2**)을 선택합니다.
6. 사용자 이름 및 비밀번호를 입력합니다.

 **주:**

사용자에게 테이블을 생성, 업데이트 및 삭제할 권한이 부여되었는지 확인합니다.

7. 서버 **호스트** 이름을 입력합니다.
8. **포트** 번호를 입력합니다.
9. **선택 사항:** 기본 테이블 영역 또는 파일 그룹 이름을 입력합니다.
10. **선택 사항:** 인덱스 테이블 영역 또는 파일 그룹 이름을 입력합니다.
11. **연결 테스트**를 누릅니다.
12. "성공" 확인이 표시되면 **저장**을 눌러 DSN을 저장합니다.
13. **새로고침**을 눌러 업데이트된 DSN 정보를 표시합니다.

연결을 생성 및 저장하면 [DSN 구성] 페이지에 표시됩니다. 나중에 편집해야 하는 경우 [DSN 편집]을 눌러 편집하거나 더 이상 필요하지 않으면 삭제할 수 있습니다.

## 스타 스키마

애플리케이션당 여러 개의 스타 스키마를 생성할 수 있습니다. 각 스타 스키마의 테이블 이름은 지정된 접두어로 시작합니다. 모든 차원에서 여러 멤버를 선택하여 내보낼 정보를 가장 잘 반영하는 스타 스키마를 생성할 수 있습니다.

 **주:**

셀 텍스트와 라인 항목 세부정보는 스타 스키마로 내보내지지 않습니다.

스타 스키마의 데이터 조합은 내보내도록 선택한 차원 멤버를 기준으로 생성됩니다. 차원 멤버를 더 많이 선택할수록 스타 스키마에 생성해야 하는 데이터 조합 수가 증가하며 익스포트 프로세스를 완료하는 데 더 오랜 시간이 걸립니다. 각 차원에 대해 선택된 멤버 수를 곱하여 데이터 조합 수를 계산할 수 있습니다.

 **주의:**

모든 차원의 멤버를 모두 내보내도록 선택하지 말고 내보낼 데이터 세그먼트를 선택하십시오. 애플리케이션 크기, 데이터 조합 수 및 익스포트를 완료하는 데 걸리는 시간에 따라 시간이 지나치게 길어질 수 있습니다.

예를 들어 다음과 같은 데이터를 내보낼 수 있습니다.

- 시나리오 - 실제
- 연도 - 2014
- 기간 - 1월
- 뷰 - 주기적
- 엔티티 - Regional, United States, Florida, Connecticut
- 값 - USD
- 계정 - Gross Margin, Sales, Total Costs
- ICP - [None]
- C1 - Balls, Tennis Balls, Golf Balls
- C2 - All Customers, Customer2, Customer3, Customer4, Customer5
- C3 - [None]
- C4 - [None]

그러면 Oracle Essbase Integration Services에서 생성된 스타 스키마를 사용하여 정보를 보고 사용해야 하는 대상을 반영하는 데이터 큐브를 하나 이상 생성할 수 있습니다. 스타 스키마에는 이러한 멤버에 대한 180개 데이터 조합이 포함됩니다( $1 * 1 * 1 * 1 * 4 * 1 * 3 * 1 * 3 * 5 * 1 * 1 = 180$ ).

#### 주:

Integration Services를 통해 메타데이터를 Oracle Essbase로 익스포트하는 경우 [데이터를 데이터베이스로 추출] 기능은 Oracle Hyperion Financial Management 애플리케이션에 표시되는 것과 다른 순서로 메타데이터를 사용합니다.

원하는 경우 로컬 통화 데이터만 추출할 수 있습니다. 로컬 통화 데이터만 추출하려면 POV에서 [값] 차원의 [엔티티 통화] 멤버를 사용합니다. 엔티티 통화는 통화의 자리 표시자 역할을 하며 선택된 각 엔티티의 기본 통화를 추출합니다.

선택한 시나리오가 YTD이면 데이터 추출은 기간 파생 데이터를 추출합니다. 예를 들어 시나리오가 ZeroView=YTD라고 가정합니다. 첫 번째 기간의 비용 계정에는 값 100이 있습니다. 두 번째 기간에는 해당 계정에 대한 데이터가 보고되지 않았습니다. 두 번째 기간에 대해 파생되는 기간 값은 -100으로, YTD 금액을 강제로 0으로 만듭니다. 기간을 기준으로 이 POV에 대해 데이터를 데이터베이스로 추출을 실행할 경우 처음 세 기간의 값은 각각 100, -100, 0입니다.

## 스타 스키마 형식

스타 스키마를 생성하는 경우 다음과 같은 추출 형식 옵션 중 하나를 선택합니다.

- 표준
- 메타데이터만
- 선택한 메타데이터만
- Essbase
- 데이터 웨어하우스

선택한 추출 형식 옵션에 따라 사용되는 스키마 형식이 결정됩니다. 각 스키마 형식은 다른 테이블 세트를 생성합니다. 사용할 수 있는 스키마 형식은 다음과 같습니다.

- 표준 Essbase 스키마 - 이 스키마는 표준, 메타데이터만 및 선택한 메타데이터만 추출 형식 옵션에 사용됩니다.
- SQL 및 Essbase 스키마 - 이 스키마는 Essbase 추출 형식 유형에 사용됩니다.
- 웨어하우스 정규화된 계층 스키마 - 이 스키마는 데이터 웨어하우스 추출 형식 유형에 사용됩니다.

## 접두어 테이블

각 스키마 형식에 대해 차원 테이블과 하나의 데이터 필드에 대한 키가 포함된 *PREFIX\_FACT* 테이블이 생성됩니다. 시스템에서는 메타데이터의 추출 타임스탬프를 추적하기 위해 *PREFIX\_DIMENSION* 테이블 및 *HFM\_EA\_EXTRACT* 테이블도 생성합니다. 엔티티 차원의 경우 *PREFIX\_ENTITY* 및 *PREFIX\_PARENT* 두 개의 테이블이 생성됩니다.

SQL 및 Essbase 합산 스키마 형식의 경우 *PREFIX\_DIMENSION\_BASE* 테이블도 생성됩니다.

웨어하우스 정규화된 계층 스키마 형식의 경우 *PREFIX\_DIMENSION\_PARENT* 테이블이 생성됩니다.

### 주:

뷰, 연도 및 상위 차원에 대해서는 Base 및 Parent 테이블이 생성되지 않습니다.

예를 들어 관계형 테이블 접두어가 DEMO인 경우 Essbase 형식에 대해 다음과 같은 테이블이 생성됩니다.

- HFM\_EA\_EXTRACT1
- DEMO\_FACT
- DEMO\_YEAR
- DEMO\_VIEW
- DEMO\_PARENT
- DEMO\_SCENARIO 및 DEMO\_SCENARIO\_BASE
- DEMO\_PERIOD 및 DEMO\_PERIOD\_BASE
- DEMO\_VALUE 및 DEMO\_VALUE\_BASE
- DEMO\_ENTITY 및 DEMO\_ENTITY\_BASE
- DEMO\_ICP 및 DEMO\_ICP\_BASE
- DEMO\_ACCOUNT 및 DEMO\_ACCOUNT\_BASE
- DEMO\_CUSTOM 및 DEMO\_CUSTOM\_BASE

## 데이터 생성 및 스타 스키마로 익스포트

스키마 및 내보낼 셀의 차원 멤버에 대해 테이블을 식별하는 접두어를 지정하여 스타 스키마를 생성할 수 있습니다. 데이터를 익스포트 전에 애플리케이션 데이터가 통합되었는지 확인합니다.

**▲ 주의:**

Oracle Hyperion Financial Management 데이터베이스를 데이터 익스포트의 대상 데이터베이스로 사용하지 마십시오.

**참고:** Oracle 데이터베이스 사용자의 경우: 추출한 텍스트 데이터는 NVARCHAR(유니코드) 형식으로 저장됩니다. SELECT 문에 Oracle "translate" 명령을 사용하여 텍스트를 유니코드에서 ANSI 형식으로 변환합니다. 예를 들어 SELECT translate (LABEL using CHAR\_CS) as LABEL FROM DEMO\_ACCOUNT를 입력합니다.

많은 양의 데이터를 추출하는 경우 추출에 사용할 디스크 공간이 충분한지 확인합니다.

웹 팝업 차단의 예외에 Financial Management를 추가하는 것이 좋습니다. 데이터 로드와 같은 일부 태스크를 수행하면 태스크 상태를 표시하는 상태 창이 팝업됩니다. 컴퓨터에 팝업 차단이 설정되어 있으면 상태 창이 표시되지 않습니다.

스타 스키마를 생성하고 Financial Management 데이터를 추출하려면 다음을 수행합니다.

1. **통합, 추출, 데이터** 순으로 선택합니다.
2. 다음 태스크 중 하나를 수행합니다.
  - POV 차원을 템플릿에 저장한 경우 **템플릿**에서 템플릿을 선택합니다.
  - 템플릿을 사용하지 않는 경우 내보낼 POV 차원 멤버를 선택합니다.
3. **추출 대상** 섹션에서 **유형** 옵션에 대해 **데이터베이스**를 선택합니다.

**✎ 주:**

이 옵션은 [관리자] 또는 [확장 분석] 보안 역할이 지정된 경우에만 사용할 수 있습니다.

4. **추출 형식**에서 다음 옵션 중 하나를 선택합니다.
  - **표준**
  - **메타데이터만** - 메타데이터만 추출합니다.
  - **선택한 메타데이터만** - 선택한 차원 멤버의 메타데이터만 추출합니다.
  - **Essbase**
  - **데이터 웨어하우스**
5. **옵션**에서 다음 옵션을 하나 이상 선택합니다.
  - **동적 계정 추출**
  - **계산된 데이터**
  - **파생된 데이터**
6. **라인 항목 세부정보**에서 다음 옵션 중 하나를 선택합니다.
  - **전체 셀 요약**
  - **없음**
7. **스키마 작업**에서 **스타 스키마 생성**을 선택합니다.



- 대상 데이터베이스(DSN)에서 스타 스키마를 내보낼 대상 데이터베이스를 선택합니다.

 주:

데이터를 내보내는 소스 Financial Management 데이터베이스를 대상 데이터베이스로 사용하지 마십시오.

- 관계형 테이블 접두어에서 스타 스키마의 테이블을 식별하는 접두어를 입력하거나 기본 애플리케이션 이름 접두어를 사용합니다.

 주:

접두어는 최대 10자의 영숫자 문자를 사용할 수 있으며 문자로 시작해야 합니다. 밑줄은 사용할 수 없습니다. 기본 접두어는 애플리케이션 이름입니다.

- 추출을 누릅니다.

## 스타 스키마 업데이트

데이터를 이전에 정의한 스타 스키마로 내보낼 수 있습니다. 스타 스키마를 업데이트할 때 다른 차원 멤버를 지정할 수 있습니다.

 주:

스타 스키마를 업데이트하면 팩트 테이블이 업데이트됩니다.

웹 팝업 차단기의 예외에 Oracle Hyperion Financial Management를 추가하는 것이 좋습니다. 데이터 로드와 같은 일부 태스크를 수행하면 태스크 상태를 표시하는 상태 창이 팝업됩니다. 컴퓨터에 팝업 차단이 설정되어 있으면 상태 창이 표시되지 않습니다.

스타 스키마를 업데이트하려면 다음을 수행합니다.

- 통합, 추출, 데이터 순으로 선택합니다.
- 다음 태스크 중 하나를 수행합니다.
  - POV 차원을 템플릿에 저장한 경우 **템플릿**에서 템플릿을 선택합니다.
  - 템플릿을 사용하지 않는 경우 내보낼 POV 차원 멤버를 선택합니다.
- 추출 대상 유형에서 데이터베이스를 선택합니다.

 주:

이 옵션은 [관리자] 또는 [확장 분석] 보안 역할이 지정된 경우에만 사용할 수 있습니다.

- 추출 형식에서 다음 옵션 중 하나를 선택합니다.

- 표준
  - 메타데이터만 - 메타데이터만 추출합니다.
  - 선택한 메타데이터만 - 선택한 차원 멤버의 메타데이터만 추출합니다.
  - Essbase
  - 데이터 웨어하우스
5. 옵션에서 다음 옵션을 하나 이상 선택합니다.
    - 동적 계정 추출
    - 계산된 데이터
    - 파생된 데이터
  6. 라인 항목 세부정보에서 다음 옵션 중 하나를 선택합니다.
    - 전체 셀 요약
    - 없음
  7. 스키마 작업에서 스타 스키마 업데이트를 선택합니다.
  8. 대상 데이터베이스(DSN)에서 스타 스키마를 내보낼 대상 데이터베이스를 선택합니다.

 주:

데이터를 내보내는 소스 Financial Management 데이터베이스를 대상 데이터베이스로 사용하지 마십시오.

9. 관계형 테이블 접두어에서 스타 스키마의 테이블을 식별하는 접두어를 입력하거나 기본 애플리케이션 이름 접두어를 사용합니다.

 주:

접두어는 최대 10자의 영숫자 문자를 사용할 수 있으며 문자로 시작해야 합니다. 밑줄은 사용할 수 없습니다. 기본 접두어는 애플리케이션 이름입니다.

10. 추출을 누릅니다.

## 스타 스키마 삭제

더 이상 필요하지 않은 스타 스키마는 삭제할 수 있습니다. 스타 스키마를 삭제하면 스타 스키마 데이터, 메타데이터 및 테이블이 모두 삭제됩니다.

스타 스키마를 삭제하려면 다음을 수행합니다.

1. 통합, 추출, 데이터 순으로 선택합니다.
2. 추출 대상 유형에서 데이터베이스를 선택합니다.
3. 대상 데이터베이스(DSN)에서 스타 스키마가 포함된 데이터베이스를 선택합니다.
4. 관계형 테이블 접두어에서 스타 스키마의 테이블을 식별하는 접두어를 입력합니다.

 주:

접두어는 최대 10자의 영숫자 문자를 사용할 수 있으며 문자로 시작해야 합니다. 기본 접두어는 애플리케이션 이름입니다.

5. 스키마 작업에서 스타 스키마 삭제를 선택합니다.

## 스타 스키마 템플리트 생성

다시 사용할 수 있도록 POV에 이름을 지정하고 저장할 수 있는 스타 스키마 템플리트를 생성할 수 있습니다.

스타 스키마 템플리트를 생성하려면 다음을 수행합니다.

1. 통합, 추출, 데이터 순으로 선택합니다.
2. POV 차원 멤버를 선택합니다.
3. 대상 데이터베이스(DSN)에서 스타 스키마를 내보낼 대상 데이터베이스를 선택합니다.
4. 관계형 테이블 접두어에 스타 스키마의 테이블을 식별하는 접두어를 입력합니다.

 주:

접두어는 최대 10자의 영숫자 문자를 사용할 수 있으며 문자로 시작해야 합니다. 기본 접두어는 애플리케이션 이름입니다.

5. 템플리트 저장을 누릅니다.
6. 템플리트 이름을 입력하고 확인을 누릅니다.

## 스타 스키마 템플리트 삭제

스타 스키마 템플리트를 삭제하려면 다음을 수행합니다.

1. 통합, 추출, 데이터 순으로 선택합니다.
2. 템플리트에서 삭제할 템플리트를 선택합니다.
3. 템플리트 삭제를 누릅니다.
4. 시스템 프롬프트에서 확인을 누릅니다.

# 9

## 보고서 정의

Oracle Hyperion Financial Management에서 다음과 같은 보고서 유형을 정의할 수 있습니다.

- 분개 보고서 - 분개 모듈에서 선택한 기준에 따라 특정 분개나 분개 목록에 대한 정보를 표시합니다.
- 내부거래 파트너(ICP) 일치 보고서 - 통합 과정에서 제거되는 내부거래 트랜잭션을 표시합니다.

각 보고서에 대해 새 보고서 정의를 생성하거나, 보고서 정의를 열고 수정할 수 있습니다.

다음 방법 중 하나를 사용하여 보고서를 정의합니다.

- 보고서 페이지 옵션을 사용하여 값 지정. *Oracle Hyperion Financial Management 사용자 가이드*를 참조하십시오.
- 텍스트 편집기를 사용하여 스크립트에서 보고서 옵션 지정

모든 보고서에 날짜, 시간 및 사용자 필드가 머리글 정보로 자동으로 표시됩니다. POV 정의는 각 보고서마다 다릅니다. 보고서에 대한 자세한 예는 [분개 보고서 스크립트 정의](#) 및 [내부거래 매칭 보고서 스크립트 정의](#)를 참조하십시오.

Financial Management용 샘플 애플리케이션을 설치하면 샘플 내부거래 파트너 일치 보고서 스크립트가 포함됩니다. 이러한 파일은 Financial Management를 설치한 디렉토리의 Sample Applications 폴더에 있습니다.

## 분개 보고서 스크립트 정의

분개 보고서를 생성하여 분개 상태를 확인하고 분개 조정을 검토합니다. 분개 보고서를 생성하여 특정 분개나 분개 목록에 대한 정보를 표시할 수 있습니다. 예를 들어 특정 상태, 대차 계정 유형, 엔티티 또는 계정이 포함된 분개만 표시하도록 선택할 수 있습니다. 보고서에 표시되는 열을 선택하고 정렬 순서를 변경할 수도 있습니다.

보고서 정의에서 분개 보고서에 대한 엔티티 및 계정 필터를 설정할 수 있습니다. 엔티티 필터링 구문은 EntityFilter=ParentName.ChildName입니다(예: UnitedStates.Maryland). 계정 필터링 구문은 AccountFilter=MemberName입니다(예: Sales).

다음 예에서는 샘플 분개 보고서 정의를 보여 줍니다.

```
ReportType=Journal

ReportDescription=Tax Journals

POV=S#Actual.Y#2014.P#January.V#<Entity Curr Adjs>

DisplayColumn_0=Account,Ascending,NoRepeat,Yes,Label

DisplayColumn_1=ICP,,NoRepeat,No,Label

DisplayColumn_2=Products,,NoRepeat,No,Label

DisplayColumn_3=Markets,,NoRepeat,No,Label

StatusFilter=Working,Submitted,Rejected,Approved,Posted

TypeFilter=Regular

BalanceTypeFilter=Balanced,Unbalanced

EntityFilter=UnitedStates.Maryland

AccountFilter=Sales
```

## 내부거래 매칭 보고서 스크립트 정의

내부거래 매칭 보고서는 분석 및 감사 용도로 내부거래 트랜잭션을 추적하는 데 유용합니다. 내부거래 매칭 보고서에는 내부거래 파트너(IsICP) 속성이 설정된 내부거래 파트너 차원 및 엔티티의 일치 항목이 표시됩니다.

내부거래 파트너(ICP) 일치 보고서에는 통합하는 동안 제거된 내부거래 트랜잭션이 나열됩니다. 내부거래 트랜잭션은 내부거래 파트너 차원에서 관리됩니다. 내부거래 차원에는 계정에 대해 존재할 수 있는 모든 내부거래 잔액이 포함됩니다. 데이터 그리드, 데이터 로드, 분개 또는 데이터 양식을 통해 내부거래 트랜잭션을 입력할 수 있습니다. Oracle Hyperion Financial Management에서는 계정 및 사용자정의 차원에서 내부거래 트랜잭션 세부정보를 추적 및 제거할 수 있습니다. 내부거래 트랜잭션은 두 내부거래 엔티티의 첫 번째 공통 상위를 기준으로 제거되며, 값 차원의 [Elimination] 멤버를 통해 제거됩니다.

사용자 인터페이스를 사용하여 내부거래 매칭 보고서를 생성할 수 있습니다.

보고서의 계정을 선택하거나, 시스템이 플러그 계정을 기반으로 계정과 일치 계정을 생성하는 플러그 계정 옵션을 사용할 수 있습니다. 반대 트랜잭션, 사용자정의 차원 또는 내부거래 세부정보를 제외하도록 선택할 수 있습니다. 보고서를 인쇄할 때 보고서 설정을 재정의하여 요구에 맞게 보고서를 사용자정의할 수 있습니다.

## 내부거래 매칭 보고서의 멤버 목록 선택

생성된 보고서의 POV를 구성하는 차원 멤버를 지정합니다. 보고서 엔티티 및 파트너의 멤버 목록을 선택할 수 있습니다. 예를 들어 엔티티 필드에 대해 지역 멤버 목록을 선택할 수 있습니다. 엔티티 및 파트너의 멤버 목록을 선택하면 파트너 목록의 모든 엔티티에 대해 엔티티 목록에 있는 모든 엔티티의 ICP 트랜잭션이 처리됩니다. ICP 속성을 사용하도록 설정된 엔티티에 대해서만 트랜잭션이 처리됩니다.

## 내부거래 매칭 보고서의 계정 선택

트랜잭션을 일치시키려는 계정과 일치 계정을 정의해야 합니다. 시스템에서 플러그 계정을 기준으로 계정과 일치 계정을 생성하려는 경우 플러그 계정 옵션을 지정할 수 있으며, 계정과 일치 계정이 자동으로 생성됩니다.

## 내부거래 매칭 보고서의 소수 자릿수 지정

내부거래 매칭 보고서 정의에 Decimal 키워드를 추가하여 보고서에 표시할 소수 자릿수를 지정할 수 있습니다. Decimal에 사용할 수 있는 값은 기본값인 0-9입니다. 10진수 값을 지정하지 않으면 계정에 정의된 기본 10진수 설정이 사용됩니다.

## 내부거래 매칭 보고서의 스타일시트 선택

보고서를 생성할 때 드롭다운 목록에서 스타일시트를 선택할 수 있습니다. 보고서 정의에 StyleSheet 키워드를 추가하여 보고서에 사용할 스타일시트를 지정할 수도 있습니다(예: `StyleSheet=HFM_IntercompanyDefault.xsl`). 보고서 정의에 스타일시트를 지정하지 않으면 기본 스타일 시트가 사용됩니다.

## 내부거래 매칭 보고서의 통화 지정

내부거래 매칭 보고서를 지정한 통화로 생성할 수 있습니다. 이렇게 하면 금액을 통합하기 전에 보고서를 실행하고 내부거래 트랜잭션을 공통 통화로 검증할 수 있습니다. 예를 들어 값을 EUR 통화로 확인하려면 POV의 값을 USD에서 EUR로 변경하고 보고서를 생성할 수 있습니다.

환산되지 않은 통화를 사용하여 보고서를 정의하면 규칙 파일의 Sub Translate 섹션에 정의된 환산 규칙을 사용하여 환산 프로세스가 수행됩니다. 또한 환산된 금액이 해당 통화 값 차원 멤버에 저장됩니다. 그러나 이전에 보고 통화가 환산되었고 엔티티의 환산 상태가 이상 없음인 경우 내부거래 매칭 보고서 처리에 저장된 환산 금액을 다시 환산하여 사용할 필요가 없습니다.

예를 들어 EUR 통화에 대한 내부거래 일치 보고서를 실행하는 경우 먼저 EUR 통화(V#EUR)에 대한 환산된 데이터가 생성되었는지 확인합니다. 또한 환산 상태가 이상 없음인지 확인합니다. 엔티티의 환산 상태가 TR(환산 필요)이면 재환산을 통해 환산된 데이터가 적합하지 확인합니다. 그렇지 않으면 저장된 환산 금액을 사용하여 처리합니다. 그러나 데이터가 EUR로 환산되지 않은 경우에는 Sub Translate에 정의된 대로 환산 프로세스가 수행되고 환산 금액이 EUR 값 멤버에 저장됩니다. EUR 환산 금액은 내부거래 매칭 보고서에도 사용됩니다.

## 내부거래 매칭 보고서의 제외 옵션

내부거래 매칭 보고서를 생성하는 경우 여러 유형의 내부거래 트랜잭션과 세부정보를 보고서에서 제외할 수 있습니다. 보고서를 인쇄할 때 이 제외 옵션을 선택할 수도 있습니다.

### 일치 항목 숨김

내부거래 매칭 보고서를 생성하면 일치하는 엔티티/파트너 트랜잭션을 표시하거나 제외할 수 있습니다. 제외하도록 선택하면 엔티티 및 파트너 계정이 일치 허용 오차 금액이나 백분율 내에 있을 경우 트랜잭션이 제외됩니다. 예를 들어 엔티티 금액이 299이고 파트너 금액이 200이면 차이는 99입니다. 이 예와 같이 일치 허용 오차 금액이 100이고 엔티티와 파트너의 차이가 100 미만이면 일치 허용 오차 내에 있으므로 트랜잭션이 제외됩니다.

엔티티/파트너 트랜잭션을 제외하지 않도록 선택하면 엔티티 및 파트너 금액이 일치 허용 오차 금액이나 백분율 내에 있어도 트랜잭션이 제외되지 않습니다. 앞의 예를 사용하면 차이 금액이 99이고 일치 허용 오차 내에 있어도 트랜잭션이 제외되지 않습니다. 지정된 일치 허용 오차는 1 단위로 표시됩니다.

### 전환된 트랜잭션 숨김

기본적으로 엔티티에 대한 트랜잭션을 표시할 때마다 내부거래 파트너의 전환된 트랜잭션이 표시됩니다. 보고서를 생성할 때 이 반대 트랜잭션을 제외할 수 있습니다.

이 옵션은 보고서에 대해 하나의 일치 계정이 있는 경우에 유용합니다. 예를 들어 Revenue 및 Expense에 대한 내부거래 트랜잭션을 저장하는 데 사용되는 Cash 계정이 있을 수 있습니다. 이 경우 보고서의 일치 계정이 하나뿐이므로 이중 입력을 방지하려면 [전환된 트랜잭션 제외] 옵션을 사용해야 합니다.

### 세부정보 숨김

다른 금액만 보고서에 표시하려는 경우 내부거래 세부정보를 제외할 수 있습니다. 이 옵션을 선택하면 보고서에 내부거래 트랜잭션이 표시되지 않고 각 엔티티/파트너 섹션의 합계 차이만 인쇄됩니다. 불일치가 있고 각 내부거래 트랜잭션을 확인해야 하는 경우 보고서를 다시 생성하고 내부거래 세부정보를 표시할 수 있습니다.

### 사용자정의 차원 숨김

임의 사용자정의 차원의 열을 제외할 수 있습니다.

### 멤버 표시 옵션

보고서에 차원 멤버의 레이블, 설명 또는 둘 다를 표시할 수 있습니다.

### 그룹화 기준 옵션

사용자정의 차원을 기준으로 내부거래 파트너 트랜잭션을 그룹화할 수 있습니다. 이 옵션에 따라 세부정보가 정렬되며 그룹의 소계가 제공됩니다.

## 내부거래 매칭 보고서 스크립트 키워드

이 절의 키워드를 사용하여 내부거래 파트너 일치 보고서 스크립트를 정의합니다. 스크립트를 생성한 후 RPT 파일 이름 확장자로 저장합니다.

**주:**

내부거래 파트너 일치 보고서 스크립트 키워드는 대/소문자를 구분하지 않습니다.

## ReportType

이 키워드는 보고서 유형을 지정하며 스크립트에 필요합니다.

### 구문

```
ReportType=Intercompany
```

## ReportDescription

이 키워드는 보고서에 대한 설명을 지정합니다. 보고서 설명에는 최대 40자를 사용할 수 있습니다. 스크립트에 필요합니다.

### 구문

```
ReportDescription=ReportDescription
```

*ReportDescription*을 보고서에 대한 설명으로 바꿉니다. 예:

```
ReportDescription=Intercompany Elimination Report
```

## StyleSheet

이 키워드는 보고서에 사용할 스타일시트를 지정합니다.

### 구문

```
StyleSheet=StyleSheetFileName
```

*StyleSheetFileName*을 보고서에 대한 스타일시트로 바꿉니다. 예:

```
ReportDescription=HFM_IntercompanyDefault.xml
```

보고서 정의에 스타일시트를 지정하지 않으면 기본 스타일 시트가 사용됩니다.

## POV

이 키워드는 보고서의 POV를 지정하며 스크립트에 필요합니다.

### 구문

```
POV=S#Scenario.Y#Year.P#Period.V#Value.W#View
```



*Scenario*, *Year*, *Period*, *Value* 및 *View*를 적합한 차원 멤버로 바꿉니다. 예:

```
POV=S#Actual.Y#2014.P#January.V#USD.W#YTD
```



주:

상위 통화 조정과 같은 조정 값 멤버는 지원되지 않습니다.

## 엔티티

이 키워드는 보고서에 표시할 엔티티 또는 엔티티 멤버 목록을 지정합니다.

### 구문

```
Entity=E#Parent.Entity
```

```
Entity=E{EntityList}
```

*Entity.Parent*를 엔티티-상위 조합으로 바꿉니다. *EntityList*를 적합한 멤버 목록의 이름으로 바꿉니다. 예:

```
Entity=E#UnitedStates.Connecticut
```

```
Entity=E{Geographical.[Base]}
```

## Partner

이 키워드는 보고서에 표시할 파트너 또는 파트너 멤버 목록을 지정합니다.

### 구문

```
Partner=E#PartnerParent.PartnerEntity
```

```
Partner=E{PartnerList}
```

*PartnerParent.PartnerEntity*를 상위-엔티티 조합으로 바꿉니다. *PartnerList*를 적합한 파트너 멤버 목록의 이름으로 바꿉니다. 예:

```
Partner=E#UnitedStates.Florida
```

```
Partner=E{Geographical.[Base]}
```

## AccountEntity 및 AccountPartner

AccountEntity\_x 및 AccountPartner\_x 키워드는 일치시킬 계정을 지정합니다. 일치시킬 각 계정 쌍에 대해 AccountEntity\_x와 AccountPartner\_x를 0부터 시작하여 지정합니다. 일대다 또는 다대다 일치 보고서를 생성하려면 정확한 키워드를 사용하여 추가 계정을 지정합니다. 한 보고서 내에서 중복 키워드를 사용할 수는 없습니다. 예를 들어 한 보고서 내에서 AccountEntity\_0을 여러 번 사용할 수 없습니다.

### 구문

```
AccountEntity_0=A#Sales.C1#GolfBalls.C2#Customer2
```

```
AccountPartner_0=A#Purchases.C1#GolfBalls.C2#Customer2
```

다음 예에서는 이러한 키워드를 사용하여 하나의 일치 보고서에 있는 모든 계정을 표시합니다.

```
AccountEntity_0=A#1004780
```

```
AccountEntity_1=A#1004790
```

```
AccountEntity_2=A#1005850
```

```
AccountEntity_3=A#1005850
```

```
AccountPartner_0=A#2000100
```

```
AccountPartner_1=A#2000140
```

```
AccountPartner_2=A#2000210
```

```
AccountPartner_3=A#2000250
```

```
AccountPartner_4=A#2000320
```

```
AccountPartner_5=A#2000430
```

```
AccountPartner_6=A#2000560
```

```
AccountPartner_7=A#2000630
```

```
AccountPartner_8=A#2000680
```

다음 예에서는 Entity 및 Partner 키워드를 사용하여 각기 다른 계정 쌍이 포함된 여러 보고서를 생성합니다. 이것은 한 계정이 두 계정과 일치하는 보고서의 예입니다(일대다).

```
Report 1
```

```
AccountEntity_0=A#1004780
```

```
AccountPartner_0=A#2000100
```

```
AccountPartner_1=A#2000140
```

다음 예에서는 두 계정이 한 계정과 일치하는 보고서를 보여 줍니다(다대일).

Report 2

AccountEntity\_0=A#1004790

AccountEntity\_1=A#2000210

AccountPartner\_1=A#2000250

보고서 3은 한 계정이 한 계정과 일치하는 보고서의 예입니다(일대일).

Report 3

AccountEntity\_0=A#1005850

AccountPartner\_0=A#2000320

보고서 4는 두 계정이 네 계정과 일치하는 보고서의 예입니다(다대다).

Report 4

AccountEntity\_0=A#1005850

AccountEntity\_1=A#1005860

AccountPartner\_0=A#2000430

AccountPartner\_1=A#2000560

AccountPartner\_2=A#2000630

AccountPartner\_3=A#2000680

## SuppressIfMatch

이 키워드는 엔티티 및 파트너 금액이 일치 허용 오차 금액 내에 있는 경우 트랜잭션을 제외합니다.

### 구문

SuppressIfMatch=*Boolean*

*Boolean*을 Yes 또는 No로 바꿉니다. 예:

```
SuppressIfMatch=Yes
```

예를 들어 엔티티 금액이 299이고 파트너 금액이 200이면 차이는 99입니다. 이 예와 같이 일치 허용 오차 금액이 100이고 엔티티와 파트너의 차이가 100 미만이면 일치 허용 오차 내에 있으므로 트랜잭션이 제외됩니다.

엔티티/파트너 트랜잭션을 제외하지 않도록 선택하면 엔티티 및 파트너 금액이 일치 허용 오차 금액 내에 있어도 트랜잭션이 제외되지 않습니다. 앞의 예를 사용하면 차이 금액이 99이고 일치 허용 오차 내에 있어도 트랜잭션이 제외되지 않습니다.



**주:**

일치 허용 오차는 1 단위로 지정됩니다.

## SuppressReversedTransactions

이 키워드는 각 해당 엔티티 트랜잭션에 대한 파트너의 전환된 트랜잭션을 제외합니다.

### 구문

```
SuppressReversedTransactions=Boolean
```

*Boolean*을 Yes 또는 No로 바꿉니다. 예:

```
SuppressReversedTransactions=Yes
```

기본적으로 엔티티에 대한 트랜잭션을 표시할 때마다 내부거래 파트너의 전환된 트랜잭션이 표시됩니다. 보고서를 생성할 때 이 전환된 트랜잭션을 제외할 수 있습니다.

이 옵션은 보고서에 대한 일치 계정이 한 개 있는 경우에 유용합니다. 예를 들어 수익 및 비용에 대한 내부거래 트랜잭션을 저장할 "wash" 계정이 한 개 있는 경우입니다. 이 경우 보고서의 일치 계정이 하나뿐이므로 이중 입력을 방지하려면 [전환된 트랜잭션 제외] 옵션을 사용해야 합니다.

## SuppressDetails

이 키워드는 내부거래 세부정보를 제외하고 각 엔티티/파트너 섹션의 합계 차이만 인쇄합니다.

### 구문

```
SuppressDetails=Boolean
```

*Boolean*을 Yes 또는 No로 바꿉니다. 예:

```
SuppressDetails=Yes
```

차액 금액만 표시하는 내부거래 매칭 보고서를 생성하려면 보고서 생성 시 내부거래 세부정보를 제외할 수 있습니다. 이 옵션을 선택하면 보고서에 내부거래 트랜잭션이 표시되지 않고 각 엔티티/파트너 섹션의 합계 차이만 인쇄됩니다. 불일치가 있고 각 내부거래 트랜잭션을 확인해야 하는 경우 보고서를 다시 생성하고 내부거래 세부정보를 표시할 수 있습니다.

## MatchingTolerance

이 키워드를 통해 값을 지정하여 잔액 차액이 특정 금액을 초과하는 트랜잭션만 보거나 기본값 0을 사용할 수 있습니다.



주:

일치 허용 오차는 1 단위로 지정됩니다.

### 구문

```
MatchingTolerance=ToleranceValue
```

*ToleranceValue*를 10억 미만의 숫자로 바꿉니다. 이 키워드는 999999999로 제한됩니다. 예:

```
MatchingTolerance=100
```

예를 들어 엔티티 금액이 299이고 파트너 금액이 200이면 차이는 99입니다. 이 예와 같이 일치 허용 오차 금액이 100이고 엔티티와 파트너의 차이가 100 미만이면 일치 허용 오차 내에 있으므로 SuppressIfMatch 키워드를 사용하여 트랜잭션이 제외되도록 할 수 있습니다.

## SuppressCustoms

이 키워드는 사용자정의 차원을 제외합니다.

### 구문

```
SuppressCustomAlias=Boolean
```

*Boolean*을 Yes 또는 No로 바꿉니다. 기본값은 Yes입니다. 예:

```
SuppressCustomFlows=Yes
```

## ScaleFactor

이 키워드는 소수점 위치를 식별하여 금액이 표시되는 단위를 지정합니다. 예를 들어 스케일 요소로 3을 입력하면 보고서 금액이 천 단위로 표시됩니다. 스케일 요소가 0이면 보고서 금액이 1 단위로 표시됩니다.

### 구문

```
ScaleFactor=Scale
```

*Scale*을 다음 숫자 중 하나로 바꿉니다.

- 0 = 단위
- 1 = 십
- 2 = 백
- 3 = 천
- 4 = 만
- 5 = 십만
- 6 = 백만
- 7 = 천만
- 8 = 억
- 9 = 십억

예:

```
ScaleFactor=3
```

다음 예에서는 숫자 12345.78이 보고서에 12.345678로 표시됩니다.

## 소수 자릿수

이 키워드는 보고서에 표시할 소수 자릿수를 지정하며 계정에 정의된 소수 자릿수를 재정의할 수 있습니다.

### 구문

```
Decimal=NumberDecimalPlaces
```

*NumberDecimalPlaces*를 숫자 0-6으로 바꿉니다. 10진수 값을 지정하지 않으면 계정에 정의된 기본 10진수 설정이 사용됩니다. 예:

```
Decimal=3
```

위의 예에서는 숫자 123.4567이 123.457로 표시됩니다.

## DisplayLabels

이 키워드는 보고서에 멤버 레이블을 표시할지 여부를 지정합니다.

### 구문

```
DisplayLabels=Boolean
```

*Boolean*을 Yes 또는 No로 바꿉니다. 기본값은 Yes입니다. 예:

```
DisplayLabels=Yes
```

## DisplayDescriptions

이 키워드는 보고서에 설명을 표시할지 여부를 지정합니다.

### 구문

```
DisplayDescriptions=Boolean
```

*Boolean*을 Yes 또는 No로 바꿉니다. 기본값은 No입니다. 예:

```
DisplayDescriptions=Yes
```

## DisplayPlugElimAccts

이 키워드는 내부거래 트랜잭션의 영향을 받는 플러그 계정 요약을 표시할지 여부를 지정합니다.

### 구문

```
DisplayPlugElimAccts=Boolean
```

*Boolean*을 Yes 또는 No로 바꿉니다. 예:

```
DisplayPlugElimAccts=Yes
```

## GroupByCustom

이 키워드는 사용자정의 차원 트랜잭션을 차원별로 그룹화합니다. 이 키워드는 선택 사항입니다.

### 구문

```
GroupByCustom=Custom
```

보고서의 사용자정의 차원을 그룹화할 사용자정의 차원 별칭으로 *Custom*을 바꿉니다. 예:

```
GroupByCustom=Flows
```

# 내부거래 트랜잭션 보고서 스크립트 정의

다음 유형의 내부거래 트랜잭션 보고서를 정의할 수 있습니다.

- IC 트랜잭션 - 트랜잭션 목록을 생성합니다.
- 계정별 내부거래 일치 - 선택한 계정을 기준으로 일치 보고서를 생성합니다.
- ID별 내부거래 일치 - 트랜잭션 ID를 기준으로 일치 보고서를 생성합니다.



**표 9-1 내부거래 트랜잭션 보고서 스크립트의 키워드**

키워드	설명
ReportType	보고서 유형을 지정합니다 (예: ReportType=ICTransactions).
ReportLabel	보고서 유형을 지정합니다 (예: ReportLabel=트랜잭션 보고서).
ReportDescription	보고서 유형을 지정합니다 (예: ReportDescription=내부거래 트랜잭션 세부정보 보고서).
ReportSecurityClass	보고서의 보안 클래스를 지정합니다. 기본값은 [Default]입니다.
POV	보고서에 적합한 POV를 지정합니다 (예: POV=S#ActMon.Y#2014.P#January)
ScaleFactor	보고서의 스케일 요소를 지정합니다. 스케일은 0에서 9 사이의 값일 수 있습니다.
소수 자릿수	보고서에 표시할 소수 자릿수를 지정합니다. 소수 자릿수는 0에서 9 사이의 값일 수 있습니다.
FromAmt 및 ToAmt	트랜잭션 금액의 범위를 지정합니다.
IncludeMatched	일치하는 트랜잭션을 보고서에 포함하려면 True를 지정하고, 그렇지 않으면 False를 지정합니다.
IncludeUnMatched	대응 취소된 트랜잭션을 보고서에 포함하려면 True를 지정하고, 그렇지 않으면 False를 지정합니다.
IncludeMisMatched	일치하지 않는 트랜잭션을 보고서에 포함하려면 True를 지정하고, 그렇지 않으면 False를 지정합니다.
IncludePosted	게시된 트랜잭션을 보고서에 포함하려면 True를 지정하고, 그렇지 않으면 False를 지정합니다.
IncludeUnPosted	게시되지 않은 트랜잭션을 보고서에 포함하려면 True를 지정하고, 그렇지 않으면 False를 지정합니다.
엔티티	엔티티는 상위 엔티티가 아니라 적합한 ICP 기본 엔티티여야 합니다.
Partner	파트너 엔티티는 계정에 적합한 ICP 엔티티여야 합니다.
Entity Account	엔티티 트랜잭션이 표시되는 경우 엔티티 계정을 지정합니다.
Partner Account	파트너 트랜잭션이 표시되는 경우 파트너 계정을 지정합니다.
TransactionID	이 ID는 필수입니다. 트랜잭션 ID를 최대 40자까지 입력해야 합니다. 이 ID는 하위 ID와 조합되어 시나리오/연도/기간 내에서 엔티티/파트너/계정/C1/C2의 고유 ID를 생성합니다.
TransactionSubID	트랜잭션 하위 ID를 지정합니다.

표 9-1 (계속) 내부거래 트랜잭션 보고서 스크립트의 키워드

키워드	설명
TransactionCurrency	송장 트랜잭션에 사용되는 통화입니다. 애플리케이션에 정의된 적합한 통화여야 합니다.
ReferenceID	선택사항입니다. 참조 ID를 입력하여 트랜잭션에 대한 참조 정보를 저장할 수 있습니다. 예를 들어 엔티티에서 송장을 발급한 엔티티와는 다른 고유한 송장 번호 지정 세트를 사용할 수도 있습니다. 정보 제공 목적으로만 사용되는 추가 정보를 이 참조 ID에 입력할 수 있습니다. [트랜잭션 ID]에 엔티티 참조 ID를 입력하고 파트너 엔티티의 해당 송장 번호를 입력할 수 있습니다.
MatchCode	선택사항입니다. 일치 코드는 각 일치 프로세스 유형을 구분하는 다음 접두어 중 하나여야 합니다. <ul style="list-style-type: none"> <li>• A - 계정을 사용하여 수행된 자동 일치</li> <li>• I - 트랜잭션 ID를 사용하여 수행된 자동 일치</li> <li>• R - 참조 ID를 사용하여 수행된 자동 일치</li> <li>• M - 수동 일치</li> </ul>
ReasonCode	선택사항입니다. 원인 코드는 관리자가 정의한 적합한 원인 코드여야 합니다. 원인 코드는 주로 파트너 엔티티의 송장 누락이나 파트너가 부정확한 금액 입력 등 트랜잭션이 [일치하지 않음] 상태가 된 원인을 나타내는 데 사용됩니다. 트랜잭션이 [일치함] 상태인 경우에는 트랜잭션에 대해 원인 코드를 지정할 필요가 없습니다. [대응 취소함] 상태의 트랜잭션에는 원인 코드를 지정할 수 없습니다.
FromDate	선택사항입니다. 적합한 날짜여야 합니다.
ToDate	선택사항입니다. 적합한 날짜여야 합니다.
DisplayColumns Section	보고서에 표시되는 열과 열 표시 방식을 지정합니다. 구문은 <DisplayColumns>, <Sort>, <Label>, <Repeat>, <Totals>입니다. 여기서 <Displayed Columns>는 적합한 열이고, <Sort>는 Sort 또는 NoSort이고, <Label>은 레이블, 설명 또는 둘 다이고, <Repeat>는 Repeat 또는 NoRepeat이고, <Total>은 Total 또는 NoTotal입니다. 사용자정의 차원의 경우 별칭은 열에 사용됩니다. 예를 들면 다음과 같습니다. Customers, NoSort, Label, Repeat, NoTotal

**표 9-1 (계속) 내부거래 트랜잭션 보고서 스크립트의 키워드**

키워드	설명
DisplayEntityTransactions	엔티티 트랜잭션을 보고서에 표시하려면 True를 지정하고, 그렇지 않으면 False를 지정합니다. 특정 엔티티와 파트너의 내부거래 트랜잭션만 표시하도록 선택하거나, 파트너가 엔티티와 수행한 해당 트랜잭션도 표시할 수 있습니다. 예를 들어 엔티티 및 파트너 선택에서 엔티티 A와 파트너 B의 트랜잭션만 선택하면 엔티티 A가 파트너 B와 수행한 트랜잭션만 표시됩니다. 하지만 엔티티 B와 파트너 A의 해당 트랜잭션을 보려는 경우 엔티티 트랜잭션과 파트너 트랜잭션을 포함하도록 선택할 수 있습니다.
DisplayPartnerTransactions	파트너 트랜잭션을 보고서에 표시하려면 True를 지정하고, 그렇지 않으면 False를 지정합니다.
SuppressDetails	트랜잭션 세부정보를 제외하고 소계 행만 표시하려면 True를 지정합니다.

# 10

## 규칙 관리

Oracle Hyperion Financial Management 규칙을 사용하여 애플리케이션 내의 데이터 계산을 자동화합니다. 다음과 같은 용도로 규칙을 사용할 수 있습니다.

- 특정 엔티티, 시나리오 및 기간의 데이터 입력 레벨 금액을 계산합니다.
- 특정 엔티티, 시나리오 및 기간에서 특정 셀의 데이터 입력을 방지합니다.
- 상위 엔티티 레벨에서 입력을 허용합니다.
- 계층 합산을 통해 계산할 수 없는 비율, 차이 분석 등의 데이터를 계산합니다.
- 상위 엔티티에서 기본 엔티티 목록으로 할당합니다.
- 복잡한 통화 변환을 수행하거나, 환율 차이를 계산하거나, 통합에 필요한 기타 계산을 수행합니다.
- 동적으로 계정을 계산하는 공식을 정의합니다.
- 내부거래 트랜잭션을 지원하는 애플리케이션 계정을 지정합니다.

Notepad ++ 등의 텍스트 편집기에서 규칙을 작성할 수 있습니다.

### 주:

Oracle Hyperion Calculation Manager의 규칙에 대해 작업하려면 [Calculation Manager](#)를 사용하여 규칙 생성을 참조하십시오.

## 규칙 유형

값을 설정 및 지우고, 데이터를 계산하고, 통화를 환산하고, 데이터를 통합하고, 한 위치의 데이터를 다른 위치에 할당하고, 데이터 입력을 방지하는 규칙을 작성할 수 있습니다.

**표 10-1** 규칙 유형

규칙 유형	설명
계산	사용자가 계산을 실행하면 계산 규칙이 실행됩니다. 계산 규칙을 사용하여 차원 계층의 기본 순서를 통해 계산할 수 없는 계산을 수행할 수 있습니다. 예를 들어 비율이나 기초 잔액을 파생시키는 계산을 생성할 수 있습니다. 데이터를 계산하거나 통합하는 경우 Calculate() 루틴이 실행됩니다.

표 10-1 (계속) 규칙 유형

규칙 유형	설명
환산	<p>사용자가 환산을 실행하면 환산 규칙이 실행됩니다. 환산 규칙을 사용하여 비표준 환산과 관련된 계산을 수행할 수 있습니다. 데이터를 환산하거나 통합하는 경우 Translate() 루틴이 실행됩니다.</p> <p>예를 들어 애플리케이션에서 기본 ClosingRate를 사용하여 자산 및 부채 계정을 환산하고, AverageRate를 사용하여 수익 및 비용 계정을 환산하는 경우 다른 환산율을 사용하여 순수익과 관련된 환산 차이를 계산할 수 있습니다.</p> <p>다음과 같은 경우 Oracle Hyperion Financial Management에서 환산 규칙을 실행합니다.</p> <ul style="list-style-type: none"> <li>• 사용자가 데이터 그리드를 마우스 오른쪽 버튼으로 누르고 표시되는 메뉴에서 [환산] 또는 [강제 환산]을 선택하여 통화 환산을 실행하는 경우. 이 기능은 데이터를 입력할 때 환산을 수행하는 경우에 유용합니다.</li> <li>• 사용자가 통합을 수행하고 상위 엔티티의 기본 통화가 1차 하위 구성요소 엔티티의 기본 통화와 다른 경우</li> </ul>
통합	<p>사용자가 통합을 실행하면 통합 규칙이 실행됩니다. 통합 규칙을 사용하여 법적 애플리케이션에서 가장 일반적으로 사용되는 비표준 통합을 수행할 수 있습니다. 데이터를 통합하는 경우 Consolidate() 루틴이 실행됩니다.</p>
할당	<p>할당 규칙은 한 엔티티의 데이터를 엔티티 목록에 할당합니다. 예를 들어 할당 규칙을 사용하여 다른 엔티티의 관리 비용 합계를 기준으로 엔티티 목록의 관리 비용을 계산할 수 있습니다.</p>
Input	<p>입력 규칙은 상위 엔티티 레벨에서 입력을 허용합니다. 엔티티 통화 값 차원만 지원됩니다. 상위 엔티티의 엔티티 통화 값 차원에 대한 1차 하위 구성요소의 기여금액 값은 롤업되지 않습니다. 상위.1차 하위 구성요소 조합의 기여금액 값이 저장됩니다.</p>

표 10-1 (계속) 규칙 유형

규칙 유형	설명
NoInput	<p>NoInput 규칙은 기본 엔티티 레벨에서 입력을 방지하여 계산에 사용할 셀을 예약합니다. NoInput 규칙에 NoInput 함수를 여러 번 사용하여 인접하지 않은 여러 셀에 대한 데이터 입력을 차단할 수 있습니다.</p> <p>NoInput 규칙에는 다음과 같은 제한과 가이드라인이 적용됩니다.</p> <ul style="list-style-type: none"> <li>• 다음과 같은 Financial Management 함수만 NoInput 규칙에서 지원됩니다. <ul style="list-style-type: none"> <li>- NoInput</li> <li>- List</li> </ul> <p>VBScript 메소드는 NoInput 규칙에서 모두 지원됩니다.</p> </li> <li>• List 함수의 경우 고정 목록, 시스템 목록 및 동적 목록이 지원됩니다. 동적 목록은 List 함수에서 지원되는 메타데이터 속성을 참조할 수 있습니다.</li> <li>• If...Then 구조에서 메타데이터 속성을 테스트할 수 있습니다. 하지만 NoInput 규칙은 현재 POV의 멤버 테스트를 지원하지 않습니다. 예를 들어 If HS.Entity.Member = "CT"는 테스트할 수 없습니다.</li> <li>• NoInput 함수를 루프에 사용할 때는 주의하십시오. 루프에 몇 개의 간단한 명령문만 사용해도 수천 개의 셀이 메모리에 로드될 수 있으므로 NoInput이 포함된 루프가 성능에 미치는 영향을 테스트해야 합니다.</li> </ul>
동적 계산	<p>동적 규칙을 사용하면 계정을 동적으로 계산하는 공식을 정의할 수 있습니다. 기존 계정만 동적으로 계산할 수 있습니다. 상위 계정에는 동적 규칙을 사용할 수 없습니다.</p> <p>동적 계산 규칙 작성에 대한 다음 가이드라인을 따르십시오.</p> <ul style="list-style-type: none"> <li>• 등식의 오른쪽에서는 동일한 시나리오/연도/엔티티 조합을 참조해야 합니다. 즉, 계산에서 이전 연도 금액을 참조할 수 없습니다.</li> <li>• 등식의 왼쪽에서는 동적 계정만 적합합니다.</li> <li>• 등식의 오른쪽에서는 동적 계정을 사용할 수 없습니다.</li> <li>• 등식의 왼쪽에서는 계정과 뷰만 적합합니다.</li> <li>• 뷰를 지정하지 않으면 YTD와 주기적에 대해 계산이 실행됩니다. 뷰를 지정하면 지정된 뷰에 대해서만 계산이 실행됩니다.</li> <li>• HS.View.PeriodNumber는 HS.Dynamic 계산에 사용할 수 있는 유일한 HS 문입니다.</li> <li>• Sub Dynamic 섹션에 있는 모든 명령문은 순서대로 실행됩니다.</li> </ul>

표 10-1 (계속) 규칙 유형

규칙 유형	설명
트랜잭션	트랜잭션 규칙은 내부거래 트랜잭션을 지원하는 애플리케이션의 계정을 지정합니다. 트랜잭션을 지원하는 셀은 데이터 그리드와 양식에서 읽기 전용입니다.
지분 매입	지분 매입 규칙은 소유 엔티티, 소유자 엔티티 및 소유 지분을 지정합니다. Sub EquityPickup 섹션을 실행하는 경우 기본 POV는 다음과 같습니다. <ul style="list-style-type: none"> <li>• 현재 시나리오, 연도 및 기간</li> <li>• 엔티티: 처리된 쌍의 소유자</li> <li>• 값: &lt;Entity Currency&gt;</li> </ul>
주문형	주문형 규칙은 데이터 양식에서 사용되며, 계산 서브세트를 실행하여 데이터 양식에서 결과를 빠르게 볼 수 있도록 합니다. Sub Calculate에서 사용할 수 있는 모든 HS 함수(다른 함수 제외)는 주문형 규칙에도 사용할 수 있습니다.

Microsoft VBScript 언어를 기반으로 하는 고유한 스크립트에 Financial Management 규칙을 생성합니다. 규칙은 스크립팅 구문을 생성하는 함수, 객체 및 기타 인수의 조합을 통해 생성됩니다. 각 루틴 내에서 다음 두 가지 유형의 함수를 사용하여 규칙을 작성합니다.

- 계산을 정의하는 Financial Management 함수
- 조건부 명령문을 작성하는 데 사용되는 VBScript 함수

## 규칙 고려 사항

다음은 애플리케이션 규칙을 작성할 때 고려할 사항입니다.

- 규칙은 사용자가 계산, 환산, 통합 및 할당을 수행할 때 실행됩니다. 계산이나 통합이 적용되는 각 엔티티/값 차원 교차에 대해 한 번씩 계산 규칙이 실행됩니다. [계산 명령이 포함된 계산 규칙](#)을 참조하십시오.
- Oracle Hyperion Financial Management에서 계산을 적용하는 차원 멤버는 사용자 커서가 놓여 있는 데이터 그리드 셀과 POV 막대에서 지정된 멤버에 따라 달라집니다. [현재 차원 멤버](#)를 참조하십시오.
- 함수가 통화 관련 값 멤버에 데이터를 넣으면 Financial Management에서 값 멤버의 현재 값을 삭제할 수 있습니다. [함수를 통해 자동으로 데이터 지우기](#)를 참조하십시오.
- 규칙은 루틴 내에서 순서대로 실행되며, 특정 루틴이 다른 루틴보다 먼저 실행되는 정해진 순서가 있습니다. [통합 중 규칙 실행](#)을 참조하십시오.

## 계산 명령

계산 루틴은 시나리오, 연도, 기간, 엔티티 및 값의 지정된 교차에 대해 실행됩니다. 계산 루틴을 실행하면 시나리오, 연도, 기간, 엔티티 및 값의 현재 교차에 해당하는 데이터 테이블에서 데이터가 기록되거나 지워집니다. 사용자가 계산 루틴을 실행하면 시스템에서 애플리케이션의 임의 위치에 있는 데이터를 읽을 수 있습니다. 하지만 현재 데이터 테이블에만 데이터가 기록됩니다.

 주:

통합 시 계층에 [None] 엔티티가 있으면 계산 루틴이 실패합니다.

다음 명령 중 하나를 선택하여 데이터 그리드에서 이러한 프로세스를 실행합니다.

- 계산
- 강제 계산
- 기여금액 계산
- 기여금액 강제 계산
- 통합
- 데이터가 있는 항목 모두 통합
- 모두 통합
- 환산
- 강제 환산

계산 명령을 선택하면 Oracle Hyperion Financial Management에서 RLE 파일의 Sub Calculate() 루틴을 실행합니다. Calculate() 루틴은 지정된 시나리오, 연도 및 기간 내의 지정된 엔티티-값 조합에 대해 계정과 사용자정의 차원 멤버를 계산합니다.

## 강제 계산

[강제 계산] 옵션을 사용하면 선택한 값 멤버와 이 멤버가 종속된 값 멤버에서만 규칙이 강제로 실행됩니다. 예를 들어 엔티티 통화에 [강제 계산]을 사용하면 엔티티 통화 멤버에서만 규칙이 실행됩니다. 엔티티 통화 합계에 [강제 계산]을 사용하면 엔티티 통화, 엔티티 통화 조정 및 엔티티 통화 합계에서 규칙이 실행됩니다. [강제 계산]은 값 3쌍의 단일 멤버에만 적용할 수 있는 유일한 명령입니다.

## 계산 명령이 포함된 계산 규칙

계산 규칙이 실행되는 횟수는 사용자 및 다른 요소에서 선택한 계산 명령에 따라 달라집니다.

 주의:

규칙을 작성하기 전에 다음 섹션의 내용을 주의해서 읽어 보십시오. 값 차원의 특정 멤버에 대해서만 작업이 수행되도록 할 수도 있으며, 이 경우 작업을 실행하기 전에 VBScript의 If 구조와 Member 함수를 사용하여 현재 멤버를 테스트해야 합니다. If 구조 및 Member 함수 사용에 대한 단계와 예는 [조건부 규칙](#)을 참조하십시오.

사용자가 [계산] 또는 [강제 계산] 명령을 선택하면 Oracle Hyperion Financial Management에서 엔티티 기본 통화의 값 멤버인 엔티티 통화와 현재 엔티티 멤버의 교차점에 대해 애플리케이션의 계산 규칙을 실행합니다. 메타데이터에서 엔티티의 AllowAdjs 속성이 활성화된 경우 Financial Management에서 두 번째로 규칙을 실행하여 엔티티 기본 통화의 조정 내용이 저장되는 값 차원 멤버인 엔티티 통화 조정과 엔티티의 교차에도 규칙을 적용합니다.



**예**

예를 들어 California라는 엔티티의 기본 통화가 USD이고 메타데이터에서 해당 AllowAdjs 속성이 활성화된 경우 계산 규칙이 California와 USD의 교차 및 California와 USD Adjs의 교차에 대해 각각 한 번씩 모두 두 번 실행됩니다.

**통합 명령이 포함된 계산 규칙**

사용자가 통합 명령 중 하나를 선택하면 Oracle Hyperion Financial Management에서 이전에 통합 취소된 각 1차 하위 엔티티와 교차하는 여러 값 차원에 대해 계산 규칙을 실행합니다. Financial Management는 1차 하위 구성요소에 대해 실행한 후 상위 엔티티와 상위 기본 통화에 대한 값 차원 멤버의 교차에 대해 규칙을 실행합니다.

1. 1차 하위 구성요소 엔티티의 기본 통화를 저장하는 값 멤버(엔티티 통화)와 1차 하위 구성요소 엔티티의 교차에 대해 규칙이 실행됩니다.
2. 1차 하위 구성요소 엔티티의 AllowAdjs 속성이 Y로 설정된 경우 1차 하위 구성요소 엔티티의 기본 통화 조정 내용을 저장하는 값 멤버(엔티티 통화 조정)와 1차 하위 구성요소 엔티티의 교차에 대해 규칙이 실행됩니다.
3. 1차 하위 구성요소의 통화가 상위의 통화와 다른 경우 상위의 홈 통화로 환산된 금액을 저장하는 값 멤버(상위 통화)와 1차 하위 구성요소 엔티티의 교차에 대해 규칙이 실행됩니다.
4. 1차 하위 구성요소의 통화가 상위의 통화와 다르고 규칙 파일에 환산 규칙이 포함되어 있는 경우 Financial Management에서 3단계 전에 환산 규칙을 실행합니다.
5. 1차 하위 구성요소의 통화가 상위의 통화와 다르고 1차 하위 구성요소 엔티티의 AllowAdjs 속성이 Y로 설정된 경우 상위의 홈 통화로 환산된 조정 내용을 저장하는 값 멤버(상위 통화 조정)와 1차 하위 구성요소 엔티티의 교차에 대해 규칙이 실행됩니다.
6. 상위 엔티티의 AllowAdjFromChildren 속성이 Y로 설정된 경우 1차 하위 구성요소 엔티티와 상위 조정 값의 교차에 대해 논리 규칙이 실행됩니다.
7. 1차 하위 구성요소 엔티티와 비례 값의 교차에 대해 규칙이 실행됩니다.
8. 1차 하위 구성요소 엔티티와 제거 값의 교차에 대해 규칙이 실행됩니다.
9. 상위 엔티티의 AllowAdjFromChildren 속성이 Y로 설정된 경우 1차 하위 구성요소 엔티티와 기여금액 조정 값의 교차에 대해 규칙이 실행됩니다.
10. 이전에 통합 취소된 데이터가 포함된 각 추가 1차 하위 구성요소 엔티티에 대해 1-9단계를 반복합니다.
11. 상위 엔티티의 기본 통화를 저장하는 값 멤버와 상위 엔티티의 교차에 대해 규칙이 실행됩니다.

**예**

예를 들어 UnitedStates라는 상위 엔티티에 EastUS 및 WestUS라는 1차 하위 구성요소가 있습니다. 1차 하위 구성요소에서 AllowAdjs 속성이 활성화되어 있습니다. 1차 하위 구성요소에서 AllowAdjs 속성이 활성화되어 있습니다. 세 엔티티는 모두 기본 통화 USD를 공유합니다.

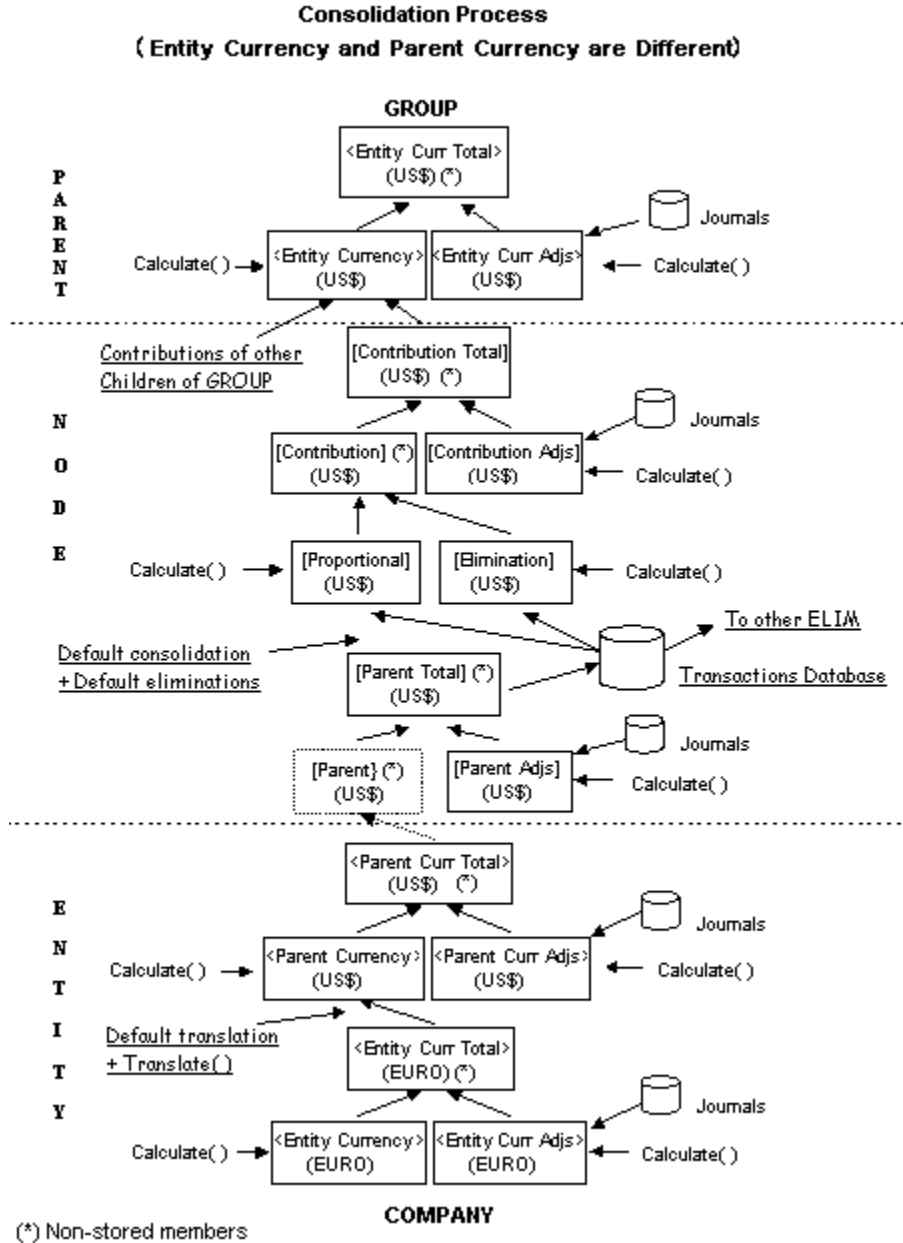
EastUS 및 WestUS의 데이터를 변경하고 UnitedStates를 통합하면 Financial Management에서 엔티티 및 값 차원의 다음 교차에 대해 각각 규칙을 실행합니다.

1. EastUS 및 엔티티 통화
2. EastUS 및 엔티티 통화 조정 (EastUS의 AllowAdjs 속성이 Y로 설정된 경우)

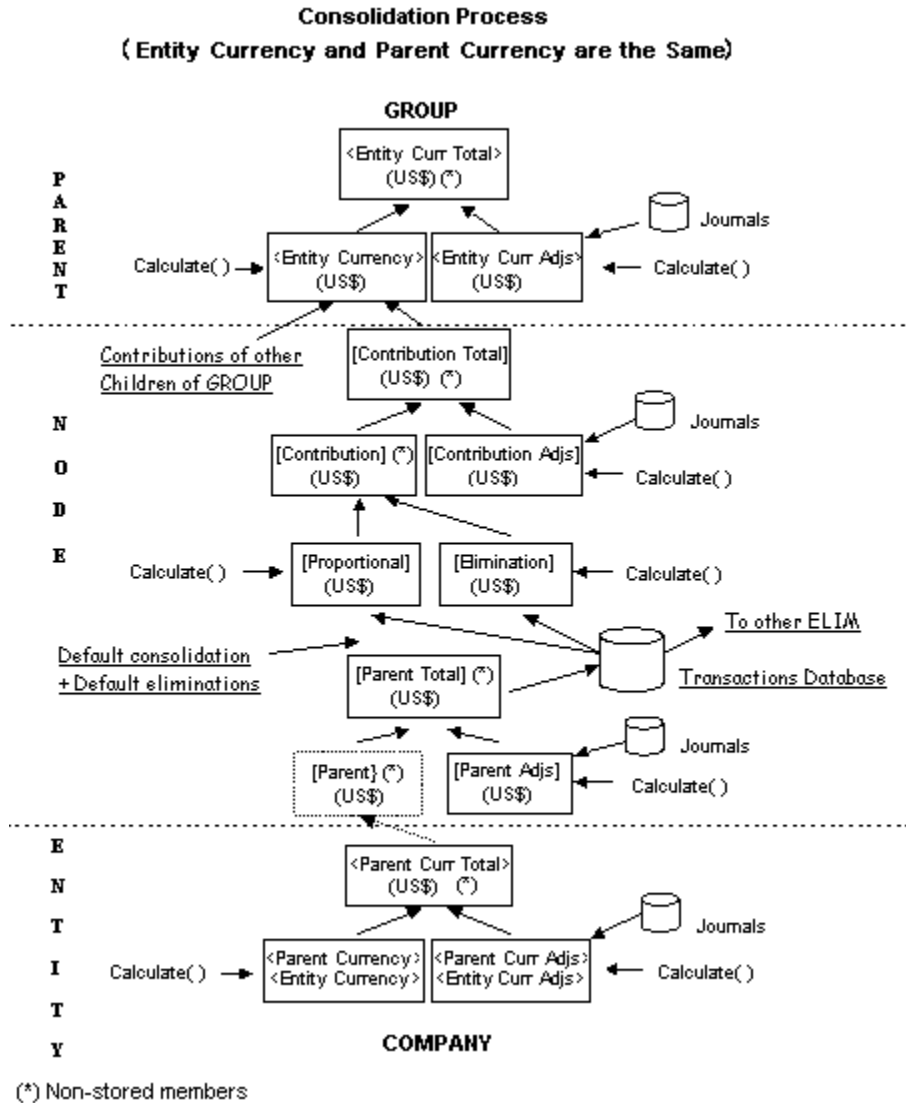
3. EastUS 및 상위
4. EastUS 및 상위 조정 (UnitedStates의 AllowAdjFromChildren 속성이 Y로 설정된 경우)
5. EastUS 및 비례
6. EastUS 및 제거
7. EastUS 및 기여금액 조정 (UnitedStates의 AllowAdjFromChildren 속성이 Y로 설정된 경우)
8. WestUS 및 엔티티 통화
9. WestUS 및 엔티티 통화 조정 (WestUS의 AllowAdjs 속성이 Y로 설정된 경우)
10. WestUS 및 상위
11. WestUS 및 상위 조정 (UnitedStates의 AllowAdjFromChildren 속성이 Y로 설정된 경우)
12. WestUS 및 비례
13. WestUS 및 제거
14. WestUS 및 기여금액 조정 (UnitedStates의 AllowAdjFromChildren 속성이 Y로 설정된 경우)
15. UnitedStates 및 엔티티 통화

다음은 통합 프로세스의 예입니다.

첫 번째 예에서는 엔티티 통화와 상위 통화가 서로 다른 경우의 프로세스를 보여 줍니다.



다음 예에서는 엔티티 통화와 상위 통화가 같은 경우의 프로세스를 보여 줍니다.



## 현재 차원 멤버

기본적으로 Oracle Hyperion Financial Management에서는 사용자가 계산 명령을 선택할 때 현재 차원 멤버에 계산 규칙을 적용합니다. **계산 명령이 포함된 계산 규칙**을 참조하십시오. 현재 차원 멤버는 다음과 같은 요소에 의해 결정됩니다.

- 사용자가 계산, 환산 또는 통합을 실행할 때 사용자 커서가 놓여 있는 데이터 그리드의 셀
- POV 막대에서 지정된 차원 멤버
- 통합의 경우 규칙을 실행할 때마다 엔티티 및 값 차원의 현재 멤버가 변경됩니다. **통합 명령이 포함된 계산 규칙**을 참조하십시오.
- 계산 규칙이 애플리케이션의 모든 계정을 처리하는 계산 규칙을 제외하고 규칙은 현재 선택한 차원 멤버를 처리합니다.

커서가 놓여 있는 셀의 차원 멤버가 POV 막대에 있는 해당 차원의 해당 멤버와 다른 경우 셀의 차원 멤버가 POV 막대의 차원 멤버를 재정의합니다. 예를 들어 사용자가 POV 막대에서 Europe

엔티티를 지정했는데 Germany 엔티티의 셀에 커서를 놓고 환산을 실행하면 Germany가 현재 엔티티입니다.

차원 멤버를 인수로 지정하면 현재 차원에 관계없이 일부 함수를 사용하여 특정 차원 멤버에 대해 작업할 수 있습니다. 예를 들어 NumBase 함수에 엔티티 객체를 사용하여 인수에 지정한 엔티티의 기본 엔티티 수를 가져올 수 있습니다.

더 복잡한 함수의 경우 계정 표현식을 생성하여 차원 멤버를 지정할 수 있습니다.

## 계정 표현식

일부 함수는 인수로 계정 표현식이 있어야 합니다. 가장 간단한 형태의 계정 표현식은 Oracle Hyperion Financial Management에서 함수를 적용하는 계정을 지정하는 문자열입니다.

다음 표에서는 계정 표현식 문자를 보여 줍니다.

**표 10-2** 계정 표현식에서 차원을 지정하는 데 사용되는 문자

문자	차원
A#	계정
I#	내부거래 파트너
C#	사용자정의
S#	시나리오
Y#	연도
P#	기간
W#	뷰
E#	엔티티
V#	값

계정 표현식을 이해하려면 함수의 인수에 지정된 차원 멤버에서 값을 제거하는 Clear 함수를 고려해 보십시오. 다음 예에서는 인수 "A#Sales"가 계정 표현식입니다.

```
HS.Clear "A#Sales"
```

A# 문자는 계정 차원을 나타내고 단어 Sales는 Financial Management에서 Clear 함수를 적용하는 계정 차원의 멤버 이름입니다. 이 계정 표현식은 Sales 계정에 저장된 데이터를 지우도록 Financial Management에 지정합니다.

계정 표현식을 사용하면 Financial Management에서 다음 차원의 현재 멤버와 계정 표현식에 지정된 계정의 교차에 함수를 적용합니다.

- 엔티티
- 기간
- 시나리오
- 값
- 뷰
- 연도

- 사용자정의 - 메타데이터에서 계정에 대해 설정된 CustomTopMember를 사용합니다. 예를 들어 계정 표현식에서 사용자정의 3 차원의 멤버를 지정하지 않는 경우 Financial Management는 계정에 지정된 CustomTopMember가 정의한 적합한 모든 사용자정의 3 멤버를 사용합니다.
- 내부거래 파트너 - 적합한 모든 ICP 멤버입니다.

계정 표현식의 멤버를 지정하여 내부거래 및 사용자정의 차원 기본값을 재정의할 수 있습니다. 각 차원은 특정 문자로 표시됩니다. 계정 표현식에 둘 이상의 차원을 포함하는 경우 차원을 마침표로 구분해야 합니다.

계정 표현식을 생성할 때 이러한 차원 멤버를 모두 지정할 필요는 없습니다. 함수를 적용할 멤버를 지정할 수 있습니다. 예를 들어 다음 함수는 Sales 계정과 Hardware ICP의 교차를 지웁니다.

```
HS.Clear "A#Sales.I#Hardware"
```

## 함수를 통해 자동으로 데이터 지우기

함수가 통화 관련 값 멤버에 데이터를 넣는 경우 다음 조건 중 하나가 적용되면 Oracle Hyperion Financial Management에서 멤버의 데이터를 자동으로 지웁니다.

- 교차하는 엔티티 멤버가 상위인 경우
- 교차하는 계정 멤버가 계산된 계정인 경우

### 팁:

통화 관련 값 멤버는 시스템 생성 엔티티 통화 멤버 및 USD, EURO, LIRA 등의 사용자 정의 통화 멤버입니다.

또한 함수가 조정을 위해 값 멤버에 데이터를 넣는 경우 교차하는 계정 멤버가 계산된 계정이면 Financial Management에서 멤버의 데이터를 자동으로 지웁니다.

### 팁:

조정 관련 값 멤버는 이름에 코드 "Adjs"가 포함된 멤버입니다(예: USD Adjs, EURO Adjs, LIRA Adjs).

## 오류 메시지

Oracle Hyperion Financial Management에서 구문 오류를 발견하면 다음 정보가 포함된 오류 메시지를 표시합니다.

- 오류를 발생시키는 RLE 파일의 라인 번호
- 해당되는 Financial Management 객체 및 함수
- 오류 설명

## 통합 중 규칙 실행

통합 프로세스에서 규칙은 사전 정의된 순서로 실행됩니다. 지정된 상위 멤버의 각 기본 하위 멤버에 대해 값 차원에 있는 여러 요소의 계산 순서가 다음 순서로 실행됩니다.

1. 메타데이터에서 IsCalculated로 정의된 계정이 EntityCurrency에서 지워집니다.
2. 메타데이터에서 IsCalculated로 정의된 계정이 EntityCurrAdjs에서 지워집니다.
3. Sub Calculate() 루틴이 EntityCurrency에서 실행됩니다.
4. Sub Calculate() 루틴이 EntityCurrAdjs에서 실행됩니다.
5. ParentCurrency 데이터가 지워집니다.
6. EntityCurrency 및 EntityCurrAdjs의 합계 금액에 대해 수익, 비용, 자산, 부채로 정의된 모든 계정에 기본 환산이 적용됩니다. Flow 또는 Balance 속성이 있는 계정의 경우 환산이 기본적으로 적용되지 않으며 EntityCurrency 및 EntityCurrAdjs의 합계 금액이 상위 통화로 롤업됩니다.
7. Sub Translate() 루틴이 실행됩니다.
8. Sub Calculate() 루틴이 ParentCurrency에서 실행됩니다.
9. 메타데이터에서 "IsCalculated"로 정의된 계정이 ParentCurrAdjs에서 지워집니다.
10. Sub Calculate() 루틴이 ParentCurrAdjs에서 실행됩니다.
11. 메타데이터에서 "IsCalculated"로 정의된 계정이 ParentAdjs에서 지워집니다.
12. Sub Calculate() 루틴이 ParentAdjs에서 실행됩니다.
13. Proportion 및 Elimination 데이터가 지워집니다.
14. Parent 및 ParentAdjs의 합계 금액에 대해 기본 통합과 제거가 수행됩니다.
15. Sub Calculate() 루틴이 Proportion 및 Elimination에서 실행됩니다.
16. 메타데이터에서 "IsCalculated"로 정의된 계정이 ContributionAdjs에서 지워집니다.
17. Sub Calculate() 루틴이 ContributionAdjs에서 실행됩니다.

각 기본 1차 하위 구성요소에 대해 이전 단계가 반복된 후 상위 엔티티에 대해 다음 시퀀스가 수행됩니다.

1. EntityCurrency 데이터가 지워집니다.
2. 모든 1차 하위 구성요소에 대한 Proportion, Elimination 및 ContributionAdjs 합계의 합계가 상위 엔티티의 EntityCurrency에 기록됩니다.
3. Sub Calculate() 루틴이 EntityCurrency에서 실행됩니다.
4. 메타데이터에서 "IsCalculated"로 정의된 계정이 EntityCurrAdjs에서 지워집니다.
5. Sub Calculate() 루틴이 EntityCurrAdjs에서 실행됩니다.

### 주:

상위가 다른 상위에 추가로 통합되어 있는 경우 이 시퀀스는 1차 하위 구성요소 통합 시퀀스의 5단계에서 계속됩니다.

## 기본 환산

다음은 기본 환산이 수행되는 순서입니다.

1. 시스템이 현재 엔티티의 직접 환산율을 확인하고 이 환율을 환산에 사용합니다.
2. 환산율을 찾을 수 없는 경우 현재 엔티티의 간접 환율에서 직접 환율이 파생됩니다.
3. 현재 엔티티에서 직접 환율과 간접 환율을 모두 찾을 수 없는 경우 시스템이 [None] 엔티티를 확인하여 직접 환율을 사용합니다.
4. [None] 엔티티에서 직접 환율을 찾을 수 없는 경우 [None] 엔티티의 간접 환율에서 직접 환율이 파생됩니다.
5. [None] 엔티티에 대한 간접 환율이 없는 경우 [None] 엔티티의 애플리케이션 통화를 사용하는 삼각 분할에 의해 환율이 파생됩니다.
6. 삼각 분할이 실패하는 경우 엔티티가 환산되지 않습니다.

### 주:

시스템은 먼저 현재 엔티티 내에서 환산율을 찾습니다. 현재 엔티티에서 찾을 수 없는 경우 [None] 엔티티 내에서 환산율을 찾습니다. [None] 엔티티에서 환산율을 찾을 수 없는 경우 삼각 분할을 사용하여 환산합니다. 삼각 분할은 제3의 공통 통화를 사용하여 한 통화에서 다른 통화로 잔액을 변환하는 방법입니다.

예를 들어 Euro를 Yen으로 변환하려고 하는데 시스템이 환산을 수행할 직접 또는 간접 환율을 찾을 수 없는 경우 Euro와 Yen이 삼각 분할을 사용하여 모두 USD로 환산될 수 있으면 시스템에서 Euro 잔액을 USD로 변환한 다음 USD 잔액을 Yen으로 변환할 수 있습니다.

## Financial Management 객체

이 섹션에서는 함수 사용 시 Oracle Hyperion Financial Management 객체를 나타내기 위해 사용해야 하는 구문에 대해 설명합니다.

Financial Management의 최상위 레벨 객체는 HS 객체입니다. 따라서 Financial Management 함수를 사용하는 경우 처음 세 문자는 다음과 같이 글자 HS와 마침표여야 합니다.

HS.

다음 예에서는 Clear 함수를 작성하는 방법을 보여 줍니다.

HS.Clear

Clear 앞에 HS. 문자를 사용하지 않으면 오류가 발생합니다.

### 주의:

표준 VBScript 함수 앞에는 HS. 문자를 사용할 수 없습니다. 사용하면 오류가 발생합니다. Financial Management 함수 앞에만 HS. 문자를 사용하십시오.



일부 Financial Management 함수는 HS 객체의 1차 하위 구성요소인 객체에만 적용됩니다. 이러한 함수를 사용하려면 적용 가능한 객체 이름을 HS 문자와 함수 이름 사이에 넣어야 합니다. 다음 객체는 HS 객체의 1차 하위 구성요소이며 Financial Management에서 사용됩니다.

- 계정
- AppSettings
- 사용자정의
- DataUnit
- 엔티티
- ICP
- 노드
- 상위
- 기간
- 시나리오
- 값
- 연도

## 규칙에서 VBScript 사용

규칙에서 조건부 명령문을 작성하는 데 사용되는 VBScript 함수입니다.

### 주:

Oracle Hyperion Financial Management에서는 COM 스크립트 가능 인터페이스로 노출되는 .NET 객체를 지원하지 않습니다. VBScript 같은 COM 지원 애플리케이션에서는 규칙 또는 멤버 목록을 통해 해당 객체에 액세스할 수 없습니다.

## VBScript 연산자

Oracle Hyperion Financial Management 규칙에서 지원되는 VBScript 연산자는 다음과 같습니다.

- And
- Eqv
- Imp
- Is
- Mod
- Not
- Or
- Xor

- =
- <
- >
- <=
- >=
- <>
- +
- &
- -
- /
- \
- \*
- ^

## VBScript 문

Oracle Hyperion Financial Management 규칙에서 지원되는 VBScript 문은 다음과 같습니다.

- Call
- Const
- Dim
- Do...Loop
- Erase
- Exit
- For...Next
- ForEach...Next
- Function...End Function
- If...Then...Else If...Else...End If
- On Error Go to 0
- On Error Resume Next
- Option Explicit
- Private
- Public
- ReDim
- ReDim Preserve
- SelectCase...Case...End Select
- Set
- While...WEnd
- With...End With

- '
  - Sub...EndSub

## VBScript 키워드

Oracle Hyperion Financial Management 규칙에서 지원되는 VBScript 키워드는 다음과 같습니다.

- Empty
- Nothing
- Null
- True
- False

## VBScript 함수

Oracle Hyperion Financial Management 규칙에서 지원되는 VBScript 함수는 다음과 같습니다.

### 날짜 및 시간 함수

함수	설명
Cdate	적합한 날짜 및 시간 표현식을 Date 하위 유형의 변형으로 변환합니다.
Date	현재 시스템 날짜를 반환합니다.
DateDiff	두 날짜 사이의 간격 수를 반환합니다.
DatePart	DatePart(interval,date[,firstdayofweek[,firstweekofyear]])
DateSerial	DateSerial(year,month,day)
Day	일을 나타내는 1에서 31(포함) 사이의 숫자를 반환합니다.
Hour	시간을 나타내는 0에서 23(포함) 사이의 숫자를 반환합니다.
IsDate	평가된 표현식을 날짜로 변환할 수 있는지를 나타내는 부울 값을 반환합니다.
Minute	분을 나타내는 0에서 59(포함) 사이의 숫자를 반환합니다.
Month	월을 나타내는 1에서 12(포함) 사이의 숫자를 반환합니다.
Month/Name	지정된 월의 이름을 반환합니다.
Now	현재 시스템 날짜 및 시간을 반환합니다.
Second	초를 나타내는 0에서 59(포함) 사이의 숫자를 반환합니다.
Time	현재 시스템 시간을 반환합니다
Timer	오전 12시 이후의 시간(초)을 반환합니다.
연도	연도를 나타내는 숫자를 반환합니다.

## 형식 함수

함수	설명
FormatNumber	형식이 지정된 표현식을 숫자로 반환합니다.
FormatFunctions	형식이 지정된 표현식을 날짜 또는 시간으로 반환합니다.

## 변환 함수

함수	설명
Asc	문자열의 첫 번째 문자를 ANSI 코드로 반환합니다.
CBool	표현식을 Boolean 하위 유형의 변형으로 변환합니다.
CByte	표현식을 Byte 하위 유형의 변형으로 변환합니다.
CDbl	표현식을 Double 하위 유형의 변형으로 변환합니다.
Chr	지정된 ANSI 코드를 문자로 변환합니다.
Cint	표현식을 Integer 하위 유형의 변형으로 변환합니다.
CLng	표현식을 Long 하위 유형의 변형으로 변환합니다.
CStr	표현식을 String 하위 유형의 변형으로 변환합니다.

## 수리 함수

함수	설명
Abs	지정된 수의 절대값을 반환합니다.
Fix	지정된 수의 정수 부분을 반환합니다.
Int	지정된 수의 정수 부분을 반환합니다.
Rnd	1보다 작지만 0보다는 크거나 같은 임의의 수를 반환합니다.
Sgn	지정된 수의 부호를 나타내는 정수를 반환합니다.
Sqr	지정된 수의 제곱근을 반환합니다.

## 배열 함수

함수	설명
배열	배열이 포함된 변수를 반환합니다.
필터	필터 기준에 따라 문자열 배열의 서브세트가 포함된 0 기반 배열을 반환합니다.
IsArray	지정된 변수가 배열인지 여부를 나타내는 부울 값을 반환합니다.
조인	배열에 있는 많은 하위 문자열로 구성된 문자열을 반환합니다.
Lbound	배열의 지정된 차원에 사용할 수 있는 가장 낮은 아래 첨자를 반환합니다.
분할	지정된 개수의 하위 문자열이 포함된 0 기반 1차원 배열을 반환합니다.

함수	설명
Ubound	배열의 지정된 차원에 사용할 수 있는 가장 높은 아래 첨자를 반환합니다.

## 문자열 함수

함수	설명
InStr	다른 문자열 내에서 한 문자열의 첫 번째 발생 위치를 반환합니다. 문자열의 첫 문자에서 검색이 시작됩니다.
InStrRev	다른 문자열 내에서 한 문자열의 첫 번째 발생 위치를 반환합니다. 문자열의 첫 문자에서 검색이 시작됩니다.
Left	문자열 왼쪽에서 지정된 개수의 문자를 반환합니다.
Len	문자열의 문자 수를 반환합니다.
LTrim	문자열의 왼쪽에서 공백을 제거합니다.
RTrim	문자열의 오른쪽에서 공백을 제거합니다.
Trim	문자열의 왼쪽과 오른쪽에서 모두 공백을 제거합니다.
Mid	문자열에서 지정된 개수의 문자를 반환합니다.
바꾸기	지정된 문자열 부분을 지정된 횟수의 다른 하위 문자열로 바꿉니다.
Right	문자열 오른쪽에서 지정된 개수의 문자를 반환합니다.
Space	지정된 수의 공백으로 구성된 문자열을 반환합니다.
StrComp	두 문자열을 비교하고 비교 결과를 나타내는 값을 반환합니다.
StrReverse	문자열을 반대로 설정합니다.
LCase	지정된 문자열을 소문자로 변환합니다.
UCase	지정된 문자열을 대문자로 변환합니다.

## 기타 함수

함수	설명
CreateObject	지정된 유형의 객체를 생성합니다.
Eval	표현식을 평가하고 결과를 반환합니다.
IsEmpty	지정된 변수가 초기화되었는지 여부를 나타내는 부울 값을 반환합니다.
IsNull	지정된 표현식에 부적합한 데이터(Null)가 포함되어 있는지 여부를 나타내는 부울 값을 반환합니다.
IsNumeric	지정된 표현식을 숫자로 평가할 수 있는지 여부를 나타내는 부울 값을 반환합니다.
Round	숫자를 반올림합니다.

## VBScript 객체

Oracle Hyperion Financial Management 규칙에서 지원되는 VBScript 객체는 다음과 같습니다.

### 파일 시스템 객체

- CreateTextFile
- OpenTextFile
- DeleteFile
- FileExist
- FolderExist
- GetBaseName
- GetParentFolderName
- GetFile
- CopyFile

### 파일 객체

- OpenAsTextStream
- Size

### 텍스트 스트림 객체

- AtEndOfStream
- Close
- WriteLine
- ReadLine

### 오류 객체

- 설명
- HelpContext
- HelpFile
- Number
- 소스
- Raise
- Clear

## 일반적으로 사용되는 규칙

이 절에서는 일반적으로 사용되는 몇 가지 간단한 규칙을 작성하는 방법을 보여 줍니다. 이 규칙에 대한 설명에는 VBScript를 잘 모르는 독자를 위한 단계별 절차가 포함되어 있습니다. 절차 뒤에는 예가 있습니다. VBScript를 잘 아는 경우 절차를 건너뛰고 예를 중점적으로 살펴볼 수 있습니다.

 **팁:**

이 절차에서는 규칙 파일이 있고 작성한 코드가 Calculate() 서브루틴에 배치된다고 가정합니다. [규칙 파일 생성](#)을 참조하십시오.

## 데이터 재사용

EXP 함수를 사용하여 한 계정의 데이터를 다른 계정에 삽입할 수 있습니다. EXP의 인수에는 설정할 계정과 값이 검색되는 계정이 포함됩니다. 인수는 등식 형태를 사용하며 등호의 왼쪽에는 대상 계정이 오고 오른쪽에는 소스 계정이 옵니다.

 **주:**

EXP는 현재 차원 멤버와 계정의 교차점에 데이터를 삽입합니다([현재 차원 멤버 참조](#)). 이외에도 계정 표현식 문자를 사용하여 현재 사용자정의 멤버와 ICP 멤버를 재정의할 수 있습니다.

다음 예에서는 계산 규칙이 PrevCash 계정을 Cash 계정의 값으로 설정합니다.

```
HS.EXP "A#PrevCash = A#Cash"
```

계정 표현식 문자를 사용하여 EXP 인수의 등호 양쪽에 차원 멤버를 지정할 수 있습니다. [Exp](#)를 참조하십시오. 다음 예에서는 PrevCash 계정 및 사용자정의 3 차원 Golf 멤버의 이전 연도 교차 데이터를 현재 연도의 PrevCash 및 Golf 교차에 삽입합니다.

```
HS.EXP "A#PrevCash.C3#Golf = A#Cash.Y#Prior.C3#Golf"
```

 **팁:**

Y# 계정 표현식 문자 뒤에 Prior 키워드를 사용하면 EXP에서 이전 연도의 데이터를 검색합니다. 계정 표현식의 연도 및 기간에 적용되는 여러 개의 유사 키워드가 있습니다. [기간 및 연도 키워드](#)를 참조하십시오.

## 금액을 계산하여 계정 설정

두 계정에 포함된 금액을 계산한 후 그 결과를 다른 계정에 삽입하는 태스크도 자주 사용됩니다. EXP 함수는 인수의 등호 오른쪽에서 더하기, 빼기, 곱하기 및 나누기를 지원합니다.

다음 예에서는 계산 규칙이 Sales 계정의 값을 UnitsSold 계정의 값으로 나눈 후 몫을 AvgPrice 계정에 삽입합니다.

```
HS.EXP "A#AvgPrice = A#Sales / A#UnitsSold"
```

## 조건부 규칙

특정 차원 멤버가 POV의 현재 멤버인 경우에만 규칙에서 작업을 실행하도록 할 수 있습니다. 예를 들어 [실제]가 현재 시나리오일 때와 [예산]이 현재 시나리오일 때 계정 값이 서로 다른 방식으로 계산되도록 할 수 있습니다.

### 💡 팁:

Oracle Hyperion Financial Management에서 현재 차원 멤버를 판별하는 방법에 대한 자세한 내용은 [현재 차원 멤버](#)를 참조하십시오.

이렇게 하려면 VBScript If 구조에 Member 함수를 사용합니다. Member는 다음 차원의 현재 멤버 이름을 가져옵니다.

- 엔티티. 엔티티 객체를 사용하여 현재 엔티티를 가져오거나 상위 객체를 사용하여 현재 엔티티의 상위를 가져옵니다.
- 기간
- 시나리오
- 값
- 연도

If 구조를 사용하면 특정 조건이 참인 경우에만 명령문을 실행할 수 있습니다. 다음 섹션에서는 If 구조에 Member를 사용하여 차원 멤버를 테스트하는 몇 가지 방법을 보여 줍니다.

### 💡 팁:

이러한 섹션에서는 If 구조의 몇 가지 측면에 대해서만 다룹니다. If 구조에 대한 자세한 내용은 Microsoft VBScript 설명서를 참조하십시오. Microsoft 웹 사이트에서 VBScript 설명서를 다운로드할 수 있습니다.

## 차원 멤버 테스트

특정 차원 멤버가 현재 멤버인 경우에만 Oracle Hyperion Financial Management에서 작업을 실행하게 하려면 멤버 함수의 반환 값을 테스트하는 If 구조를 사용합니다.

다음 예에서는 현재 시나리오가 예산인 경우 Financial Management에서 UnitsSold 및 Price 계정의 금액을 곱한 후 그 결과를 Sales 계정에 삽입합니다.

```
If HS.Scenario.Member = "Budget" Then

    HS.EXP "A#Sales = A#UnitsSold * A#Price"

End If
```



 **팁:**

모든 If 구조는 If...Then 문으로 시작하고 End If 문으로 끝나야 합니다. 조건이 충족될 경우 실행할 작업은 위에 표시된 대로 If...Then 문과 End If 문 사이에 배치됩니다.

## 둘 이상의 멤버 테스트

If...Then 문에서 둘 이상의 멤버를 테스트할 수 있습니다. 즉, 한 차원의 둘 이상 멤버에 대해 작업을 실행할 수 있습니다. 차원 멤버 테스트 섹션의 예를 고려해 보십시오. 현재 시나리오가 예산이나 예측인 경우 Sales 계정의 값이 계산되도록 할 수 있습니다.

둘 이상의 멤버를 테스트하려면 두 개의 Member 함수와 VBScript Or 키워드를 If...Then 문에 사용합니다. 첫번째 Member 함수 뒤에 Or를 넣은 다음 Or와 Then 사이에 두번째 Member 함수를 넣습니다.

 **팁:**

이러한 방법으로 둘 이상의 멤버를 테스트할 수 있습니다. 테스트할 각 멤버에 대해 Member 함수와 Or 키워드의 조합을 추가로 포함합니다.

다음 예에서는 현재 시나리오가 예산이나 예측인 경우 Oracle Hyperion Financial Management에서 UnitsSold 및 Price 계정의 금액을 곱한 후 그 결과를 Sales 계정에 삽입합니다.

```
If HS.Scenario.Member = "Budget" Or HS.Scenario.Member = "Forecast"
Then
```

```
    HS.EXP "A#Sales = A#UnitsSold * A#Price"
```

```
End If
```

## 각 멤버에 대해 서로 다른 작업 수행

규칙에서 차원의 각 멤버에 대해 서로 다른 작업을 수행하도록 할 수 있습니다. 예를 들어 현재 시나리오가 [예산]일 때와 현재 시나리오가 [실제]일 때 서로 다른 계산이 수행되도록 할 수 있습니다.

여러 다른 작업을 조건부로 수행하려면 If 구조에 Elself 문을 하나 이상 포함합니다. 각 Elself 문에서 서로 다른 멤버를 테스트하도록 하고 멤버에 대해 수행할 작업을 Elself 문 아래에 넣습니다.

다음 예에서는 현재 시나리오가 예산인지 또는 실제인지에 따라 서로 다른 계정이 업데이트됩니다.

- 현재 시나리오가 예산인 경우 Oracle Hyperion Financial Management에서 UnitsSold 및 Price 계정의 금액을 곱한 후 그 결과를 Sales 계정에 삽입합니다.

- 현재 시나리오가 실제이면 Financial Management에서 Sales 계정의 값을 UnitsSold 계정의 값으로 나눈 후 몫을 Price 계정에 삽입합니다.

```
If HS.Scenario.Member = "Budget" Then

    HS.EXP "A#Sales = A#UnitsSold * A#Price"

    ElseIf HS.Scenario.Member = "Actual" Then

        HS.EXP "A#Price = A#Sales / A#UnitsSold"

End If
```

### 💡 팁:

If...Then 문과 ElseIf 문에 지정된 조건이 하나도 충족되지 않을 경우 작업이 실행되도록 하려면 VBScript를 사용하여 If 구조에 Else 문을 추가할 수 있습니다. 자세한 내용은 Microsoft VBScript 설명서를 참조하십시오.

## 모든 계정의 기초 잔액 설정

계정의 기초 잔액을 설정하려면 Exp 및 IsFirst 함수를 If 구조에 사용합니다. Exp 및 If 구조에 대해서는 이전 섹션에서 설명했습니다. IsFirst는 현재 기간이 현재 시나리오의 기본 빈도에서 첫 번째 기간인지 여부를 확인합니다. 예를 들어 시나리오의 기본 빈도가 매월이면 IsFirst는 현재 기간이 연도의 첫 번째 달인지 여부를 확인합니다.

기초 잔액을 설정하려면 If 구조의 If...Then 문에 IsFirst를 넣은 다음 이 명령문과 End If 문 사이에 Exp를 넣습니다. Exp의 인수에 특정 계정을 포함할 수도 있지만 모든 계정의 기초 잔액을 설정하는 것이 좋습니다.

다음 예에서는 모든 계정의 기초 잔액을 설정하는 방법을 보여 줍니다. 이 예를 수정하지 않고 계산 규칙에 재입력하거나 복사할 수 있습니다.

```
HS.EXP "A#ALL = A#ALL.P#Prior"
```

### ✎ 주:

이 예에서는 A# 뒤에 키워드 ALL이 오기 때문에 규칙이 모든 계정에 적용됩니다. 이외에도 P# 문자 뒤에 키워드 Prior가 오기 때문에 EXP에서 현재 기간의 이전 기간에서 계정 데이터를 가져옵니다.

## 규칙 파일 생성

Notepad ++ 등의 텍스트 편집기에서 규칙을 생성할 수 있습니다. 규칙 파일은 멀티바이트 문자 세트(MBCS)를 지원하는 ASCII 형식이거나, Little Endian 바이트 순서를 사용하여 유니코드 형식으로 인코딩된 파일일 수 있습니다. Visual Basic 스크립트 함수와 Oracle Hyperion Financial Management 함수를 규칙 파일에 사용합니다. 기본적으로 규칙 파일은 RLE 파일 확장자를 사용합니다.

모든 유형의 Financial Management 규칙을 임의 순서로 규칙 파일에 포함할 수 있습니다. [테이블 1](#)에서는 각 루틴을 정의하는 기본 구문을 보여줍니다.

**표 10-3** 규칙 루틴

규칙 루틴	구문
Sub Calculate	Sub Calculate() '여기에 계산 규칙을 입력합니다. End Sub
Sub Translate	Sub Translate() '여기에 환산 규칙을 입력합니다. End Sub
Sub Allocate	Sub Allocate() '여기에 할당 규칙을 입력합니다. End Sub
Sub Input	Sub Input() '여기에 입력 규칙을 입력합니다. End Sub
Sub NoInput	Sub NoInput() '여기에 NoInput 규칙을 입력합니다. End Sub
Sub Consolidate	Sub Consolidate() '여기에 통합 규칙을 입력합니다. End Sub
Sub Dynamic	Sub Dynamic() '여기에 동적 규칙을 입력합니다. End Sub
Sub Transactions	Sub Transactions() '여기에 트랜잭션 규칙을 입력합니다. End Sub
Sub EquityPickup	Sub EquityPickUp() '여기에 지분 매입 규칙을 입력합니다. End Sub

표 10-3 (계속) 규칙 루틴

규칙 루틴	구문
Sub OnDemand	Sub OnDemand_<ruleName> '여기에 주문형 규칙을 입력합니다.' End Sub

### 지분 매입 규칙 예

다음 절에서는 샘플 지분 매입 규칙 섹션을 보여 줍니다. 지분 매입을 계산하려면 애플리케이션 관리자가 규칙 파일에 Sub EquityPickUp이라는 새 섹션을 생성해야 합니다. 여기서 EPU 계산이 정의됩니다. 섹션이 실행될 때의 기본 POV는 다음과 같습니다.

- 현재 시나리오, 연도 및 기간
- 엔티티: 처리된 쌍의 소유자
- 값: 엔티티 통화

```
Sub EquityPickUp()

Owner = Hs.Entity.Member

Owned = Hs.Entity.Owned

OwnerDefaultCurrency = HS.Entity.DefCurrency("")

lPown = Hs.GetCell("E#" & Owned & ".I#" & Owner &
".V#[None].A#[Shares%Owned].C1#[None].C2#[None].C3#[None].C4#[None]")

Hs.Clear "A#IncomeFromSubsidiary.I#" & Owned

Hs.Exp "A#IncomeFromSubsidiary.I#" & Owned & " = E#" & Owned & ".V#" &
OwnerDefaultCurrency & ".A#NetIncome.I#[ICP Top] *" & lPown

End Sub
```

### 주문형 규칙 예

데이터 형식에서 주문형 규칙을 실행하려면 관리자가 규칙 파일에 Sub OnDemand라는 새 섹션을 생성해야 합니다. 여기서 주문형 규칙이 정의됩니다.

```
Sub OnDemand_Calculation
```

```
HS.Exp "A#CogsTP=15424"
```

```
HS.Exp "A#Admex=32452"
```

```
End Sub
```

Sub Calculate에서 사용할 수 있는 모든 HS 함수(다른 함수 제외)는 주문형 규칙에도 사용할 수 있습니다. 또한 Sub Calculate와 달리 이전에 IsCalculated 데이터 포인트에 기록된 데이터는 주문형 규칙을 실행할 때 지워지지 않습니다.

## 규칙 로드

규칙 변경은 데이터에 영향을 줄 수 있으며 메타데이터를 사용합니다. 따라서 규칙 로드 프로세스에서는 Oracle Hyperion Financial Management 서버 클러스터에 글로벌 잠금을 적용합니다. 이전에 시작한 다음 유형의 작업이 완료되어야 규칙 로드를 계속할 수 있습니다.

- 통합
- 데이터 입력
- 데이터, 분개 또는 보안 로드
- 데이터를 데이터베이스로 추출
- 멤버 목록 로드
- 메타데이터 로드

규칙 스캔 프로세스에서는 규칙 로드와 동일한 제한에 따라 시스템이 차원 멤버와 기타 매개변수를 검증할 수 있도록 합니다. 규칙 스캔 및 로드 프로세스는 대기되고 차단 태스크가 완료된 후 자동으로 시작됩니다. 예를 들어 장기 실행 통합 등의 작업을 수행하는 동안이 아니라 서버 클러스터에서 가벼운 작업이 수행되는 기간 동안 규칙을 로드하는 것이 좋습니다. [실행 중인 태스크] 페이지를 검사하여 진행 중인 프로세스(예: 통합 또는 데이터 로드)를 확인할 수 있습니다.

규칙 파일을 애플리케이션에 로드하면 해당 애플리케이션의 사용자는 시스템이 변경되었으며 애플리케이션에서 로그오프한 후 다시 로그온해야 한다는 통지를 받습니다.

규칙 파일에 내부거래 트랜잭션이 포함된 경우 애플리케이션의 게시된 트랜잭션을 규칙 파일의 새 트랜잭션과 비교하여 검증할 수 있습니다. 규칙 파일의 Sub Transactions 섹션에서는 내부거래 트랜잭션 세부정보를 지원하는 계정을 정의합니다.

 주:

웹 팝업 차단이 예외에 Financial Management를 추가하는 것이 좋습니다. 데이터 로드와 같은 일부 태스크를 수행하면 태스크 상태를 표시하는 상태 창이 팝업됩니다. 컴퓨터에 팝업 차단이 설정되어 있으면 상태 창이 표시되지 않습니다.

규칙을 로드하려면 다음을 수행합니다.

1. **통합, 로드, 애플리케이션 요소** 순으로 선택합니다.
2. **규칙** 섹션에서 로드할 파일 이름을 입력하거나 **찾아보기**를 눌러 파일을 찾습니다.

 주:

기본적으로 규칙 파일은 RLE 파일 확장자를 사용합니다. 로드 프로세스에서도 TXT 파일 확장자가 허용되지만 RLE 파일 확장자를 사용하는 것이 좋습니다.

3. **선택 사항: 무결성 확인**을 선택하여 게시된 내부거래 트랜잭션이 로드되는 규칙 파일의 Sub Transactions 섹션에 있는 명령문에서 적합한지 확인합니다.
4. **선택 사항: 스캔**을 눌러 파일 형식이 정확한지 확인합니다.
5. **로드**를 누릅니다.

 팁:

옵션을 기본값으로 재설정하려면 **재설정**을 누릅니다.

6. **선택사항: 로그** 파일을 다운로드하려면 **로그 다운로드**를 누릅니다. **열기**를 눌러 로그 파일을 표시하거나 **저장**을 누르고 로컬에서 파일을 저장할 위치를 선택합니다.

## 규칙 추출

규칙을 추출하면 멀티바이트 문자 세트(MBCS)를 지원하는 ASCII 파일에 저장됩니다. 기본적으로 규칙 파일은 RLE 파일 확장자를 사용합니다. 규칙을 추출한 후 텍스트 편집기에서 보고 수정할 수 있습니다.

규칙을 추출하려면 다음을 수행합니다.

1. **통합, 추출, 애플리케이션 요소** 순으로 선택합니다.
2. **규칙** 섹션에서 **추출**을 누릅니다.
3. 브라우저에 표시되는 다운로드 지침에 따라 추출된 파일을 다운로드합니다.  
지침은 사용 중인 웹 브라우저에 따라 달라집니다. 설정한 웹 디렉토리에 파일을 저장해야 합니다.
4. **선택사항: 로그** 파일을 다운로드하려면 **로그 다운로드**를 누릅니다. **열기**를 눌러 로그 파일을 표시하거나 **저장**을 누르고 로컬에서 파일을 저장할 위치를 선택합니다.

## 규칙 프로파일링

규칙 프로파일링을 사용하여 주요 서브루틴 레벨의 규칙 실행에 대한 자세한 정보를 캡처할 수 있습니다. 규칙 프로파일에는 통합 태스크 중 각 엔티티를 계산, 환산 및 통합하는 데 걸린 시간에 대한 통계가 포함됩니다. 규칙 프로파일을 보고 규칙이 효율적으로 실행되는지 확인하고 통합 중 규칙 실행 또는 엔티티 처리에서 발생하는 이슈를 해결할 수 있습니다.

규칙 프로파일을 실행하고 보려면 애플리케이션 관리자 보안 액세스 권한이 있어야 합니다.

## 규칙 파일에 프로파일 정보 추가

특정 프로파일링 데이터를 캡처하기 위해 규칙 파일 내에 블록을 정의할 수 있습니다.

규칙 파일에 프로파일 정보를 추가하려면 다음을 수행합니다.

1. 규칙 파일을 열고 다음과 같이 블록 레벨 정보가 있는 주요 서브루틴을 업데이트합니다.
  - HS.StartTimer를 사용하여 블록의 시작을 정의합니다.
  - HS.EndTimer를 사용하여 블록의 끝을 정의합니다.
  - 규칙 파일에 고유한 타이머 ID를 정의합니다. 양의 정수를 사용하십시오.
  - 사용자 작업(예: 계산)을 정의합니다. 블록 이름은 \_TimeID와 함께 추가됩니다.

예:

```
HS.StartTimer 1, "Calculate", "", "", "" '1: Unique Timer ID,
Calculate: User Defined Block Name
```

```
          If Hs.Scenario.Member ="BudV1" or
Hs.Scenario.Member="BudV2" Then
```

```
          If HS.Entity.IsBase "Geographical", Then
```

...

```
          HS.EndTimer 1 '1: Corresponding Unique Timer ID of
earlier User defined block to represent End of the Block
```

2. 업데이트된 규칙 파일을 로드합니다.
3. 프로파일링할 작업을 실행합니다.

## 규칙 프로파일링 시작 및 중지

기본적으로 규칙 프로파일링은 사용으로 설정되어 있지 않습니다. 애플리케이션에 대해 사용으로 설정해야 합니다.

개별 서버가 아니라 모든 서버에서 애플리케이션별 프로파일링을 사용할 수 있습니다.

규칙 프로파일링을 시작하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리, 애플리케이션** 순으로 선택하거나 **관리 태스크**에서 **애플리케이션**을 선택합니다.
3. 애플리케이션을 선택하고 마우스 오른쪽 버튼을 눌러 **프로파일 요약**을 선택합니다.

애플리케이션에서 프로파일링이 실행되지 않은 경우 프로파일 요약 페이지에 결과가 표시되지 않습니다. 프로파일링이 이전에 실행된 경우 이전 실행의 결과를 볼 수 있습니다.

4. 화면 맨위에서 **시작**을 선택합니다.
5. 다음 확인 메시지가 표시됩니다. "활성 사용자가 있는 프로덕션 시스템에서는 프로파일링을 사용으로 설정하지 않는 것이 좋습니다. 프로파일링을 사용으로 설정하면 시스템 성능이 저하될 수 있습니다. 프로파일링을 사용으로 설정하시겠습니까?" 확인 프롬프트가 나타나면 **예**를 누릅니다.



**주:** 이미 시작된 프로파일링을 중지하려면 **중지**를 누릅니다.

## 규칙 프로파일 결과 보기

사용자가 규칙을 호출하는 작업을 수행하면 프로파일에서 규칙 실행에 대한 정보를 캡처합니다. 이러한 작업에는 데이터 그리드, 데이터 양식 및 지분 매입 모듈에서 사용할 수 있는 프로세스가 포함되어 있습니다. 작업으로는 계산, 통합, 할당, 환산, 입력, 입력 없음, 동적, 트랜잭션, 지분 매입 등이 있습니다.

프로파일 요약의 규칙 뷰 또는 엔티티 뷰에서 결과를 볼 수 있습니다. 규칙 뷰에는 실행된 주요 규칙이 표시됩니다.

엔티티 뷰에는 엔티티 구조를 기반으로 엔티티가 표시됩니다. 통합 규칙이 실행된 경우에만 사용할 수 있으며 통합, 계산(1차 하위 노드 포함) 및 환산 프로세스의 결과가 표시됩니다.

규칙 프로파일 결과를 보려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리, 애플리케이션** 순으로 선택합니다.
3. 애플리케이션을 선택하고 마우스 오른쪽 버튼을 눌러 **프로파일 요약**을 선택합니다.

프로파일 요약 페이지에는 태스크, 사용자, 규칙이 실행된 날짜와 시간 등 규칙 프로파일이 나열되어 있습니다. 프로파일이 여러 개 있는 경우 드롭다운에서 보려는 프로파일을 선택합니다.

4. **규칙 뷰** 또는 **엔티티 뷰**를 선택합니다.
5. 결과 페이지를 새로고치려면 **새로고침**을 누릅니다.



## 규칙 뷰

### 사용자 작업 기준

- 이 사용자 작업에 걸린 총 시간
- 규칙 처리에 걸린 시간(총 시스템 시간에 대한 퍼센트로 표시)

### 주 서브루틴 기준

- 서브루틴/블록 - 실행된 서브루틴입니다.
- 총 호출 - 규칙에서 사용자 정의 주 서브루틴/블록이 해당 사용자 작업에 대해 실행된 횟수입니다.
- 평균 시간 - 각 서브루틴 실행에 걸린 평균 시간으로, 모든 개별적으로 경과된 시간의 합계를 총 서브루틴/블록 호출 수로 나눈 값입니다.
- 총 시간 - 모든 개별적으로 경과된 서브루틴 시간의 합계입니다. 예를 들어 주 서브루틴 계산이 병렬로 10개 엔티티에 대해 실행되었고 각각 1초의 시간이 걸린 경우 총 시간은 10초입니다.
- POV 세부정보 - 서브루틴 실행에 걸린 시간 기준 최고 엔티티(최고 시간부터 최저 시간 순으로 정렬됨)

Sub/Block	Total Invocations	Average Time	Total Time	Entity	Period	Value	Elapsed Time
Calculate	33	166	5506	UKSales	February	GSP Total	494.0
Consolidate	8	934	7474	UKSales	February	[Proportion]	454.0
CustomLogic	1	1	1				
Translate	1	250	250				

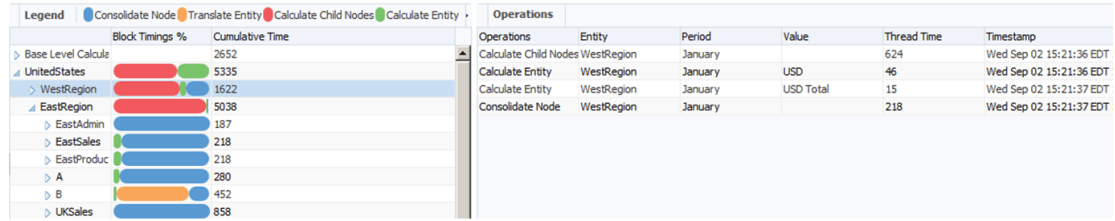
## 엔티티 뷰

엔티티 뷰에서는 사용자 시작 통합 태스크에 대한 정보(예: 모두 통합, 데이터가 있는 항목 모두 통합, 변경된 항목 통합 및 기여금액 계산)를 제공합니다. 엔티티 막대 그래프에서 행을 마우스로 가리켜 블록 타이밍 퍼센트를 확인할 수 있습니다. 엔티티 행을 축소하고 확장할 수 있습니다.

### 개별 사용자 통합 태스크

통합 태스크에 걸린 총 시간

### 엔티티당



엔티티 계층에서 각 분기를 통합하는 데 걸린 시간입니다. 계층의 엔티티를 기준 레벨 엔티티로 확장할 수 있습니다. 여러 상위이 있는 엔티티는 중복 항목으로 표시됩니다.

- 누적 시간 - 왼쪽 패널의 시간은 엔티티 계층에서 분기를 통합하는 데 걸린 총 시간을 나타냅니다. 각 엔티티에 대해 표시되는 값은 엔티티 및 해당 하위를 처리하는 데 걸린 실제 시간입니다. 통합의 다중 스레드 특성으로 인해 1차 하위 엔티티의 총 개별 시간 합계가 해당 상위 엔티티의 총 시간보다 많을 수 있습니다.
- 작업 - 오른쪽 패널의 각 행은 왼쪽 패널에 강조 표시된 엔티티 처리 중 하위 태스크를 수행하는 별도의 스레드를 표시합니다.
- 스레드 시간 - 각 스레드에서 각 하위 태스크를 완료하는 데 걸리는 시간

## 규칙 프로파일 삭제

특정 작업(예: 통합 또는 환산)에 대한 규칙 프로파일 결과를 삭제하거나 해당 애플리케이션의 모든 작업에 대한 모든 결과를 삭제할 수 있습니다.

규칙 프로파일 결과를 삭제하려면 다음을 수행합니다.

1. 탐색, 관리, 통합 관리 순으로 선택합니다.
2. 관리, 애플리케이션 순으로 선택합니다.
3. 애플리케이션을 선택하고 마우스 오른쪽 버튼을 눌러 **프로파일 요약**를 선택합니다.
4. 다음 태스크 중 하나를 수행합니다.
  - 특정 작업에 대한 결과를 삭제하려면 드롭다운 목록에서 작업을 선택하고 해당 작업 옆에 있는 **삭제** 아이콘을 누릅니다.
  - 모든 결과를 삭제하려면 맨위 메뉴에서 **모두 삭제**를 누릅니다.
5. 확인 프롬프트가 나타나면 **예**를 누릅니다.

# 11

## 규칙 함수

이 함수를 사용하여 다음과 같은 규칙 유형을 작성할 수 있습니다.

- 할당 규칙
- 계산 규칙
- 통합 규칙
- 동적 계산 규칙
- 지분 매입 규칙
- 입력 규칙
- NoInput 규칙
- 주문형 규칙
- 트랜잭션 규칙
- 환산 규칙

특정 루틴에서만 사용되는 함수도 있고, 여러 루틴 내에서 여러 유형의 규칙과 함께 사용할 수 있는 함수도 있습니다.

모든 함수에 대한 개요는 [함수 개요](#)를 참조하십시오.

## 함수 개요

**테이블 1**에는 Oracle Hyperion Financial Management 함수, 함수와 함께 사용할 수 있는 객체 및 함수를 사용할 수 있는 규칙 유형이 요약되어 있습니다. 함수는 영문자별로 표시됩니다. 각 함수에 대한 세부 섹션은 표 뒤에 나와 있습니다.

### 주:

11.1.2.2 이전 버전의 Financial Management 릴리스에서 마이그레이션된 레거시 애플리케이션은 "사용자정의1...4" 객체를 계속 사용할 수 있습니다. 11.1.2.2에서 생성된 모든 애플리케이션은 새 Custom(Alias) 객체 구문을 사용하여 사용자정의 별칭/약식 레이블/긴 레이블을 통해 참조되는 사용자정의 차원을 지정해야 합니다.

이 표에서 규칙 유형은 다음과 같이 약어로 표시됩니다.

- Alloc - 할당
- Calc - 계산
- Con - 통합
- Dyn - 동적 계산
- EPU - 지분 매입

- Tran - 환산
- Trans - 트랜잭션

**표 11-1 Financial Management 함수**

함수	설명	객체	규칙 유형
ABSExp	계산 표현식을 실행하고 결과를 절대값으로 저장합니다.	HS	Calc, Tran, Alloc
AccountType	멤버의 계정 유형을 가져옵니다.	계정	Calc, Tran, Con, Alloc
AccountTypeID	멤버의 계정 유형 ID를 가져옵니다.	계정	Calc, Tran, Con, Alloc
AddEntityToList	멤버 목록에 지정된 엔티티와 상위를 추가합니다.	HS	멤버 목록
AddEntityToListUsingIDs	엔티티 및 상위 ID를 사용하여 내부 목록에 지정된 엔티티와 상위를 추가합니다.	HS	멤버 목록
AddMemberToList	멤버 목록에 멤버를 추가합니다.	HS	멤버 목록
AddMemberToListUsingMemberID	멤버 ID를 사용하여 내부 목록에 지정된 멤버를 추가합니다.	HS	멤버 목록
Alloc	셀에 데이터를 할당합니다.	HS	Alloc
AllowAdjFromChildren	멤버에 대해 1차 하위 구성요소의 분개 계시가 허용되는지 여부를 확인합니다.	엔티티, 상위	Calc, Tran, Con, Alloc
AllowAdjs	멤버에 대해 분개 계시가 허용되는지 여부를 확인합니다.	엔티티, 상위	Calc, Tran, Con, Alloc
ApplicationName	규칙이 실행되는 애플리케이션의 이름을 반환합니다.	AppSettings	Calc, Tran, Con, Alloc
CalculateExchangeRate	하나의 통화에서 다른 통화로의 환율을 계산합니다.	HS	Calc, Tran, Con, Alloc, Dyn, Trans
CalculateRate	지정된 엔티티의 현재 환율을 가져옵니다.	HS	Calc, Tran, Con, Alloc, Dyn, Trans
CalcStatus	셀의 계산 상태를 가져옵니다.	HS	Calc, Tran, Con, Alloc
Clear	셀에서 데이터를 제거합니다.	HS	Calc, Tran, Con, Alloc

표 11-1 (계속) Financial Management 함수

함수	설명	객체	규칙 유형
Con	[Proportion] 및 [Elimination] 계정에 데이터를 넣습니다.	HS	Con
Consol1, Consol2, Consol3	노드의 Consol1...3 시스템 계정 값을 가져옵니다.	노드	Calc, Tran, Con, Alloc
ConsolidateYTD	시나리오가 YTD 또는 주기적 메소드를 사용하여 통합되는지 여부를 확인합니다.	시나리오	Calc, Tran, Con, Alloc
ContainsCellText	셀에 셀 텍스트가 포함되어 있는지 여부를 확인합니다.	HS	Calc, Tran, Con, Alloc
ContainsCellTextWithLabel	지정된 레이블의 셀 텍스트가 셀에 포함되어 있는지 여부를 확인합니다.	HS	Calc, Tran, Con, Alloc
통화	애플리케이션이나 값 멤버의 통화 유형을 가져옵니다.	AppSettings, 값	Calc, Tran, Con, Alloc
CustomTop	계정의 CustomTopMember를 가져옵니다.	계정	Calc, Tran, Con, Alloc
DataUnitItem	통합, 계산 또는 환산하는 동안 처리할 데이터 단위 항목을 가져옵니다.	HS	Calc, Tran, Con
소수 자릿수	지정된 계정의 소수 자릿수를 가져옵니다.	계정	Calc, Tran, Con, Alloc
DefaultFreq	기본 시나리오 빈도를 가져옵니다.	시나리오	Calc, Tran, Con, Alloc
DefaultParent	멤버의 기본 상위 멤버를 가져옵니다.	계정	Calc, Tran, Con, Alloc
DefaultTranslate	규칙을 건너뛰어 환산을 계산합니다.	HS	Tran
DefaultView	기본 시나리오 뷰를 가져옵니다.	시나리오	Calc, Tran, Con, Alloc
DefCurrency	엔티티나 상위의 기본 통화를 가져옵니다.	엔티티, 상위	Calc, Tran, Con, Alloc
DOWn	노드의 직접 소유 지분 (DOWn)을 가져옵니다.	노드	Calc, Tran, Con, Alloc
Dynamic	계산이 필요한 동적 계정에 대한 공식을 지정합니다.	HS	Dyn

표 11-1 (계속) Financial Management 함수

함수	설명	객체	규칙 유형
Exp	계산 표현식을 실행하고 지정된 POV에 데이터를 넣습니다.	HS	Calc, Tran, Alloc
GetCell	셀에 포함된 데이터를 가져옵니다.	HS	Calc, Tran, Con, Alloc
GetCellNoData	셀에 포함된 데이터를 가져오고 셀에 데이터가 포함되어 있는지 여부를 나타냅니다.	HS	Calc, Tran, Con, Alloc
GetCellRealData	셀에 포함된 데이터를 가져오고 셀에 실제 데이터가 포함되어 있는지 여부를 나타냅니다.	HS	Calc, Tran, Con, Alloc
GetCellText	지정된 POV의 기본 셀 텍스트를 가져옵니다.	HS	Calc, Tran, Con, Alloc
GetCellTextWithLabel	POV에서 지정된 셀 텍스트 레이블의 셀 텍스트를 가져옵니다.	HS	Calc, Tran, Con, Alloc
GetCellType	셀 유형을 가져옵니다.	HS	Calc, Tran, Con, Alloc
GetCustomLabelArray	사용자정의 차원 이름과 별칭 목록을 반환합니다.	HS	Calc, Tran, Con, Alloc
GetItem	통합을 위해 처리할 개별 레코드를 가져옵니다.	DataUnit	Calc, Tran, Con
GetItemIDs2	차원 ID 번호를 사용하여 통합을 위해 처리할 개별 레코드를 가져옵니다.	DataUnit	Calc, Tran, Con
GetNumItems	통합을 위해 처리할 레코드 수를 가져옵니다.	DataUnit	Calc, Tran, Con
GetNumLID	지정된 POV의 라인 항목 세부정보 수를 가져옵니다.	HS	Calc, Tran, Con, Alloc
GetRate	셀의 환율을 가져옵니다.	HS	Calc, Tran, Con, Alloc
GetSubmissionGroup	셀의 제출 그룹을 가져옵니다.	HS	Calc, Tran, Con, Alloc
GetSubmissionPhase	셀의 제출 단계를 가져옵니다.	HS	Calc, Tran, Con, Alloc
Holding	멤버의 지주 회사를 가져옵니다.	엔티티, 상위	Calc, Tran, Con, Alloc

표 11-1 (계속) Financial Management 함수

함수	설명	객체	규칙 유형
ICPTopMember	현재 계정 차원 멤버나 지정된 계정 멤버의 ICPTopMember를 가져옵니다.	계정	Calc, Tran, Con, Alloc
ICPWeight	[ICP Top] 값 멤버에 합산되는 ICP 엔티티 잔액의 백분율을 가져옵니다.	AppSettings	Calc, Tran, Con, Alloc
IDFromMember	지정된 멤버의 ID 번호를 가져옵니다.	계정, 사용자정의1 - 사용자정의4, 엔티티, ICP, 상위, 연도, 기간, 시나리오, 값, 뷰	Calc, Tran, Con, Alloc
ImpactStatus	지정된 데이터 단위의 상태를 영향 받음으로 변경합니다.	HS	Calc
Input	지정된 계정에 대해 상위 엔티티 레벨의 입력을 활성화합니다.	HS	Input
IsAlmostEqual	두 값이 같은지 여부를 확인합니다.	HS	Calc, Tran, Con, Alloc, Dyn, Trans
IsBase	멤버가 기준 멤버인지 여부를 확인합니다.	계정, 사용자정의1 - 사용자정의4, 엔티티, 상위, 노드	Calc, Tran, Con, Alloc
IsCalculated	계정이 계산되는지 여부를 확인합니다.	계정	Calc, Tran, Con, Alloc
IsChild	멤버가 다른 멤버의 1차 하위 구성요소인지 여부를 확인합니다.	계정, 사용자정의1 - 사용자정의4, 엔티티, 상위, 노드	Calc, Tran, Con, Alloc
IsConsolidated	계정이 통합되는지 여부를 확인합니다.	계정	Calc, Tran, Con, Alloc
IsDescendant	멤버가 다른 멤버의 하위 멤버인지 여부를 확인합니다.	계정, 사용자정의1 - 사용자정의4, 엔티티, 상위, 노드	Calc, Tran, Con, Alloc
IsFirst	애플리케이션의 첫 번째 기간이나 연도인지 여부를 확인합니다.	기간, 연도	Calc, Tran, Con, Alloc
IsICP	멤버가 ICP인지 여부를 확인합니다.	계정, 엔티티, 상위	Calc, Tran, Con, Alloc
IsLast	애플리케이션의 마지막 기간이나 연도인지 여부를 확인합니다.	기간, 연도	Calc, Tran, Con, Alloc
IsTransCur	값 멤버가 환산된 통화 멤버인지 여부를 확인합니다.	값	Calc

표 11-1 (계속) Financial Management 함수

함수	설명	객체	규칙 유형
IsTransCurAdj	값 멤버가 환산된 통화 조정 멤버인지 여부를 확인합니다.	값	Calc
IsValidDest	지정된 POV가 적합한 대상인지 여부를 확인합니다.	HS	Calc, Tran, Con, Alloc
IsZero	전달된 값이 0인지 여부를 확인합니다.	HS	Calc, Tran, Con, Alloc, Dyn, Trans
List	목록에 포함된 요소를 가져옵니다.	계정, 사용자정의1 - 사용자정의4, 엔티티, 상위, ICP, 노드, 시나리오	Calc, Tran, Con, Alloc
Member	멤버 이름을 가져옵니다.	엔티티, 상위, 기간, 시나리오, 값, 연도, 뷰	Calc, Tran, Con, Alloc
MemberFromID	지정된 ID 번호의 멤버를 가져옵니다.	계정, 사용자정의1 - 사용자정의4, 엔티티, ICP, 상위, 연도, 기간, 시나리오, 값, 뷰	Calc, Tran, Con, Alloc
Method	멤버의 통합 메소드를 가져옵니다.	노드	Calc, Tran, Con, Alloc
NoInput	셀에 입력할 수 없도록 합니다.	HS	Noinput
NoRound	반올림을 해제합니다.	HS	Calc, Tran, Con, Alloc
NumBase	기준 멤버 수를 가져옵니다.	계정, 사용자정의1 - 사용자정의4, 엔티티, 상위, 노드, 기간	Calc, Tran, Con, Alloc
Number	현재 기간 번호를 가져옵니다.	기간	Calc, Tran, Con, Alloc
NumChild	멤버의 1차 하위 구성요소 수를 가져옵니다.	계정, 사용자정의1 - 사용자정의4, 엔티티, 상위, 노드	Calc, Tran, Con, Alloc
NumCustom	애플리케이션에 대해 정의된 사용자정의 차원 수를 가져옵니다.	HS	Calc, Tran, Con, Alloc
NumDescendant	멤버의 하위 멤버 수를 가져옵니다.	계정, 사용자정의1 - 사용자정의4, 엔티티, 상위, 노드	Calc, Tran, Con, Alloc
NumPerInGen	처리되는 현재 기간에 대한 세대의 기간 수를 가져옵니다.	기간	Dynamic
NumPeriods	지정된 시나리오의 빈도에 대해 정의된 기간 수를 가져옵니다.	시나리오	Dynamic
OpenCellTextUnit	여러 셀에 대한 셀 텍스트를 반환합니다.	HS	Calc, Tran, Con



표 11-1 (계속) Financial Management 함수

함수	설명	객체	규칙 유형
OpenDataUnit	통합에 사용할 데이터 단위를 가져옵니다.	HS	Calc, Tran, Con
OpenDataUnitSorted	계산, 환산 또는 통합에 사용할 데이터 단위를 정렬된 순서로 가져옵니다.	HS	Calc, Tran, Con
Owned	현재 처리되는 쌍의 소유 엔티티를 가져옵니다.	엔티티	지분 매입
Owner	현재 처리되는 쌍의 소유자 엔티티를 가져옵니다.	엔티티	지분 매입
PCon	연결 지분을 가져옵니다.	노드	Calc, Tran, Con, Alloc
PEPU	EPU 테이블에서 소유 지분을 가져옵니다.	HS	지분 매입
PeriodNumber	검색되는 데이터에 대한 뷰의 기간 수를 가져옵니다.	뷰	Dynamic
PlugAcct	플러그 계정을 가져옵니다.	계정	Calc, Tran, Con, Alloc
POwn	소유 지분을 가져옵니다.	노드	Calc, Tran, Con, Alloc
PVAForBalance	BALANCE 계정의 기본 환산 방법을 결정합니다.	AppSettings	Calc, Tran, Con, Alloc
PVAForFlow	FLOW 계정의 기본 환산 방법을 결정합니다.	AppSettings	Calc, Tran, Con, Alloc
RateForBalance	잔액의 기본 환율을 가져옵니다.	AppSettings	Calc, Tran, Con, Alloc
RateForFlow	플로우의 기본 환율을 가져옵니다.	AppSettings	Calc, Tran, Con, Alloc
ReviewStatus	셀의 프로세스 관리 검토 상태를 가져옵니다.	HS	Calc, Tran, Con, Alloc
ReviewStatusUsingPhase	셀의 단계 ID별 프로세스 관리 검토 상태를 가져옵니다.	HS	Calc, Tran, Con, Alloc
Round	데이터를 반올림합니다.	HS	Calc, Tran, Con, Alloc
Scale	지정된 통화의 스케일을 가져옵니다.	통화	Calc, Tran, Con, Alloc
SecurityAsPartner	ICP 엔티티의 보안 클래스를 가져옵니다.	엔티티, 상위	Calc, Tran, Con, Alloc

표 11-1 (계속) Financial Management 함수

함수	설명	객체	규칙 유형
SecurityClass	차원 멤버의 보안 클래스를 가져옵니다.	계정, 시나리오, 엔티티, 상위, 사용자정의1 - 사용자정의4	Calc, Tran, Con, Alloc
SetCellTextWithLabel	지정된 POV 및 셀 텍스트 레이블의 셀 텍스트에 텍스트 문자열을 씁니다.	HS	Calc
SetData	개별 레코드를 설정합니다.	HS	Calc, Tran
SetDataWithPOV	노드나 통화 큐브에 데이터를 삽입합니다.	HS	Calc, Tran
SubmissionGroup	차원 멤버의 프로세스 관리 제출 그룹을 가져옵니다.	HS	Calc, Tran, Con, Alloc
SupportsProcessManagement	시나리오에서 프로세스 관리를 지원하는지 여부를 확인합니다.	시나리오	Calc, Tran, Con, Alloc
SupportsTran	내부거래 트랜잭션 세부정보 지원이 필요한 애플리케이션 계정을 정의합니다.	시나리오, 연도, 엔티티, 계정, C1...4	Trans
SwitchSign	멤버에 대해 대변이 차변으로 전환되는지 여부를 확인합니다.	사용자정의1 - 사용자정의4	Calc, Tran, Con, Alloc
SwitchType	멤버에 대해 계정 유형이 전환되는지 여부를 확인합니다.	사용자정의1 - 사용자정의4	Calc, Tran, Con, Alloc
Trans	YTD 방법을 사용하여 환산합니다.	HS	Tran
TransPeriodic	주기적 방법을 사용하여 환산합니다.	HS	Tran
UD1...3	멤버의 사용자 정의 속성을 가져옵니다.	계정, 엔티티, 상위, 시나리오, 사용자정의1 - 사용자정의4	Calc, Tran, Con, Alloc
ValidationAccount	검증 계정을 가져옵니다.	AppSettings	Calc, Tran, Con, Alloc
ValidationAccountEx	프로세스 관리 제출 단계의 검증 계정을 가져옵니다.	AppSettings	Calc, Tran, Con, Alloc
XBRLTags	계정의 XBRL 태그를 가져옵니다.	계정	Calc, Tran, Con, Alloc

## ABSEXP

계산 표현식을 실행하고 결과를 절대값으로 저장합니다. 이 함수는 결과 값을 절대값으로 저장한다는 점을 제외하고는 Exp 함수와 같으며 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 할당

### 구문

```
HS.ABSEXP "DestPOV = Expression"
```

**표 11-2 ABSEXP 함수 구문**

매개변수	설명
<i>DestPOV</i>	<p>데이터를 넣을 위치를 식별하는 대상 POV입니다. 계정 멤버를 지정해야 하며, 선택적으로 ICP 및 사용자정의 멤버를 지정할 수 있습니다. 다음과 같은 사용 규칙을 확인합니다.</p> <ul style="list-style-type: none"> <li>• ICP 멤버를 지정하지 않는 경우 기본값은 [ICP None]입니다.</li> <li>• 데이터베이스에 불필요한 값을 채우지 않도록 하려면 데이터 위치 정의와 관련해서 규칙이 가능한 한 명시적이어야 합니다. 계정 차원에 대해 명확하게 정의된 사용자정의 차원 교차를 포함하는 것이 좋습니다. 명확하게 정의된 이러한 교차점에서는 Oracle Hyperion Financial Management 검증 확인을 통해 부적합한 교차점에 데이터를 쓰지 않도록 방지합니다. <a href="#">차원 교차점 고려 사항</a>을 참조하십시오.</li> </ul>
<i>Expression</i>	계산 표현식입니다.

### 반환 값

없습니다.

### 예

다음 예에서는 StateTax 계정의 금액을 설정합니다. 절대 금액은 2014년 Sales 계정의 금액에 2014년 StateRate 계정의 금액을 곱하여 계산됩니다.

```
HS.ABSEXP "A#StateTax = A#Sales.Y#2014 * A#StateRate.Y#2014"
```

## AccountType

현재 계정 멤버나 지정된 계정의 계정 유형을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산

- 환산
- 통합
- 할당

## 구문

```
HS.Account.AccountType ("Account")
```

```
HS.Account.AccountType ("")
```

여기서, *Account*는 적합한 계정 멤버의 이름입니다.

빈 문자열("")을 사용하면 Sub Consolidate 서브루틴에 함수를 사용하는 경우에만 현재 멤버에 이 함수를 적용할 수 있습니다. 그렇지 않으면 이 함수를 사용할 때 계정을 지정하십시오.

## 반환 값

지정된 계정의 계정 유형입니다.



### 주:

계정 유형은 모두 대문자를 사용합니다.

적합한 계정 유형은 다음과 같습니다.

- ASSET
- LIABILITY
- REVENUE
- EXPENSE
- DYNAMIC
- FLOW
- BALANCE
- BALANCERECURRING
- CURRENCYRATE
- GROUPLABEL

**예**

다음 예에서는 Sales 계정의 계정 유형이 REVENUE인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Account.AccountType("Sales") = "REVENUE" Then
```

```
...
```

```
End If
```

## AccountTypeID

현재 계정 멤버나 지정된 계정의 계정 유형 ID를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당
- 동적 계산
- 트랜잭션

**구문**

```
HS.Account.AccountTypeID("Account")
```

```
HS.Account.AccountTypeID("")
```

여기서 *Account*는 적합한 계정 멤버의 이름입니다

빈 문자열("")을 사용하면 Sub Consolidate 서브루틴에 함수를 사용하는 경우에만 현재 멤버에 이 함수를 적용할 수 있습니다. 그렇지 않으면 이 함수를 사용할 때 계정을 지정해야 합니다.

**반환 값**

지정된 계정의 ID입니다. [테이블 1](#)에는 해당 ID를 사용하는 적합한 계정 유형이 나열되어 있습니다.

**표 11-3** 계정 유형 ID

계정 유형	ID
REVENUE(이전 릴리스의 INCOME)	0
EXPENSE	1
ASSET	2
LIABILITY	3

표 11-3 (계속) 계정 유형 ID

계정 유형	ID
BALANCE	4
FLOW	5
CURRENCYRATE	7
GROUPLABEL	10
BALANCERECURRING	11
DYNAMIC	12

## 예

```
If HS.Account.AccountTypeID("Investments") = 2 Then
```

```
...
```

```
End If
```

## AddEntityToList

멤버 목록에 지정된 멤버를 추가합니다. 이 함수는 멤버 목록 파일에서만 사용할 수 있습니다.

## 구문

```
HS.AddEntityToList("Parent", "Child")
```

표 11-4 AddEntityToList 함수 구문

매개변수	설명
<i>Parent</i>	적합한 상위 엔티티의 이름입니다.
<i>Child</i>	상위 엔티티의 1차 하위 구성요소 이름입니다.

## 반환 값

없음

## 예

```
HS.AddEntityToList "UnitedStates", "Maine"
```

## AddEntityToListUsingIDs

엔티티와 상위 멤버의 ID를 사용하면 멤버 목록에 지정된 멤버가 추가됩니다. 이 함수는 멤버 목록 파일에서만 사용할 수 있습니다.

### 구문

```
HS.AddEntityToListUsingIDs(ParentID, ChildID)
```

**표 11-5 AddEntityToListUsingIDs 함수 구문**

매개변수	설명
<i>ParentID</i>	상위 엔티티의 ID입니다.
<i>ChildID</i>	상위 엔티티의 1차 하위 구성요소 ID입니다.

### 반환 값

없음

## AddMemberToList

멤버 목록에 지정된 멤버를 추가합니다. 이 함수는 멤버 목록 파일에서만 사용할 수 있습니다.

### 구문

```
HS.AddMemberToList("Member")
```

여기서 *Member*는 적합한 차원 멤버의 이름입니다.

### 반환 값

없음

### 예

```
HS.AddMemberToList "July"
```

## AddMemberToListUsingIDs

멤버 목록에 지정된 멤버를 추가합니다. 이 함수는 멤버 목록 파일에서만 사용할 수 있습니다.

### 구문

```
HS.AddMemberToListUsingIDs(MemberID)
```

여기서 *MemberID*는 적합한 차원 멤버의 ID입니다.

반환 값

없음

## Alloc

한 POV의 데이터를 다른 POV에 할당합니다. 이 함수는 할당 규칙에 사용할 수 있습니다.

구문

```
HS.Alloc ("SourcePOV", "DestPOV", "EntityList", "AllocExp", "PlugAccount")
```

**표 11-6 Alloc 함수 구문**

매개변수	설명
<i>SourcePOV</i>	<p>할당되는 데이터의 소스 POV입니다.</p> <p>계정 멤버를 지정해야 하며, 선택적으로 ICP 및 사용자정의 멤버를 지정할 수 있습니다. ICP 및 사용자정의 멤버를 지정하지 않는 경우 다음 값이 사용됩니다.</p> <ul style="list-style-type: none"> <li>기본 ICP 멤버는 ICP Top입니다.</li> <li>기본 사용자정의 멤버는 해당 계정의 TopMember입니다.</li> </ul>
<i>DestPOV</i>	<p>데이터를 할당할 위치를 식별하는 대상 POV입니다.</p> <p>계정 멤버를 지정해야 하며, 선택적으로 ICP 및 사용자정의 멤버를 지정할 수 있습니다. 다음과 같은 사용 규칙을 확인합니다.</p> <ul style="list-style-type: none"> <li>ICP 멤버를 지정하지 않는 경우 기본값은 [ICP None]입니다.</li> <li>사용자정의 멤버를 지정하지 않는 경우 기본값은 [None]입니다.</li> <li>엔티티 멤버를 지정하면 <i>EntityList</i> 인수가 [Base]인 경우 해당 멤버가 상위로 사용됩니다.</li> </ul>
<i>EntityList</i>	<p>데이터가 할당되는 엔티티를 식별하는 멤버 목록입니다. 시스템 정의 [Base] 엔티티 목록을 사용하거나 사용자 정의 목록을 사용할 수 있습니다.</p> <p>[Base] 시스템 정의 목록을 사용하는 경우 대상 POV에 지정된 엔티티가 상위 멤버로 사용됩니다. 사용자 정의 목록을 사용하고 이 목록에 상위 멤버가 포함되어 있는 경우 해당 멤버는 무시됩니다.</p>



표 11-6 (계속) Alloc 함수 구문

매개변수	설명
<i>AllocExp</i>	<p>각 엔티티에 할당할 데이터를 식별하는 표현식입니다. 이 표현식에는 다음 유형의 값이 포함될 수 있습니다.</p> <ul style="list-style-type: none"> <li>숫자</li> <li>숫자 값을 식별하는 계정 표현식. 계정 멤버를 지정할 수 있으며, 선택적으로 ICP 및 사용자정의 차원의 멤버를 지정할 수 있습니다. 다음과 같은 사용 규칙을 확인합니다. <ul style="list-style-type: none"> <li>사용자정의 멤버를 지정하지 않는 경우 기본값은 TopMember입니다.</li> <li>ICP 멤버를 지정하지 않는 경우 기본값은 ICP Top입니다.</li> <li>시나리오, 연도, 기간, 뷰 또는 값 멤버를 지정하지 않는 경우 기본값은 현재입니다.</li> <li>엔티티 멤버를 지정하지 않는 경우 기본값은 대상 엔티티입니다.</li> </ul> </li> </ul>
<i>PlugAccount</i>	<p>플러그 계정의 이름입니다. 이 인수는 선택 사항이며, 소스 POV 계정과 플러그 계정의 속성을 고려하여 소스 POV 금액을 전환하는 데 사용됩니다.</p> <p>계정 멤버를 지정해야 하며, 선택적으로 ICP 및 사용자정의 멤버를 지정할 수 있습니다. 다음과 같은 사용 규칙을 확인합니다.</p> <ul style="list-style-type: none"> <li>소스 POV 엔티티 멤버가 상위이거나 소스 POV 값 멤버가 엔티티 통화가 아닌 경우에는 이 인수를 사용할 수 없습니다.</li> <li>ICP 멤버를 지정하지 않는 경우 기본값은 ICP None입니다.</li> <li>사용자정의 멤버를 지정하지 않는 경우 기본값은 None입니다.</li> </ul>

**반환 값**

없습니다.

**예**

다음 예에서는 TangibleAssets 계정의 데이터가 Cash 계정에 할당됩니다.

```
Call HS.ALLOC
("A#TangibleAssets", "A#Cash", "NewEngland", "A#TangibleAssets", "A#Plug")
```

## AllowAdjFromChildren

지정된 엔티티나 상위 멤버에 대해 하위 멤버의 분개 계정을 허용할지 여부를 지정합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

 주:

트리 계층에서 멤버의 한 레벨 아래에 있는 경우 멤버는 1차 하위 구성요소입니다.

### 구문

```
HS.Entity.AllowAdjFromChildren("Entity")
```

```
HS.Entity.AllowAdjFromChildren(" ")
```

```
HS.Parent.AllowAdjFromChildren("Entity")
```

```
HS.Parent.AllowAdjFromChildren(" ")
```

여기서 *Entity*는 적합한 엔티티 또는 상위 멤버의 이름입니다.

현재 엔티티나 상위 멤버에 이 함수를 적용하려면 빈 문자열(" ")을 사용하십시오.

### 반환 값

지정된 엔티티에 대해 1차 하위 구성요소의 분개 게시를 허용하면 True이고, 1차 하위 구성요소의 분개 게시를 허용하지 않으면 False인 부울 표현식입니다.

### 예

다음 예에서는 France에 대해 1차 하위 구성요소의 분개 게시를 허용하는 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Entity.AllowAdjFromChildren("France") = TRUE then
```

```
...
```

```
End If
```

## AllowAdjs

지정된 엔티티나 상위 멤버에 대해 분개 게시를 허용할지 여부를 지정합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.Entity.AllowAdjs("Entity")
```

```
HS.Entity.AllowAdjs(" ")
```

```
HS.Parent.AllowAdjs("Entity")
```

```
HS.Parent.AllowAdjs(" ")
```

여기서 *Entity*는 적합한 엔티티 또는 상위 멤버의 이름입니다.

현재 엔티티나 상위 멤버에 이 함수를 적용하려면 빈 문자열(" ")을 사용하십시오.

### 반환 값

지정된 엔티티에 대해 분개 게시를 허용하면 True이고, 분개 게시를 허용하지 않으면 False인 부울 표현식입니다.

### 예

다음 예에서는 France에 대해 분개 게시를 허용하는 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Entity.AllowAdjs("France") = TRUE then
```

```
...
```

```
End If
```

## ApplicationName

규칙이 실행되는 애플리케이션의 이름을 반환합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.ApplicationName
```

**반환 값**

규칙이 실행되는 애플리케이션의 이름입니다.

**예**

```
strApp=HS.ApplicationName
```

## CalculateExchangeRate

하나의 통화에서 다른 통화로의 환율을 계산합니다.

**구문**

```
HS.CalculateExchangeRate (ScenarioID, YearID, PeriodID, EntityID,
RateAccountID, FromCurrencyID, ToCurrencyID)
```

**표 11-7 CalculateExchangeRate 함수 구문**

매개변수	설명
<i>ScenarioID</i>	시나리오 ID입니다.
<i>YearID</i>	연도 ID입니다.
<i>PeriodID</i>	기간 ID입니다.
<i>EntityID</i>	엔티티 ID입니다.
<i>RateAccountID</i>	환율 계정 ID입니다.
<i>FromCurrencyID</i>	소스 통화의 ID입니다.
<i>ToCurrencyID</i>	대상 통화의 ID입니다.

**반환 값**

환율입니다.

## CalculateRate

지정된 엔티티의 현재 환율을 가져옵니다.

**구문**

```
HS.CalculateRate (ScenarioID, YearID, PeriodID, EntityID, ValueID,
RateAccountID)
```

**표 11-8 CalculateRate 함수 구문**

매개변수	설명
<i>ScenarioID</i>	시나리오 ID입니다.
<i>YearID</i>	연도 ID입니다.
<i>PeriodID</i>	기간 ID입니다.
<i>EntityID</i>	엔티티 ID입니다.
<i>ValueID</i>	값 ID입니다.
<i>RateAccountID</i>	환율 계정 ID입니다.

**반환 값**

지정된 엔티티의 환율입니다.

## CalcStatus

지정된 POV의 계산 상태를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.CalcStatus("Scenario.Year.Period.Entity.Value")
```

**표 11-9 CalcStatus 함수 구문**

매개변수	설명
<i>Scenario</i>	적합한 시나리오 멤버의 이름입니다.
<i>Year</i>	적합한 연도입니다.
<i>Period</i>	적합한 기간입니다.
<i>Entity</i>	적합한 엔티티 멤버의 이름입니다.
<i>Value</i>	적합한 값 멤버의 이름입니다.

**반환 값**

지정된 POV의 계산 상태가 포함된 문자열입니다. 적합한 상태 코드는 아래에 나와 있습니다.

**표 11-10 계산 상태 코드**

상태 코드	설명
확인	지정된 시나리오, 연도, 기간, 엔티티 및 값 차원의 데이터가 하나도 변경되지 않았습니다.

표 11-10 (계속) 계산 상태 코드

상태 코드	설명
OK ND	확인 - 데이터 없음. 실제로 계산이 실행되었지만 데이터 없음에 대해 계산이 실행되지 않았습니다.
OK SC	확인 - 시스템 변경됨. 지정된 시나리오, 연도, 기간, 엔티티 및 값 차원의 데이터에 영향을 미칠 수 있는 변경이 수행되었습니다. 예를 들어 새 규칙 파일, 메타데이터 파일 또는 멤버 목록 파일이 로드되었거나 환율이 변경되었습니다. 값 10,000 등의 데이터 자체는 변경되지 않았습니다. 대신 차원 멤버 속성에서 일부 변경이 수행되었습니다. 예를 들어 계정과 연결된 소수 자릿수가 2로 설정되었으며 값을 10,000.00으로 변경해야 합니다.
CH	계산 필요. 지정된 시나리오, 연도, 기간, 엔티티 및 값 차원의 데이터 셀이 하나 이상 변경되었거나 메타데이터 매개변수나 규칙이 변경되었습니다. 따라서 계산이 실행되지 않았으므로 이 차원의 다른 데이터 셀이 최신 상태가 아닐 수 있습니다. 기본 레벨 엔티티의 경우 데이터 입력이나 데이터 파일 로드에 의해 데이터 셀이 입력되었을 수 있습니다. 임의 엔티티의 경우 분개 계시에 의해 데이터 셀이 입력되었을 수 있습니다.
CH ND	계산 필요 - 데이터 없음. 셀에서 계산이 처음으로 수행됨을 나타냅니다.
TR	환산 필요. 선택한 값 차원 멤버가 엔티티의 기본 통화가 아니며 환산된 값이 최신 상태가 아닐 수 있습니다.
TR ND	환산 필요 - 데이터 없음. 셀에서 환산이 처음으로 수행됨을 나타냅니다.
CN	통합 필요. 다음 중 하나가 변경되었으므로 지정된 시나리오, 연도, 기간, 엔티티 및 값 차원의 데이터가 최신 상태가 아닐 수 있습니다. <ul style="list-style-type: none"> <li>• 하위 엔티티의 데이터</li> <li>• 동일한 엔티티의 기본 통화 데이터</li> <li>• 메타데이터 매개변수 또는 규칙</li> </ul>
CN ND	통합 필요 - 데이터 없음. 상위에는 데이터가 없지만 1차 하위 구성요소 엔티티의 데이터가 변경되었습니다. 셀에서 통합이 처음으로 수행됨을 나타냅니다.
잠김	관리자가 지정된 시나리오, 연도, 기간, 엔티티 및 값 차원의 데이터를 잠갔습니다. 더 이상 수동으로 또는 계산을 통해 수정할 수 없습니다. <b>Alloc</b> 함수를 사용하여 잠긴 대상 POV의 데이터를 수정할 수 있습니다.
NoData	지정된 시나리오, 연도, 기간, 엔티티 및 값 차원의 데이터가 없습니다.
NoAccess	사용자에게 지정된 차원 멤버에 대한 권한이 없습니다.

## 예

다음 예에서는 지정된 POV의 상태가 "OK"인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.CalcStatus("S#Actual.Y#2014.P#January.E#Connecticut.  
V#<EntityCurrency>") = "OK" Then
```

...

```
End If
```

## CellTextUnitItem

통합, 계산 또는 환산하는 동안 처리할 셀 텍스트 단위 항목을 반환합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합

### 구문

```
HS.OpenCellTextUnit("", "[Default]", "Entity", "Ascending")
```

### 반환 값

The entry for the specified item.

## 예

```
Set MyCellTextUnit = HS.OpenCellTextUnit("", "[Default]", "Entity",
"Ascending")

NumItems= MyCellTextUnit.GetNumItems

For i = 0 to NumItems - 1

Scenario = MyCellTextUnit.Item(i).Scenario

Year = MyCellTextUnit.Item(i).Year

Period = MyCellTextUnit.Item(i).Period

Entity = MyCellTextUnit.Item(i).Entity

Value = MyCellTextUnit.Item(i).Value

Account = MyCellTextUnit.Item(i).Account

ICP = MyCellTextUnit.Item(i).ICP

Flow = MyCellTextUnit.Item(i).Custom("Flows")

Nature = MyCellTextUnit.Item(i).Custom("Nature")

...

...

...

CellText = MyCellTextUnit.Item(i).CellText

POV = MyCellTextUnit.Item(i).POV

Next
```



## Clear

계정, ICP 및 사용자정의 멤버의 조합에서 데이터를 제거합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.Clear "Account.ICP.Custom1.Custom2.Custom3.Custom4"
```

**표 11-11 Clear 함수 구문**

매개변수	설명
<i>Account</i>	적합한 계정 멤버의 이름입니다.
<i>ICP</i>	적합한 ICP 멤버의 이름입니다.
<i>Custom1, Custom2, Custom3, Custom4</i>	적합한 사용자정의1, 사용자정의2, 사용자정의3 및 사용자정의4 멤버의 이름입니다.

현재 엔티티, 기간, 시나리오, 값, 뷰 및 연도 멤버로 구성된 현재 POV와 교차하는 모든 셀에서 데이터를 제거하려면 다음 예와 같이 A# 문자 뒤에 ALL 키워드를 넣습니다.

```
HS.Clear "A#ALL"
```

셀과 사용자정의 또는 ICP 차원의 모든 교차를 지우려면 ALL 키워드를 사용하거나 A# 문자를 생략합니다. 다음 예에서는 A# 문자를 생략하여 사용자정의1 차원의 GolfBalls 멤버와 교차하는 모든 계정에서 데이터를 지웁니다.

```
HS.Clear "C1#GolfBalls"
```

### 반환 값

없습니다.

### 예

다음 예에서는 사용자정의1 차원의 Golfballs 멤버와 Sales 계정의 교차에 저장된 데이터를 지웁니다.

```
HS.Clear "A#Sales.C1#GolfBalls"
```

# Con

[Proportion] 및 [Elimination] 값 차원 멤버에 데이터를 넣습니다. 이 함수는 통합 규칙에 사용할 수 있습니다.

## 구문

Hs.Con (\*"DestPOV", Factor, "Nature")

**표 11-12 Con 함수 구문**

매개변수	설명
<i>DestPOV</i>	<p>다음 요소의 조합입니다.</p> <ul style="list-style-type: none"> <li>계정</li> <li>사용자정의1, 사용자정의2, 사용자정의3, 사용자정의4</li> <li>내부거래</li> <li>엔티티</li> <li>값</li> </ul>
<i>Factor</i>	<p>수치 연산자(+ - * /) 또는 HS.GetCell 같은 함수를 사용하는 표현식이나 숫자입니다.</p>
<i>Nature</i>	<p>감사 용도로 사용되는 문자열입니다. 이 문자열은 데이터베이스에 저장되며 트랜잭션의 회계 목적에 대한 정보를 제공합니다.</p> <p>사용자가 통합을 실행한 후 통합 소스 및 대상 트랜잭션을 볼 수 있도록 하려면 이 매개변수에 텍스트를 포함해야 합니다. 텍스트를 포함하지 않으면 트랜잭션 정보가 저장되지 않습니다. 데이터 그리드 소스 트랜잭션 또는 대상 트랜잭션 옵션이나 엔티티 세부정보 보고서에서 트랜잭션 정보를 볼 수 있습니다.</p> <p>특성 문자열 정보를 사용하여 통합 및 제거 항목에 대한 분개 보고서를 생성할 수도 있습니다. 분개 보고서에서 이 데이터를 보려는 경우 이 매개변수가 필요합니다. Proportion 및 Elimination 항목에 대한 분개는 RTS/RTD 테이블에서 보고됩니다. 요구사항에 따라 RTS/RTD 테이블 항목의 볼륨을 줄이려면 [Proportion] HS.Con 항목이 아니라 [Elimination] HS.Con 항목에 대한 분개 데이터를 생성하는 것이 적합할 수 있습니다.</p> <p>RTS/RTD 테이블 항목을 생성하는 Nature 매개변수를 사용하면 데이터베이스 크기가 증가하며 통합 성능에 영향을 줄 수 있습니다.</p>

## 반환 값

없습니다.

**예**

```
Call HS.Con ("V#[Elimination]",-1*dPCon,"")
```

```
Call HS.Con ("V#[Elimination]",-1*dPCon, "Elimination")
```

```
Call HS.Con ("V#[Elimination]"PCON, "DefaultConsolidation")
```

## Consol1, Consol2, Consol3

지정된 상위 엔티티 노드에 대한 콘솔1, 콘솔2 또는 콘솔3 계정의 값을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 할당

**구문**

시나리오, 연도, 기간 및 상위 엔티티 멤버의 조합입니다.

```
HS.Node.Consoln ("S#Scenario.Y#Year.P#Period.E#Parent.Entity")
```

```
HS.Node.Consoln ("")
```

 **주:**

현재 시나리오, 연도, 기간 및 엔티티에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-13 Consol1, Consol2, Consol3 함수 구문**

매개변수	설명
<i>Scenario</i>	적합한 시나리오 멤버의 이름입니다.
<i>Year</i>	적합한 연도입니다.
<i>Period</i>	적합한 기간입니다.
<i>Parent.Entity</i>	적합한 상위 엔티티 노드의 이름입니다.

**반환 값**

콘솔1, 콘솔2 또는 콘솔3 시스템 계정의 값입니다.

**예**

다음 예에서는 실제 시나리오에서 Group1.Ent1 노드의 Consol1 값을 가져옵니다.

```
dVar1 = HS.Node.Consol1 ("S#Actual.E#Group1.Ent1")
```

## ConsolidateYTD

현재 시나리오 멤버나 지정된 시나리오 멤버가 통합된 연 누계인지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.Scenario.ConsolidateYTD("Scenario")
```

```
HS.Scenario.ConsolidateYTD("")
```

여기서 Scenario는 적합한 시나리오 멤버의 이름입니다.

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**반환 값**

연 누계 방법을 사용하여 시나리오가 통합되면 True이고, 주기적 방법을 사용하여 시나리오가 통합되면 False인 부울 표현식입니다.

**예**

다음 예에서는 실제 시나리오가 연 누계 방법을 사용하여 통합되도록 설정된 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Scenario.ConsolidateYTD("Actual") = TRUE Then
```

```
...
```

```
End If
```

## ContainsCellText

지정된 셀에 셀 텍스트가 포함되어 있는지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산

- 환산
- 통합
- 할당

## 구문

```
HS.ContainsCellText ("POVExpression")
```

여기서 *POVExpression*은 멤버 조합입니다. 차원을 지정하지 않으면 다음 기본값이 사용됩니다.

- 계정 - [none]
- ICP - [ICP None]
- 사용자정의1...4 - [None]
- 시나리오 - 현재 시나리오 멤버
- 엔티티 - 현재 엔티티 멤버
- 값 - 현재 값 멤버
- 연도 및 기간 - 현재 멤버

### 주:

기본 항목은 계산 규칙에서 이 함수를 사용할 때 적용됩니다. 할당, 환산 또는 통합 규칙에서 사용할 경우 전체 서브큐브 계정/ICP/사용자정의 POV를 지정해야 합니다. 시나리오, 연도, 기간, 뷰, 엔티티 및 값 차원은 기본적으로 현재 멤버로 지정됩니다.

## 반환 값

지정된 데이터 셀에 임의 셀 텍스트 레이블의 셀 텍스트가 포함되어 있으면 True이고, 그렇지 않으면 False인 부울 표현식입니다.

## 예

다음 예에서는 지정된 셀에 임의 셀 텍스트 레이블의 셀 텍스트가 포함되어 있지 않은 경우 If...Then 라인과 End If 라인 사이의 명령문이 실행됩니다.

```
If HS.ContainsCellText ("A#Sales.C1#Prod1.C2#Region1") = "False" then
```

```
...
```

```
End If
```

 주:

셀 텍스트 레이블을 사용하는 경우 기존 셀 텍스트 레이블([Default] 포함)에 셀 텍스트 항목이 포함되어 있으면 이 함수에서 True 값이 반환됩니다.

## ContainsCellTextWithLabel

지정된 레이블의 셀 텍스트가 셀에 포함되어 있고 셀에서 셀 텍스트 정보를 업데이트하는지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.ContainsCellTextWithLabel("POVExpression", "CellTextLabel")
```

여기서 POVExpression 은 멤버 조합입니다. 차원을 지정하지 않으면 다음 기본값이 사용됩니다.

- 계정 - [none]
- ICP - [ICP None]
- 사용자정의1...4 - [None]
- 시나리오 - 현재 시나리오 멤버
- 엔티티 - 현재 엔티티 멤버
- 값 - 현재 값 멤버
- 연도 및 기간 - 현재 멤버

CellTextLabel은 기본 셀 텍스트 레이블([Default])이거나 로드된 적합한 셀 텍스트 레이블 중 하나입니다.

 주:

기본 항목은 계산 규칙에서 이 함수를 사용할 때 적용됩니다. 할당, 환산 또는 통합 규칙에서 사용할 경우 전체 서브큐브 계정/ICP/사용자정의 POV를 지정해야 합니다. 시나리오, 연도, 기간, 뷰, 엔티티 및 값 차원은 기본적으로 현재 멤버로 지정됩니다.

### 반환 값

지정된 데이터 셀에 지정된 셀 텍스트 레이블의 셀 텍스트가 포함되어 있으면 True이고, 그렇지 않으면 False인 부울 표현식입니다.

### 예

다음 예에서는 지정된 셀에 지정된 셀 텍스트 레이블의 셀 텍스트가 포함되어 있지 않은 경우 If...Then 라인과 End If 라인 사이의 명령문이 실행됩니다.

```
If HS.ContainsCellTextWithLabel ("A#Sales.C1#Prod1.C2#Region1", "Rating") =
"False" then ... End If
```

## 통화

애플리케이션이나 현재 값 차원 멤버의 통화를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.AppSettings.Currency
```

```
HS.Value.Currency
```

### 반환 값

애플리케이션이나 값 멤버의 통화 이름이 포함된 문자열입니다.

### 예

다음 예에서는 애플리케이션의 통화가 유로인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.AppSettings.Currency = "Euro" Then
```

```
...
```

```
End If
```

## CustomTop

현재 또는 지정된 계정 멤버의 CustomTopMember를 반환합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.Account.CustomTop("Member","", "CustomDimName")
```

```
HS.Account.CustomTop("", "CustomDimName")
```

여기서 *CustomDimName*은 사용자정의 차원이고 *Member*는 계정 차원 멤버입니다. 지정되지 않은 멤버는 Sub Consolidate 서브루틴에만 적합합니다.



### 주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

## 반환 값

지정된 계정 멤버의 CustomTopMember입니다.

## 예

```
HS.Account.CustomTop("Prod", "Sales")
```

# DataUnitItem

통합, 계산 또는 환산하는 동안 처리할 데이터 단위 항목을 반환합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합

## 구문

```
HS.OpenDataUnit("")
```

## 반환 값

지정된 항목의 멤버 레이블입니다.



**예**

```
Set MyDataUnit = HS.OpenDataUnit("")

NumItems= MyDataUnit.GetNumItems

For i = 0 to NumItems - 1

Account = MyDataUnit.Item(i).Account

ICP = MyDataUnit.Item(i).ICP

Flow = MyDataUnit.Item(i).Custom("Flows")

Nature = MyDataUnit.Item(i).Custom("Nature")

...

...

...

Data = MyDataUnit.Item(i).Data

POV = MyDataUnit.Item(i).POV

Next
```

## 소수 자릿수

지정된 계정의 소수 자릿수를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.Account.Decimal("AccountName")
```

```
HS.Account.Decimal("")
```

### 주:

Sub Consolidate 서브루틴을 사용할 때 현재 계정에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

```
HS.Account.Decimal(Var1)
```

**표 11-14** Decimal 함수 구문

매개변수	설명
<i>AccountName</i>	적합한 계정 멤버의 이름입니다
<i>Var1</i>	계정 멤버를 나타내는 VBScript 변수입니다

### 반환 값

계정에 지정된 소수 자릿수를 나타내는 정수입니다. 적합한 값은 0-9입니다.

### 예

다음 예에서는 Sales 계정에 지정된 소수 자릿수가 2인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Account.Decimal("Sales") = 2 Then
```

```
...
```

```
End If
```

## DefaultFreq

현재 시나리오 멤버나 지정된 시나리오 멤버의 기본 빈도를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.Scenario.DefaultFreq("Scenario")
```

```
HS.Scenario.DefaultFreq("")
```



### 주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

여기서 *Scenario*는 적합한 시나리오 멤버의 이름입니다.

## 반환 값

시나리오의 기본 빈도가 포함된 문자열입니다.

## 예

다음 예에서는 실제 시나리오의 기본 빈도가 YTD인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Scenario.DefaultFreq("Actual") = "YTD" Then
```

```
...
```

```
End If
```

## DefaultParent

현재 멤버나 지정된 멤버의 기본 상위 멤버를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.<Object>.DefaultParent("Member")
```

```
HS.<Object>.DefaultParent("")
```

```
HS.Custom("Label").DefaultParent("Member")
```

```
HS.Custom("Label").DefaultParent("")
```

여기서 *Member*는 적합한 차원 멤버입니다.



### 주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-15** DefaultParent 함수 구문

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의(사용자정의 차원 레이블)</li> <li>엔티티</li> <li>시나리오</li> </ul>
멤버	선택한 객체에 따라 다음 차원 중 하나에서 적합한 멤버의 이름입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의</li> <li>엔티티</li> <li>시나리오</li> </ul>

### 반환 값

멤버의 기본 상위 멤버가 포함된 문자열입니다.

## DefaultTranslate

규칙을 건너뛰어 환산을 계산합니다. 이 함수는 애플리케이션 설정을 재정의하며 SubTranslate 규칙에만 사용할 수 있습니다.

## 구문

```
HS.DefaultTranslate(dRateForBalanceAccounts, dRateForFlowAccounts,
bUsePVAForFlowAccounts, bUsePVAForBalanceAccounts)
```

**표 11-16 DefaultTranslate 함수 구문**

매개변수	설명
<i>dRateForBalanceAccounts</i>	환율을 나타내는 숫자입니다.
<i>dRateForFlowAccounts</i>	환율을 나타내는 숫자입니다.
<i>bUserPVAForFlowAccounts</i>	True 또는 False입니다.
<i>bUsePVAForBalanceAccounts</i>	True 또는 False입니다.

## 반환 값

없습니다.

## 예

다음 예에서는 상위 멤버가 United States인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Parent.Member="UnitedStates" Then

    HS.DefaultTranslate .25, .27, True, False

End If
```

## DefaultView

현재 시나리오 멤버나 지정된 시나리오 멤버의 기본 뷰를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.Scenario.DefaultView("Scenario")
```

```
HS.Scenario.DefaultView("")
```

 주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

여기서 *Scenario*는 적합한 시나리오 멤버의 이름입니다.

**반환 값**

지정된 시나리오의 기본 뷰가 포함된 문자열입니다. 적합한 값은 YTD와 주기적입니다.

**예**

다음 예에서는 실제 시나리오의 기본 뷰가 YTD인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Scenario.DefaultView("Actual") = "YTD" Then
```

```
...
```

```
End If
```

## DefCurrency

현재 엔티티 또는 상위 멤버나 지정된 엔티티 또는 상위 멤버의 기본 통화를 가져옵니다. 엔티티를 지정하면 엔티티 통화가 반환됩니다. 상위 통화를 가져오려면 상위 엔티티를 지정해야 합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.<Object>.DefCurrency("Entity")
```

```
HS.<Object>.DefCurrency("")
```

 주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

표 11-17 DefCurrency 함수 구문

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>엔티티</li> <li>상위</li> </ul>
Entity	적합한 엔티티 멤버의 이름입니다

**반환 값**

지정된 엔티티나 상위의 기본 통화가 포함된 문자열입니다.

**예**

다음 예에서는 Connecticut의 기본 통화가 EastRegion의 기본 통화와 다른 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Entity.DefCurrency("Connecticut") <>
HS.Parent.DefCurrency("EastRegion") Then
```

```
...
```

```
End If
```

## Down

지정된 상위 엔티티 노드의 직접 소유 지분을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

시나리오, 연도, 기간 및 상위 엔티티 멤버의 조합입니다.

```
HS.Node.Down("S#Scenario.Y#Year.P#Period.E#Parent.Entity")
```

```
HS.Node.Down("")
```

**주:**

현재 노드에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-18 DOwn 함수 구문**

매개변수	설명
<i>Scenario</i>	적합한 시나리오 멤버의 이름입니다
<i>Year</i>	적합한 연도입니다.
<i>Period</i>	적합한 기간입니다.
<i>Parent.Entity</i>	적합한 상위 엔티티 노드의 이름입니다

**반환 값**

직접 소유 지분을 나타내는 숫자입니다.

**예**

```
dVar1 = HS.Node.DOwn("S#Actual.Y#2014.P#Q1.E#Group1.Ent1")
```

## Dynamic

계산이 필요한 동적 계정에 대한 공식을 지정합니다. 이 함수는 동적 규칙에 사용할 수 있습니다.

이 함수는 동일한 서브큐브의 데이터만 참조할 수 있습니다. 다른 서브큐브의 데이터를 참조해야 하는 경우 다른 큐브의 정보를 저장할 계정을 생성해야 할 수도 있습니다. 예를 들어 공식에서 이전 연도의 데이터를 참조하려면 작년 데이터 정보를 저장할 계정을 사용하여 동일한 큐브 내의 동적 계산에서 참조할 수 있도록 해야 합니다.

**주:**

Dynamic 함수에 HS.View.PeriodNumber 함수를 포함할 수 있습니다. 예:

```
HS.Dynamic "A#AvgUnits = A#AccumUnits.I#ICP None.C1#None.C2#None /
HS.View.PeriodNumber"
```

**구문**

```
HS.Dynamic "DestPOV = Expression"
```



**표 11-19 Dynamic 함수 구문**

매개변수	설명
<i>DestPOV</i>	유형이 [동적]으로 설정된 적합한 계정 멤버입니다. 계산을 실행할 뷰를 지정할 수도 있습니다. 뷰를 지정하지 않으면 [주기적] 및 YTD에 대해 공식이 실행됩니다. 각 뷰에 서로 다른 공식을 사용하려면 공식에 [주기적] 또는 YTD를 지정해야 합니다.
<i>Expression</i>	계산 표현식입니다.



**주:**

Dynamic 함수는 IF...THEN 문을 지원하지 않습니다.

**반환 값**

없습니다.

**예**

Sub Dynamic

```
HS.Dynamic "A#GM% = A#GM / (A#Sales * 100) / HS.View.PeriodNumber"
```

End Sub

GM% 계정의 예상 결과:

Custom1	판매	GM	GM%(동적 계산)	공식 기반 계산
제품	600	140	23.33%	140 / 600 * 100
P1	100	10	10%	10 / 100 * 100
P2	200	40	20%	40 / 200 * 100
P3	300	90	30%	90 / 300 * 100

## Exp

계정, ICP 및 사용자정의1...4 멤버의 조합에 데이터를 넣습니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 할당

## 구문

```
HS.Exp "DestPOV = Expression"
```

표 11-20 Exp 함수 구문

매개변수	설명
<i>DestPOV</i>	데이터를 넣을 위치를 식별하는 대상 POV입니다. 다음과 같은 사용 규칙을 확인합니다. 데이터베이스에 불필요한 값을 채우지 않도록 하려면 데이터 위치 정의와 관련해서 규칙이 가능한 한 명시적이어야 합니다. 계정 차원에 대해 명확하게 정의된 ICP 및 사용자정의 차원 교차를 포함하는 것이 좋습니다. 명확하게 정의된 이러한 교차점에서는 Oracle Hyperion Financial Management 검증 확인을 통해 부적합한 교차점에 데이터를 쓰지 않도록 방지합니다.
<i>Expression</i>	계산 표현식입니다.

데이터의 대상은 등호 왼쪽에 있고 지정할 데이터는 오른쪽에 있습니다. 다음 예에서는 UnitsSold 계정과 사용자정의 차원의 [None] 멤버를 교차하는 셀을 40000으로 설정합니다.

```
HS.Exp "A#UnitsSold.C1#[None].C2#[None].C3#[None].C4#[None]" _
```

```
& " = 40000"
```

등호의 오른쪽에 계정 표현식 문자를 사용하여 차원 멤버를 나타낼 수 있습니다. 따라서 셀 그룹의 데이터를 다른 셀 그룹에 지정할 수 있습니다. 다음 예에서는 사용자정의 차원의 [None] 멤버와 Taxes 계정을 교차하는 셀을 지정된 사용자정의 차원과 Sales 계정을 교차하는 셀에 있는 데이터의 6%로 설정합니다.

```
HS.Exp "A#Taxes.C1#[None].C2#[None].C3#[None].C4#[None]" _
```

```
& " = A#Sales.C1#AllProducts.C2#AllCustomers.C3#[None]" _
```

```
& ".C4#[None] * .06"
```

 **팁:**

하나의 Exp 문으로 여러 계정을 설정할 수 있습니다. 동시에 여러 계정 설정을 참조하십시오.

## 차원 교차점 고려 사항

Exp가 데이터를 가져오는 소스 차원 교차점과 Exp가 데이터를 넣는 대상 차원 교차점을 지정하지 않으면 소스 및 대상 교차점은 다음과 같은 요소에 의해 결정됩니다.

- 대상. 내부거래 파트너 차원이나 사용자정의 차원의 멤버를 등호 왼쪽에 지정하지 않으면 Exp는 계정과 차원의 각 적합한 교차에 데이터를 넣습니다. 대상 계정을 지정하지 않으면 Oracle Hyperion Financial Management에서 현재 POV에 적합한 모든 계정에 데이터를 삽입합니다. **동시에 여러 계정 설정**을 참조하십시오.
- 소스. 차원의 멤버를 등호 오른쪽에 지정하지 않으면 다음과 같은 여러 결과가 나타날 수 있습니다.
  - 차원에 멤버가 하나뿐인 경우 Exp는 이 멤버와 소스 계정의 교차에서 데이터를 가져옵니다.
  - 차원에 소스 계정과의 적합한 교차가 하나뿐인 경우 Exp는 이 교차에서 데이터를 가져옵니다.
  - 차원에 소스 계정과 교차하는 멤버가 여러 개 있는 경우 등식 왼쪽 부분에 의해 데이터의 소스 교차가 결정됩니다.
    - \* 왼쪽에 멤버가 지정된 경우 Exp는 이 멤버와 소스 계정의 교차에서 데이터를 가져옵니다.
    - \* 왼쪽에 멤버가 지정되지 않은 경우 Exp는 대상 계정과 차원 멤버의 각 적합한 교차에 데이터를 넣습니다. Exp는 멤버와 소스 계정의 해당 교차에서 대상 교차의 데이터를 가져옵니다.

### 주:

소스 교차가 부적합한 경우에는 Exp에서 해당 대상 교차의 데이터를 변경하지 않습니다.

이러한 고려 사항이 설명된 자세한 예는 [표현식 및 차원 교차 고려 사항](#)을 참조하십시오.

## 기간 및 연도 키워드

동적 규칙을 작성하기 위해 멤버 이름 대신 [테이블 1](#)에 있는 키워드를 사용하여 대상 연도 또는 기간 차원의 멤버를 나타낼 수 있습니다.

**표 11-21** 기간 및 연도 키워드와 설명

키워드	설명
CUR	현재 기간이나 연도입니다.
FIRST	애플리케이션에 적합한 첫 번째 기간이나 연도입니다.
LAST	애플리케이션에 적합한 마지막 기간이나 연도입니다.
NEXT	현재 기간이나 연도 뒤에 오는 기간이나 연도입니다.
PRIOR	현재 기간이나 연도 앞에 오는 기간이나 연도입니다.

 주:

기간 및 연도 키워드는 대/소문자를 구분하며 모두 대문자여야 합니다.

기간 및 연도 키워드와 함께 더하기(+) 및 빼기(-) 기호를 사용할 수 있습니다. 다음 예에서는 MiscPast 계정을 현재 연도보다 2년 전의 Misc 계정 금액으로 설정합니다.

```
HS.Exp "A#MiscPast = A#Misc.Y#CUR-2"
```

Prior, First, Last, Current 또는 Next 키워드 바로 뒤에 더하기(+) 및 빼기(-) 기호와 숫자를 사용하는 경우 등식 순서에 대한 올바른 구문을 사용해야 합니다. 이 경우 다음 방법 중 하나를 사용하여 규칙을 작성할 수 있습니다.

항상 괄호를 사용하여 변수를 정확하게 구분합니다. 예:

```
HS.Exp "A#9001_Group.C4#[None] = A#9001_Group.P#Prior" & "+" & VAR &
"*A#9001_Group.V#[ParentTotal])"
```

또는

```
HS.Exp "A#9001_Group.C4#[None] = (A#9001_Group.P#Prior" & "+" & VAR & "
*A#9001_Group.V#[Parent Total])"
```

## 수치 계산

등호 오른쪽에서 더하고 빼고 곱하고 나눌 수 있습니다. 다음과 같은 표준 VBScript 문자를 사용해야 합니다.

+ - \* /

다음 예에서는 이전 연도의 Taxes 및 Purchases 계정의 값을 더한 후 그 합계를 Cash 계정에 삽입합니다.

```
HS.Exp "A#Cash = A#Taxes.Y#PRIOR + A#Purchases.Y#PRIOR"
```

 주:

[데이터 없음] 상태의 계정을 사용하여 곱하거나 나누면 등호 왼쪽의 계정 데이터가 변경되지 않습니다. 0은 데이터로 간주됩니다. 즉, 데이터로 0.00을 포함하는 계정은 [데이터 없음] 상태가 *아닙니다*.

## Exp 안에 다른 함수 배치

함수에서 단일 숫자 값이 반환되는 경우 Exp 함수 안에 함수를 중첩할 수 있습니다. 하지만 문자열 인수가 포함된 함수를 중첩하는 경우 문자열을 따옴표로 묶을 수 없습니다. 다음 예에서는 NumBase 함수가 Exp 함수에 중첩되므로 문자열 인수를 따옴표로 묶지 *않습니다*.

```
HS.Exp "A#SalesAlloc = A#Sales/HS.Entity.NumBase(Regional) "
```

## 동시에 여러 계정 설정

현재 POV와 교차하는 모든 계정에 데이터를 삽입하려면 계정 표현식에 All을 사용합니다. 계정 표현식을 사용하여 모든 계정의 기초 잔액을 설정할 수 있습니다. 다음 예에서 IsFirst 함수는 현재 기간이 첫 번째 기간인지 여부를 테스트합니다. 첫 번째 기간인 경우 Exp에서 현재 기간에 대한 각 계정의 값을 이전 연도의 마지막 기간에서 가져온 계정의 값으로 설정합니다.

```
If HS.Period.IsFirst = TRUE Then
```

```
    HS.Exp "A#ALL = A#ALL.Y#PRIOR.P#LAST"
```

```
End If
```

계정과 사용자정의 또는 내부거래 파트너 차원의 모든 교차에 데이터를 삽입하려면 All 키워드를 사용하거나 A# 문자를 생략합니다. A# 문자를 생략하는 다음 예에서는 사용자정의1 차원의 GolfBalls 멤버와 교차하는 각 계정에 데이터를 삽입합니다. GolfBalls와 계정의 적합한 각 교차에 대해 이전 기간의 교차 금액이 현재 기간의 교차에 삽입됩니다.

```
HS.Exp "C1#GolfBalls = C1#GolfBalls.P#PRIOR"
```

### 반환 값

없습니다.

### 예

다음 예에서는 StateTax 계정의 금액을 설정합니다. 이 금액은 2014년 Sales 계정의 금액에 2014년 StateRate 계정의 금액을 곱하여 계산됩니다.

```
HS.Exp "A#StateTax = A#Sales.Y#2014 * A#StateRate.Y#2014"
```

## 표현식 및 차원 교차 고려 사항

다음 예에서는 [차원 교차점 고려 사항](#)에 소개된 고려 사항에 대해 설명합니다. 다음과 같은 교차 유형에 대해 다룹니다.

- 모든 교차가 소스 및 대상 계정에 적합합니다. [모든 교차점 적합](#)을 참조하십시오.
- 일부 교차는 소스 및 대상 계정에 적합하지만 다른 교차는 부적합합니다. [부적합한 교차점](#)을 참조하십시오.
- 한 멤버가 소스 계정에 적합합니다. [오른쪽에 하나의 적합한 멤버 지정](#)을 참조하십시오.

이러한 모든 예에서는 TargAcct 및 SourceAcct 계정을 Member001, Member002 및 Member003이라는 사용자정의1 차원의 멤버와 함께 사용합니다. 모든 예의 소스 교차점 데이터는 [테이블 1](#)에 나열되어 있습니다.

**표 11-22 차원 교차 데이터 예**

Member	SourceAcct 교차 데이터
Member001	10
Member002	데이터 없음 상태
Member003	15

## 모든 교차점 적합

다음 예에서는 TargAcct 및 SourceAcct 계정과 사용자정의1 멤버의 모든 교차가 적합합니다.

```
HS.Exp "A#TargAcct = A#SourceAcct"
```

이 함수는 TargAcct 계정과 사용자정의1 멤버의 교차에 이 데이터를 넣습니다.

사용자정의1 멤버	데이터	교차점
Member001	10	SourceAcct 및 Member001
Member002	---	해당 없음. SourceAcct 및 Member002의 교차가 [데이터 없음] 상태이므로 TargAcct 및 Member002의 교차는 변경되지 않습니다.
Member003	15	SourceAcct 및 Member003

다음 예에서는 등호 왼쪽에 Member001 멤버를 지정하여 Exp를 사용합니다.

```
HS.Exp "A#TargAcct.C1#Member001 = A#SourceAcct"
```

TargAcct 및 Member001의 교차는 10으로 설정됩니다. 왼쪽에 Member001이 지정되었으므로 Exp는 SourceAcct 및 Member001의 교차에서 데이터를 가져옵니다.

다음 예에서는 등호 오른쪽에 Member003을 배치하여 EXP를 사용합니다.

```
HS.Exp "A#TargAcct = A#SourceAcct.C1#Member003"
```

이 함수는 사용자정의1 멤버와 TargAcct 계정의 교차에 이 데이터를 넣습니다.

사용자정의1 멤버	데이터	교차점
Member001	15	SourceAcct 및 Member003
Member002	15	SourceAcct 및 Member003
Member003	15	SourceAcct 및 Member003

## 부적합한 교차점

다음 예에서는 소스 계정과 대상 계정에 각각 부적합한 교차가 있습니다.

- SourceAcct. Member002와 Member003은 적합하고 Member001은 부적합합니다.
- TargAcct. Member001과 Member002는 적합하고 Member003은 부적합합니다.

다음 예에서는 등호의 한 쪽에 사용자정의1 멤버를 지정하지 않고 Exp를 사용합니다.

```
HS.Exp "A#TargAcct = A#SourceAcct"
```

Exp는 다음 작업을 수행하려고 하므로 TargAcct 계정의 데이터를 변경하지 않습니다.

- TargAcct.Member001 = SourceAcct.Member001. SourceAcct와 Member001은 부적합한 교차입니다.
- TargAcct.Member002 = SourceAcct.Member002. SourceAcct와 Member002의 교차가 [데이터 없음] 상태이므로 TargAcct와 Member002의 교차는 변경되지 않습니다.
- TargAcct.Member003 = SourceAcct.Member003. TargAcct와 Member003은 부적합한 교차입니다.

다음 예에서는 등호 왼쪽에 Member001을 지정하여 Exp를 사용합니다.

```
HS.Exp "A#TargAcct.C1#Member001 = A#SourceAcct"
```

Exp가 부적합한 교차(SourceAcct 및 Member001)에서 데이터를 검색하려고 하므로 TargAcct.Member001은 변경되지 않습니다.

다음 예에서는 등호 오른쪽에 Member003을 지정하여 Exp를 사용합니다.

```
HS.Exp "A#TargAcct = A#SourceAcct.C1#Member003"
```

이 함수는 사용자정의1 멤버와 TargAcct 계정의 교차에 이 데이터를 넣습니다.

사용자정의1 멤버	데이터	교차점
Member001	15	SourceAcct 및 Member003
Member002	15	SourceAcct 및 Member003
Member003	N/A	해당 사항이 없습니다. Member003은 TargAcct 계정의 부적합한 교차입니다.

## 오른쪽에 하나의 적합한 멤버 지정

다음 예에서는 소스 계정에 하나의 적합한 멤버만 있고 대상 계정에 두 개의 적합한 멤버가 있습니다.

- SourceAcct. Member003만 적합한 교차입니다.
- TargAcct. Member001과 Member002는 적합하고 Member003은 부적합합니다.

다음 예에서는 등호의 한 쪽에 사용자정의1 멤버를 지정하지 않고 Exp를 사용합니다.

```
HS.Exp "A#TargAcct = A#SourceAcct"
```

이 함수는 사용자정의1 멤버와 TargAcct 계정의 교차에 이 데이터를 넣습니다.

사용자정의1 멤버	데이터	교차점
Member001	15	SourceAcct 및 Member003(SourceAcct 계정의 유일한 적합한 교차)
Member002	15	SourceAcct 및 Member003(SourceAcct 계정의 유일한 적합한 교차)
Member003	N/A	해당 없음. Member003은 TargAcct 계정의 부적합한 교차입니다.

다음 예에서는 등호 왼쪽에 Member001을 지정하여 Exp를 사용합니다.

```
HS.Exp "A#TargAcct.C1#Member001 = A#SourceAcct"
```

TargAcct 및 Member001의 교차는 15로 설정됩니다. 이 값은 SourceAcct 및 Member003의 교차에 있는 데이터입니다.

#### 팁:

SourceAcct 계정과 사용자정의1 차원의 적합한 교차가 둘 이상 있으면 Exp는 SourceAcct와 Member001의 교차에서 데이터를 가져옵니다. 부적합한 교차이면 대상 계정이 변경되지 않습니다.

## GetCell

셀에 포함된 데이터를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

#### 구문

```
HS.GetCell("POVExpression")
```

여기서 *POVExpression*은 적합한 POV입니다.



**반환 값**

지정된 셀에 저장된 데이터입니다.

**주:**

함수에서 둘 이상의 값이 반환되는 경우 오류가 발생합니다.

**예**

다음 예에서는 사용자정의1 차원의 Golfballs 멤버와 Sales 계정의 교차에 저장된 금액을 dData 변수에 지정합니다.

```
Dim dData
```

```
dData = HS.GetCell("A#Sales.I#[ICP  
None].C1#Golfballs.C2#Customer2.C3#Increases.C4#[None]")
```

## GetCellNoData

셀에 포함된 데이터를 가져오고 셀에 데이터가 포함되어 있는지 여부를 나타냅니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.GetCellNoData("POV", Var1)
```

**표 11-23 GetCellNoData 함수 구문**

매개변수	설명
<i>POV</i>	적합한 POV입니다.
<i>Var1</i>	셀에 데이터가 포함되어 있는지 여부를 지정하는 변수입니다.

**반환 값**

가능한 반환 값은 다음과 같이 셀의 내용에 따라 달라집니다.

- 셀에 실제 데이터가 포함되어 있으면 데이터 값이 반환되고 *Var1*에 대해 반환되는 부울 값이 False입니다.

- 셀에 데이터가 포함되어 있지 않으면 데이터 값으로 0이 반환되고 *Var1*에 대해 반환되는 부울 값이 True입니다.
- 셀에 파생된 데이터가 포함되어 있으면 파생된 값이 반환되고 *Var1*에 대해 반환되는 부울 값이 False입니다.

### ▲ 주의:

인수로 인해 GetCellNoData에서 값이 둘 이상 반환되면 오류가 발생합니다.

### 예

다음 예에서는 Sales 계정의 금액이 dSalesAmt 변수에 지정됩니다. Sales 계정에 데이터가 없는 경우 If Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
dSalesAmt = HS.GetCellNoData("A#Sales.I#[ICP
None].C1#[None].C2#[None].C3#[None].C4#[None]",bIsNoData)
```

```
If bIsNoData = TRUE then
```

```
...
```

```
End If
```

## GetCellRealData

셀에 포함된 데이터를 가져오고 셀에 실제 데이터가 포함되어 있는지 여부를 나타냅니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.GetCellRealData("POV", Var1)
```

**표 11-24** GetCellRealData 함수 구문

매개변수	설명
<i>POV</i>	적합한 POV입니다.
<i>Var1</i>	셀에 실제 데이터가 포함되어 있는지 여부를 지정하는 변수입니다.

### 반환 값

가능한 반환 값은 다음과 같이 셀의 내용에 따라 달라집니다.

- 셀에 실제 데이터가 포함되어 있으면 데이터 값이 반환되고 *Var1*에 대해 반환되는 부울 값이 True입니다.
- 셀에 데이터가 포함되어 있지 않으면 데이터 값으로 0이 반환되고 *Var1*에 대해 반환되는 부울 값이 False입니다.
- 셀에 파생된 데이터가 포함되어 있으면 파생된 값이 반환되고 *Var1*에 대해 반환되는 부울 값이 False입니다.

### 예

```
dData = HS.GetCellRealData("A#Sales.C1#Prod1",bIsRealData)
```

```
If bIsRealData = TRUE then
```

```
...
```

```
End If
```

## GetCellText

POV에서 기본 셀 텍스트를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.GetCellText("POVExpression")
```

여기서 *POVExpression*은 적합한 POV입니다.

### 반환 값

POV의 [Default] 레이블 셀 텍스트입니다.

### 예

```
HS.GetCellText("S#Actual.Y#2014.P#January.E#Connecticut.V#<Entity  
Currency>.A#Sales.I#[ICP None].C1#[None].C2#[None].C3#[None].C4#[None]")
```

 주:

셀 텍스트 레이블을 사용하는 경우 이 함수는 셀 텍스트 레이블 [Default]의 셀 텍스트를 검색합니다.

## GetCellTextWithLabel

POV에서 지정된 셀 텍스트 레이블의 셀 텍스트를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.GetCellTextWithLabel("POVExpression", "CellTextLabel")
```

여기서 `POVExpression` 은 멤버 조합입니다. 차원을 지정하지 않으면 다음 기본값이 사용됩니다.

- 계정 - [none]
- ICP - [ICP None]
- 사용자정의1...4 - [None]
- 시나리오 - 현재 시나리오 멤버
- 엔티티 - 현재 엔티티 멤버
- 값 - 현재 값 멤버
- 연도 및 기간 - 현재 멤버

`CellTextLabel`은 기본 셀 텍스트 레이블([Default])이거나 로드된 적합한 셀 텍스트 레이블 중 하나입니다.

 주:

기본 차원 멤버 값은 계산 규칙에서 이 함수를 사용할 때 적용됩니다. 할당, 환산 또는 통합 규칙에서 사용할 경우 계정, ICP 및 사용자정의 1...4 멤버를 지정해야 합니다. 시나리오, 연도, 기간, 뷰, 엔티티 및 값은 기본적으로 현재 멤버로 지정됩니다.

### 반환 값

지정된 POV 및 셀 텍스트 레이블의 셀 텍스트입니다.

## 예

```
HS.GetCellTextWithLabel("A#Sales.I#[ICP  
None].C1#[None].C2#[None].C3#[None].C4#[None]","Rating")
```

## GetCellType

셀 유형을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당



주:

계정 유형은 모두 대문자를 사용합니다.

## 구문

```
HS.GetCellType("POVExpression")
```

여기서 *POVExpression*은 적합한 POV입니다.

## 반환 값

지정된 셀의 유형입니다.

적합한 유형은 다음과 같습니다.

- ASSET
- LIABILITY
- REVENUE
- EXPENSE
- FLOW
- BALANCE
- BALANCERECURRING
- CURRENCYRATE
- GROUPLABEL

**예**

다음 예에서는 셀 유형이 EXPENSE인지 여부를 확인합니다. EXPENSE인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.GetCellType("S#Actual.C4#[None]") = "EXPENSE" Then
    ...
End If
```

## GetCustomLabelArray

사용자 지정 차원 레이블 목록을 배열로 반환합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.GetCustomLabelArray (Names, Aliases)
```

**반환 값**

사용자정의 약식 이름이 포함된 배열과 사용자정의 긴 이름이 포함된 배열 등 두 개의 배열을 반환합니다.

**예**

```
HS.GetCustomLabelArray (Custom1, Products)
```

## GetItem

통합을 위해 처리할 개별 레코드를 가져옵니다. 이 함수는 통합 규칙에 사용할 수 있습니다.

**주:**

이 함수는 4개 이하의 사용자정의 차원에만 사용할 수 있습니다. 5개 이상의 사용자정의 차원에는 [DataUnitItem](#)을 사용합니다.

## 구문

```
DataUnit.GetItem (lItem, strAccount, strICP, strCustom1, strCustom2,
strCustom3, strCustom4, dData)
```

**표 11-25** GetItem 함수 구문

매개변수	설명
<i>lItem</i>	레코드 번호입니다.
<i>strAccount</i>	적합한 계정 차원 멤버의 이름입니다.
<i>strICP</i>	적합한 ICP 차원 멤버의 이름입니다.
<i>strCustom1</i>	적합한 사용자정의1 차원 멤버의 이름입니다.
<i>strCustom2</i>	적합한 사용자정의2 차원 멤버의 이름입니다.
<i>strCustom3</i>	적합한 사용자정의3 차원 멤버의 이름입니다.
<i>strCustom4</i>	적합한 사용자정의4 차원 멤버의 이름입니다.
<i>dData</i>	지정된 셀의 데이터입니다.

## 반환 값

계정, ICP, 사용자정의1...4, 데이터를 포함하는 배열입니다.

## 예

```
Call DataUnit.GetItem(lItem, strAccount, strICP, strCustom1, strCustom2,
strCustom3, strCustom4, dData)
```

```
If dData = 0 Then
```

```
...
```

```
End If
```

## GetItemIDs2

차원 ID를 사용하여 통합을 위해 처리할 개별 레코드를 가져옵니다. 이 함수는 통합 규칙에 사용할 수 있습니다.

### 주:

이 함수는 4개 이하의 사용자정의 차원에만 사용할 수 있습니다. 5개 이상의 사용자정의 차원에는 [GetItemIDs2ExtDim](#)을 사용합니다.

## 구문

```
DataUnit.GetItemIDs2(lItem, lAccountID, lICPID, lCustom1ID,
lCustom2ID, lCustom3ID, lCustom4ID, dData)
```

**표 11-26** GetItemIDs2 함수 구문

매개변수	설명
<i>lItem</i>	레코드 번호입니다.
<i>lAccountID</i>	적합한 계정 차원 멤버의 ID 번호입니다.
<i>lICPID</i>	적합한 ICP 차원 멤버의 ID 번호입니다.
<i>lCustom1ID</i>	적합한 사용자정의1 차원 멤버의 ID 번호입니다.
<i>lCustom2ID</i>	적합한 사용자정의2 차원 멤버의 ID 번호입니다.
<i>lCustom3ID</i>	적합한 사용자정의3 차원 멤버의 ID 번호입니다.
<i>lCustom4ID</i>	적합한 사용자정의4 차원 멤버의 ID 번호입니다.
<i>dData</i>	지정된 셀의 데이터입니다.

## 반환 값

계정, ICP, 사용자정의1...4 데이터가 포함된 변수입니다.

## 예

```
Call DataUnit.GetItemIDs2(lItem, lAccount, lICP, lCustom1, lCustom2,
lCustom3, lCustom4, dData)
```

```
If dData = 0 Then
```

```
...
```

```
End If
```

## GetItemIDs2ExtDim

차원 ID를 사용하여 통합을 위해 처리할 개별 레코드를 가져옵니다. 이 함수는 통합 규칙에 사용할 수 있습니다.

### 주:

이 함수는 5개 이상의 사용자정의 차원이 있는 경우에 사용됩니다.



## 구문

```
DataUnit.GetItemIDs2ExtDim(lItem, lAccountID, lICPID, lCustomID, dData)
```

**표 11-27** GetItemIDs2 함수 구문

매개변수	설명
<i>lItem</i>	레코드 번호입니다.
<i>lAccountID</i>	적합한 계정 차원 멤버의 ID 번호입니다.
<i>lICPID</i>	적합한 ICP 차원 멤버의 ID 번호입니다.
<i>lCustomID</i>	적합한 사용자정의 차원 멤버의 ID 번호입니다.
<i>dData</i>	지정된 셀의 데이터입니다.

## 반환 값

차원과 멤버 둘 다의 ID가 포함된 2열 배열입니다.

## 예

```
Call DataUnit.GetItemIDs2(lItem, lAccount, lICP, lCustom5, dData)
```

```
If dData = 0 Then
```

```
...
```

```
End If
```

# GetNumItems

통합을 위해 처리할 레코드 수를 가져옵니다. 이 함수는 통합 규칙에 사용할 수 있습니다.

## 구문

```
NumItems = DataUnit.GetNumItems
```

## 반환 값

데이터 단위의 레코드 수입니다.

## 예

```

Set dataUnit = HS.OpenDataUnit("")

lNumItems = dataUnit.GetNumItems

for lItem = 0 to lNumItems - 1

' Get the next item from the Data Unit

Call dataUnit.GetItem(lItem, strAccount, strICP, strCustom1,
strCustom2, strCustom3, strCustom4, dData)

```

## GetNumLID

지정된 POV의 라인 항목 세부정보 수를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.GetNumLID("POVExpression")
```

여기서 *POVExpression*은 적합한 POV 조합입니다. 차원을 지정하지 않으면 다음 값이 사용됩니다.

- 계정 - [none]
- ICP - [ICP None]
- 사용자정의1...4 - [None]
- 시나리오 - 현재 시나리오 멤버
- 엔티티 - 현재 엔티티 멤버
- 값 - <entity currency>
- 연도 및 기간 - 현재 멤버

#### 주:

부적합한 교차가 지정된 경우 반환 값은 0입니다.

**반환 값**

지정된 셀에 대해 입력된 라인 항목 세부정보 수입니다.

**예**

다음 예에서는 지정된 셀에 대해 라인 항목 세부정보가 입력되지 않은 경우 If...Then 라인과 End If 라인 사이의 명령문이 실행됩니다.

```
If HS.GetNumLID("A#Sales.C1#Prod1.C2#Region1") = 0 then
```

```
...
```

```
End If
```

## GetRate

현재 POV나 지정된 POV의 환율을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.GetRate("POVExpression")
```

여기서 POVExpression은 적합한 POV입니다.



**주:**

환율 계정을 지정해야 합니다.

**반환 값**

지정된 POV의 환율입니다.

**예**

```
dVar1 = HS.GetRate("S#Actual.Y#2014.P#March.V#Euro.E#Connecticut.A#AvgRate")
```

## GetSubmissionGroup

셀의 프로세스 관리 제출 그룹을 가져옵니다.

### 구문

```
HS.GetSubmissionGroup("A#Account.C1#Custom1.C2#Custom2.C3#Custom3.C4#Custom4.I#ICP")
```

### 반환 값

프로세스 관리 제출 그룹을 나타내는 정수입니다. 적합한 값은 1-99입니다.

### 예

```
dVar1=HS.GetSubmissionGroup("A#Sales.C1#Golfballs.C2#Tennisballs.C3#Soccerballs.C4#Basketb
```

## GetSubmissionPhase

셀의 프로세스 관리 제출 단계를 가져옵니다.

### 구문

```
HS.GetSubmissionPhase("S#Scenario.P#Period.A#Account.C1#Custom1.C2#Custom2.C3#Custom3.C4#C
```

### 반환 값

프로세스 관리 제출 단계를 나타내는 정수입니다. 적합한 값은 1-9입니다.

### 예

```
dVar1=HS.GetSubmissionPhase("S#Actual.P#January.A#Sales.C1#Golfballs.C2#Tennisballs.C3#Soc
```

## Holding

현재 또는 지정된 엔티티나 상위 차원 멤버의 지주 회사를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.Entity.Holding("Entity")
```

```
HS.Entity.Holding("")
```

```
HS.Parent.Holding("Entity")
```

```
HS.Parent.Holding("")
```

여기서 *Entity*는 적합한 엔티티 차원 또는 상위 멤버의 이름입니다. 빈 문자열("")을 사용하면 현재 멤버에 이 함수를 적용할 수 있습니다.

## 반환 값

지정된 엔티티 멤버나 상위 멤버의 지주 회사 이름이 포함된 문자열입니다.

## 예

다음 예에서는 Europe이 엔티티 France의 지주 회사인 경우 If...Then 라인과 End If 라인 사이의 명령문이 실행됩니다.

```
If HS.Entity.Holding("France") = "Europe" Then
```

```
...
```

```
End If
```

## ICPTopMember

현재 계정 차원 멤버나 지정된 계정 멤버의 ICPTopMember를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.Account.ICPTopMember("AccountName")
```

```
HS.Account.ICPTopMember("")
```

```
HS.Account.ICPTopMember(Var1)
```

### 주:

빈 문자열("")을 사용하면 Sub Consolidate 서브루틴에 함수를 사용하는 경우에만 현재 계정에 이 함수를 적용할 수 있습니다.

**표 11-28 ICPTopMember 함수 구문**

매개변수	설명
<i>AccountName</i>	적합한 계정 차원 멤버의 이름입니다.
<i>Var1</i>	VisualBasic 변수입니다.

### 반환 값

계정의 ICPTopMember 이름이 포함된 문자열입니다.

### 예

다음 예에서는 Sales 계정의 ICPTopMember가 TotalProd인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Account.ICPTopMember("Sales") = "TotalProd" Then
```

```
...
```

```
End If
```

## ICPWeight

애플리케이션의 ICP 가중치를 가져옵니다. 값 차원의 [ICP Top] 멤버에 합산되는 내부거래 파트너 엔티티 [ICP Entities] 금액의 비율입니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합

- 할당

### 구문

```
HS.AppSettings.ICPWeight
```

### 반환 값

ICPTopMember에 합산되는 ICP 엔티티의 백분율입니다. 값은 백으로 스케일링된 비율로, 1.0은 100%입니다.

### 예

다음 예에서는 현재 애플리케이션의 ICPWeight가 1인 경우 If...Then 라인과 End If 라인 사이의 명령문이 실행됩니다.

```
If HS.AppSettings.ICPWeight = 1 Then
```

```
...
```

```
End If
```

## IDFromMember

지정된 멤버의 ID 번호를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.<Object>.IDFromMember("Element")
```

표 11-29 IDFromMember 함수 구문

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의 (사용자정의 차원 레이블)</li> <li>엔티티</li> <li>ICP</li> <li>상위</li> <li>기간</li> <li>시나리오</li> <li>값</li> <li>연도</li> <li>뷰</li> </ul>
Element	선택한 객체에 따라 다음 차원 중 하나에서 적합한 멤버의 이름입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의</li> <li>엔티티</li> <li>ICP</li> <li>상위</li> <li>기간</li> <li>시나리오</li> <li>값</li> <li>연도</li> </ul>

**반환 값**

지정된 멤버의 ID 번호입니다. 멤버 입력이 적합한 멤버가 아니면 -1이 반환됩니다.

**예**

다음 예에서는 Connecticut의 ID 번호를 가져옵니다.

```
1EntityID = HS.Entity.IDFromMember("Connecticut")
```

```
1CustomID=HS.Custom("Prod").IDFromMember("P3000-Phones")
```

## ImpactStatus

지정된 데이터 단위의 상태를 영향 받음으로 변경합니다. 이 함수는 계산 규칙에 사용할 수 있습니다.



## 구문

시나리오, 연도, 기간, 엔티티 및 값 멤버의 조합입니다. 시나리오가 동일한 경우 연도 및 기간 조합이 미래 기간이어야 합니다. 값 멤버를 지정하지 않으면 현재로 간주됩니다.

```
HS.ImpactStatus"S#Scenario.Y#Year.P#Period.E#Entity.V#Value"
```

**표 11-30 ImpactStatus 함수 구문**

매개변수	설명
<i>Scenario</i>	적합한 시나리오 차원 멤버의 이름입니다.
<i>Year</i>	적합한 연도입니다.
<i>Period</i>	적합한 기간입니다.
<i>Entity</i>	적합한 엔티티 차원 멤버의 이름입니다.
<i>Value</i>	적합한 값 차원 멤버의 이름입니다.

### 주:

지정한 POV가 처리되는 데이터 단위와 동일한 시나리오, 연도, 기간 및 엔티티인 경우 오류가 발생하고 데이터 단위에 영향을 주지 않습니다. 규칙을 실행할 때 대상 기간이 잠겨 있으면 VBScript 오류가 반환되고 데이터 단위에 영향을 주지 않습니다.

## 반환 값

없습니다.

## 예

```
HS.ImpactStatus "S#Actual.Y#2014.P#January"
```

# Input

기본 엔티티 레벨에서 입력할 수 있는 데이터 포인트에 대한 상위 엔티티의 <Entity Currency>에 데이터를 입력할 수 있습니다. IsCalculated 플래그가 지정된 데이터 포인트와 NoInput으로 설정된 데이터 포인트는 여기에 포함되지 않습니다. 엔티티 통화 값 차원만 지원됩니다. 이 함수를 사용하는 경우 상위 엔티티 레벨의 값이 1차 하위 구성요소의 기여금액 합계와 같지 않습니다.

이 함수를 사용하면 상위 엔티티 레벨에 입력할 수 있으므로 1차 하위 구성요소 기여금액이 상위 엔티티의 엔티티 통화 값 차원에 롤업되지 않습니다. 하지만 상위-1차 하위 구성요소 기여금액 값이 저장되며 기여금액 조정 값 차원에 분개를 게시할 수 있습니다. 이 함수는 입력 규칙에 사용할 수 있습니다.

**▲ 주의:**

상위 엔티티 레벨에 입력할 수 있는 계정의 경우 상위 엔티티의 엔티티 통화 값 멤버 값이 모든 1차 하위 구성요소 기여금액 합계와 같지 않다는 것을 기억해야 합니다.

**구문**

```
HS.Input"POVExpression"
```

여기서 *POVExpression*은 POV입니다.

**반환 값**

없습니다.

**예**

다음 예에서는 Sales 계정과 예산 시나리오가 교차되는 셀에 입력할 수 있도록 합니다.

```
Sub Input
```

```
    HS.Input"A#Sales.S#Budget"
```

```
End Sub
```

## IsAlmostEqual

사전 정의된 Oracle Hyperion Financial Management 값을 기준으로 전달된 값이 같은지 여부를 확인합니다. 이 함수는 모든 유형의 규칙에 사용할 수 있습니다.

-0.0000000000001에서 0.0000000000001 사이의 차이는 차이가 0인 것으로 간주됩니다.

**구문**

```
BooleanValue = HS.IsAlmostEqual(Value1, Value2)
```

**반환 값**

전달된 값이 같으면 True이고, 같지 않으면 False인 부울 표현식입니다.

**예**

```
Dim BoolVal

Dim Value1

Dim Value2

Value1 = 10.1299999999

Value2 = 10.13

BoolVal = HS.IsAlmostEqual(Value1, Value2)

If BoolVal = true Then

    'do processing

Else

    'do Processing

End If
```

## IsBase

현재 멤버나 지정된 멤버가 애플리케이션이나 지정된 상위의 기본 멤버인지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

 **주:**

멤버는 1차 하위 구성요소가 없는 경우 기본 멤버입니다. 즉 트리 계층에서 분기의 끝에 있습니다.

## 구문

```
HS.<Object>.IsBase("Parent","Element")
```

```
HS.Node.IsBase("Parent"."Entity"."S#Scenario.Y#Year.P#Period")
```

```
HS.<Object>.IsBase("", "")
```

```
HS.Custom(Dimension).IsBase(Member)
```



### 주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-31 IsBase 함수 구문**

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의 (사용자정의 차원 레이블)</li> <li>엔티티</li> <li>상위</li> </ul>
Parent	적합한 상위 멤버입니다. Parent는 노드와 함께 사용하는 경우에만 필수입니다.
Element	선택한 객체에 따라 다음 차원 중 하나에서 적합한 멤버의 이름입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의</li> <li>엔티티</li> <li>상위</li> </ul>
Entity	적합한 엔티티 차원 멤버의 이름입니다.
Scenario	적합한 시나리오 차원 멤버의 이름입니다.
Year	적합한 연도입니다.
Period	적합한 기간입니다.



### 주:

노드의 경우 요소가 지정된 상위의 활성화 기본 멤버인지 여부를 확인합니다.

## 반환 값

요소가 지정된 상위 아래의 기본 멤버이거나 상위가 지정되지 않은 경우 애플리케이션의 기본 멤버이면 True이고, 요소가 기본 멤버가 아니면 False인 부울 표현식입니다. 요소가 기본 멤버가 아니면 False인 부울 표현식입니다.

노드의 경우 요소가 지정된 POV에서 상위 아래의 활성 기본 엔티티이면 True이고, 요소가 활성 기본 엔티티가 아니면 False입니다.

## 예

다음 예에서는 Connecticut이 EastRegion 아래의 기본 엔티티인 경우 If...Then 라인과 End If 라인 사이의 명령문이 실행됩니다.

```
If HS.Entity.IsBase("EastRegion","Connecticut") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Prod").IsBase("P3000-Phones") = TRUE Then
```

```
...
```

```
End If
```

## IsCalculated

현재 계정 차원 멤버나 지정된 계정 멤버가 계산된 계정인지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.Account.IsCalculated("Account")
```

여기서 *Account*는 적합한 계정 멤버의 이름입니다.

```
HS.Account.IsConsolidated("")
```

 주:

빈 문자열("")을 사용하면 Sub Consolidate 서브루틴에 함수를 사용하는 경우에만 현재 멤버에 이 함수를 적용할 수 있습니다.

**반환 값**

계정이 계산된 계정이면 True이고, 계정이 계산된 계정이 아니면 False인 부울 표현식입니다.

**예**

다음 예에서는 Sales 계정이 계산되는 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Account.IsCalculated("Sales") = TRUE Then
```

```
...
```

```
End If
```

## IsChild

현재 멤버나 지정된 멤버가 지정된 상위의 1차 하위 구성요소인지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

 주:

트리 계층에서 멤버의 한 레벨 아래에 있는 경우 멤버는 1차 하위 구성요소입니다.

## 구문

```
HS.<Object>.IsChild("Parent","Element")
```

```
HS.Node.IsChild("Parent"."Entity"."S#Scenario.Y#Year.P#Period")
```

```
HS.<Object>.IsChild("Parent","")
```

```
HS.Custom(Dimension).IsChild("P3000-Phones")
```



### 주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

표 11-32 IsChild 함수 구문

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의 (사용자정의 차원 레이블)</li> <li>엔티티</li> <li>상위</li> </ul>
Parent	상위 멤버의 이름입니다. Parent는 필수입니다.
Element	선택한 객체에 따라 다음 차원 중 하나에서 적합한 멤버의 이름입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의</li> <li>엔티티</li> <li>상위</li> </ul>
Entity	적합한 엔티티 차원 멤버의 이름입니다.
Scenario	적합한 시나리오 차원 멤버의 이름입니다.
Year	적합한 연도입니다.
Period	적합한 기간입니다.



### 주:

노드의 경우 멤버가 지정된 상위의 활성 1차 하위 구성요소인지 여부를 확인합니다.

## 반환 값

요소가 지정된 상위의 1차 하위 구성요소이면 True이고, 요소가 지정된 상위의 1차 하위 구성요소가 아니면 False인 부울 표현식입니다.

노드의 경우 요소가 지정된 상위의 활성 1차 하위 구성요소이면 True이고, 요소가 지정된 상위의 활성 1차 하위 구성요소가 아니면 False입니다.

## 예

다음 예에서는 Connecticut이 EastRegion의 1차 하위 구성요소인 경우 If...Then 라인과 End If 라인 사이의 명령문이 실행됩니다.

```
If HS.Entity.IsChild("EastRegion","Connecticut") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Prod").IsChild("P3000-Phones") = TRUE Then
```

```
...
```

```
End If
```

## IsConsolidated

현재 계정 차원 멤버나 지정된 계정 멤버가 통합된 계정인지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.Account.IsConsolidated("Account")
```

여기서 *Account*는 적합한 계정 멤버의 이름입니다.

```
HS.Account.IsConsolidated("")
```



 주:

빈 문자열("")을 사용하면 Sub Consolidate 서브루틴에 함수를 사용하는 경우에만 현재 멤버에 이 함수를 적용할 수 있습니다.

**반환 값**

계정이 상위 계정에 통합되면 True이고, 계정이 상위 계정에 통합되지 않으면 False인 부울 표현식입니다.

**예**

다음 예에서는 Sales 계정이 통합되는 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Account.IsConsolidated("Sales") = TRUE Then
```

```
...
```

```
End If
```

## IsDescendant

현재 멤버나 지정된 멤버가 지정된 상위의 하위 멤버인지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

 주:

트리 계층에서 상위보다 한 레벨 아래에 있는 경우 멤버는 하위 멤버입니다. 하위 멤버는 동일한 트리 분기 내에 있습니다.

예를 들어 이 계층에서 FosterCity와 Sunnyvale은 California와 UnitedStates의 하위 멤버입니다.



## 구문

```
HS.<Object>.IsDescendant ("Parent", "Element")
```

```
HS.Node.IsDescendant ("Parent"."Entity", "S#Scenario.Y#Year.P#Period")
```

```
HS.<Object>.IsDescendant ("Parent", "")
```

```
HS.Custom(Dimension).IsDescendant(Member)
```



주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-33 IsDescendant 함수 구문**

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의 (사용자정의 차원 레이블)</li> <li>엔티티</li> <li>상위</li> </ul>
Parent	적합한 상위 멤버의 이름입니다. Parent는 필수입니다.
Element	선택한 객체에 따라 다음 차원 중 하나에서 적합한 멤버의 이름입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의</li> <li>엔티티</li> <li>상위</li> </ul>
Entity	적합한 엔티티 차원 멤버의 이름입니다.
Scenario	적합한 시나리오 차원 멤버의 이름입니다.
Year	적합한 연도입니다.
Period	적합한 기간입니다.



주:

노드를 객체로 사용하는 경우 함수는 멤버가 지정된 상위의 활성 하위 멤버인지 여부를 확인합니다.

## 반환 값

요소가 지정된 상위의 하위 멤버이면 True이고, 요소가 지정된 상위의 하위 멤버가 아니면 False인 부울 표현식입니다.

노드의 경우 요소가 지정된 상위의 활성 하위 멤버이면 True이고, 요소가 지정된 상위의 활성 하위 멤버가 아니면 False입니다.

## 예

다음 예에서는 Connecticut이 Regional의 하위 멤버인 경우 If...Then 라인과 End If 라인 사이의 명령문이 실행됩니다.

```
If HS.Entity.IsDescendant("Regional","Connecticut") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Prod").IsDescendant("All_Phones", P3000_Phones")
```

```
...
```

```
End If
```

## IsFirst

현재 기간이나 연도가 애플리케이션의 첫 번째 기간이나 연도인지 여부를 확인합니다. 현재 시나리오의 기본 빈도를 사용하여 현재 기간이나 연도가 애플리케이션의 첫 번째 기간이나 연도인지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.<Object>.IsFirst
```

여기서 <Object>는 다음 키워드 중 하나입니다.

- 기간
- 연도

## 반환 값

현재 기간이나 연도가 첫 번째 기간이나 연도이면 True이고, 현재 기간이나 연도가 첫 번째 기간이나 연도가 아니면 False인 부울 표현식입니다.

## 예

다음 예에서는 현재 기간이 첫 번째 기간인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Period.IsFirst = TRUE Then
```

```
...
```

```
End If
```

## IsICP

현재 계정 또는 엔티티 차원 멤버나 지정된 계정 또는 엔티티 멤버가 내부거래 파트너(ICP) 인지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.<Object>.IsICP("Element")
```

```
HS.<Object>.IsICP("")
```



### 주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-34 IsICP 함수 구문**

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>• 계정</li> <li>• 엔티티</li> </ul>

표 11-34 (계속) IsICP 함수 구문

매개변수	설명
<i>Element</i>	선택한 객체에 따라 다음 차원 중 하나에서 적합한 멤버의 이름입니다. <ul style="list-style-type: none"> <li>계정</li> <li>엔티티</li> </ul>

**반환 값**

계정이나 엔티티 멤버가 내부거래 파트너이면 True이고, 계정이나 엔티티 멤버가 내부거래 파트너가 아니면 False인 부울 표현식입니다.

**예**

다음 예에서는 Sales 계정이 내부거래 파트너인 경우 If...Then 라인과 End If 라인 사이의 명령문이 실행됩니다.

```
If HS.Account.IsICP("Sales") = TRUE Then
```

```
...
```

```
End If
```

## IsLast

현재 기간이나 연도가 애플리케이션의 마지막 기간이나 연도인지 여부를 확인합니다. 현재 시나리오의 기본 빈도를 사용하여 현재 기간이나 연도가 애플리케이션의 마지막 기간이나 연도인지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.<Object>.IsLast
```

여기서 <Object>는 다음 키워드 중 하나입니다.

- 기간
- 연도

**반환 값**

현재 기간이나 연도가 마지막 기간이나 연도이면 True이고, 현재 기간이나 연도가 마지막 기간이나 연도가 아니면 False인 부울 표현식입니다.

## 예

다음 예에서는 현재 기간이 마지막 기간인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Period.IsLast = TRUE Then  
  
    ...  
  
End If
```

## IsTransCur

현재 값 차원 멤버가 환산된 통화 멤버인지 여부를 확인합니다. 이 함수는 계산 규칙에 사용할 수 있습니다.

### 구문

```
HS.Value.IsTransCur
```

### 반환 값

현재 값 멤버가 환산된 통화 멤버이면 True이고, 현재 값 멤버가 환산된 통화 멤버가 아니면 False인 부울 표현식입니다.

## 예

다음 예에서는 값 멤버가 환산된 통화 멤버인 경우 If...Then 문과 End If 문 사이의 모든 명령문이 실행됩니다.

```
If HS.Value.IsTransCur = TRUE Then  
  
    ...  
  
End If
```

## IsTransCurAdj

현재 값 차원 멤버가 환산된 통화 조정 멤버인지 여부를 확인합니다. 이 함수는 계산 규칙에 사용할 수 있습니다.

### 구문

```
HS.Value.IsTransCurAdj
```

### 반환 값

현재 값 멤버가 환산된 통화 조정 멤버이면 True이고, 현재 값 멤버가 환산된 통화 조정 멤버가 아니면 False인 부울입니다.

### 예

다음 예에서는 값 멤버가 환산된 통화 조정 멤버인 경우 If...Then 문과 End If 문 사이의 모든 명령문이 실행됩니다.

```
If HS.Value.IsTransCurAdj = TRUE Then
```

```
...
```

```
End If
```

## IsValidDest

지정된 POV가 적합한 대상인지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 주:

이 함수는 셀이 계산된 셀인지 여부를 확인하지 않습니다.

### 구문

```
HS.IsValidDest("POVExpression")
```

여기서 *POVExpression*은 POV입니다. 차원을 지정하지 않으면 다음 기본값이 사용됩니다.

- 계정 - Sub Consolidate 서브루틴에 사용되는 경우 현재 계정 멤버. 그렇지 않으면 계정이 필요합니다.
- 사용자정의 및 ICP - Sub Consolidate 서브루틴에 사용되는 경우 현재 멤버. 그렇지 않으면 계정의 TopMember가 사용됩니다.
- 시나리오 - 현재 시나리오 멤버
- 엔티티 - 현재 엔티티 멤버
- 값 - 현재 값 멤버
- 연도 및 기간 - 현재 멤버

## 반환 값

지정된 POV가 적합한 대상이면 True이고, 그렇지 않으면 False인 부울입니다.

## 예

다음 예에서는 지정된 대상이 적합한 경우 If...Then 문과 End If 문 사이의 모든 명령문이 실행됩니다.

```
If HS.IsValidDest("A#Sales.I#CT.C1#P1.C2#R1.C3#[None].C4#[None]") =  
TRUE Then
```

```
...
```

```
End If
```

## IsZero

사전 정의된 Oracle Hyperion Financial Management 값을 기준으로 전달된 값이 0에 가까운지 여부를 확인합니다. 이 함수는 모든 유형의 규칙에 사용할 수 있습니다.

부동 소수점 산술 시 무시할 수 있는  $1 \times 10 \text{ E-}10$  미만 오류가 발생하는 0과 정확한 비교 대신 이 함수를 사용하는 것이 좋습니다.

대체:

```
Difference = Value1 - Value2
```

```
If Difference = 0 Then
```

```
    'process where Difference = 0
```

```
Else
```

```
    'process where Difference <> 0
```

```
End If
```



사용:

```
Difference = Value1 - Value2
```

```
If HS.IsZero(Difference) Then
```

```
    'process where Difference = 0
```

```
Else
```

```
    'process where Difference <> 0
```

```
End If
```

**구문**

```
BooleanValue = HS.IsZero(Value)
```

**반환 값**

전달된 값이 0에 가까우면 True이고, 그렇지 않으면 False인 부울입니다.

**예**

```
Dim BoolVal

Dim Value

Value = 0.000000001

BoolVal = HS.IsZero(Value)

If BoolVal = true Then

    'do processing

Else

    'do Processing

End If
```

## List

지정된 목록에 포함된 요소를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.<Object>.List("Parent","Listname")
```

```
HS.Node.List("Parent","Listname"."S#Scenario.Y#Year.P#Period")
```

```
HS.Custom("Dimension").List("Parent","Listname")
```

표 11-35 List 함수 구문

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의 (사용자정의 차원 레이블)</li> <li>엔티티</li> <li>상위</li> <li>ICP</li> <li>시나리오</li> </ul>
Parent	적합한 상위 멤버의 이름입니다.
Listname	적합한 시스템 목록이나 사용자 정의 목록의 이름입니다. 노드의 경우 적합한 엔티티 시스템 목록이어야 합니다.
Scenario	적합한 시나리오 차원 멤버의 이름입니다.
Year	적합한 연도입니다.
Period	적합한 기간입니다.

**반환 값**

지정된 목록의 모든 요소가 포함된 배열입니다. 노드의 경우 목록의 활성 요소만 반환됩니다.

**예**

다음 예에서는 현재 계정의 사용자 정의 목록인 MyBaseList의 요소를 가져옵니다.

```
HS.Account.List("", "MyBaseList")
```

다음 예에서는 TotalAssets 계정의 시스템 목록인 [Base]의 요소를 가져옵니다.

```
HS.Account.List("TotalAssets", "[Base]")
```

```
Hs.Custom("Prod").List("Products", "[Base]")
```

## Member

현재 멤버의 이름을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.<Object>.Member
```

여기서 <Object>는 다음 키워드 중 하나입니다.

- 엔티티
- 상위
- 기간
- 시나리오
- 값
- 연도
- 뷰

## 반환 값

현재 멤버 이름이 포함된 문자열입니다.

Value 객체의 경우 Member에서 값과 연결된 통화가 아니라 현재 값 멤버의 이름이 반환됩니다. 예를 들어 현재 값 멤버가 Entity Currency이고 값이 USD 통화와 연계된 경우 HS.Parent.Member에서 USD가 아니라 Entity Currency가 반환됩니다.



### 팁:

현재 값 멤버의 통화를 가져오려면 DefCurrency 함수를 사용하십시오.

## 예

다음 예에서는 현재 엔티티가 California인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Entity.Member = "California" Then
```

```
...
```

```
End If
```

## MemberFromID

지정된 ID 번호의 차원 멤버를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합

- 할당

## 구문

```
HS.<Object>.MemberFromID(ElementID)
```

```
HS.Custom(Dimension).MemberFromID(ElementID)
```

**표 11-36 MemberFromID 함수 구문**

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>• 계정</li> <li>• Custom1...4</li> <li>• 사용자정의 (<i>사용자정의 차원 레이블</i>)</li> <li>• 엔티티</li> <li>• ICP</li> <li>• 상위</li> <li>• 기간</li> <li>• 시나리오</li> <li>• 값</li> <li>• 연도</li> <li>• 뷰</li> </ul>
<i>Parent</i>	적합한 상위 멤버입니다.
<i>ElementID</i>	선택한 객체에 따라 다음 차원 중 하나에서 적합한 멤버의 ID 번호입니다. <ul style="list-style-type: none"> <li>• 계정</li> <li>• Custom1...4</li> <li>• 사용자정의</li> <li>• 엔티티</li> <li>• ICP</li> <li>• 상위</li> <li>• 기간</li> <li>• 시나리오</li> <li>• 값</li> <li>• 연도</li> </ul>

## 반환 값

차원 멤버 이름입니다.

## 예

다음 예에서는 ID 번호 001의 멤버를 가져옵니다.

```
strEntity = HS.Entity.MemberFromID(001)
```

```
strCustom = HS.Custom("Prod").MemberFromID(001)
```

## Method

지정된 멤버의 통합 메소드를 가져옵니다. 0이 아닌 값이 여러 개 있으면 찾은 첫 번째 값이 반환됩니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.Node.Method("POVExpression")
```

여기서 *POVExpression*은 시나리오, 연도, 기간 및 엔티티 멤버의 조합입니다. 연도, 기간 및 엔티티 멤버의 조합입니다.

### 반환 값

지정된 POV의 통합 메소드를 지정하는 문자열입니다.

### 예

다음 예에서는 POV의 메소드가 GLOBAL인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Node.Method("S#Actual.Y#2014.P#January.E#Regional.Connecticut")
= "GLOBAL" Then
```

```
...
```

```
End If
```

## NoInput

사용자가 특정 셀이나 셀 슬라이스에 데이터를 입력할 수 없도록 합니다. 이 함수는 입력 또는 계산되지 않는 계정이 있는 경우에 유용합니다.

계산된 계정을 설정하면 실제로 사용자가 해당 계정에 입력할 수 없게 됩니다. 하지만 일부 차원 교차에는 데이터를 입력할 수 있고 다른 차원에는 입력할 수 없는 계정이 있는 경우 NoInput을 사용할 수 있습니다. 이 함수는 NoInput 규칙에 사용할 수 있습니다.

### 구문

```
HS.NoInput "POVExpression"
```

여기서 *POVExpression*은 POV입니다.

반환 값  
없습니다.

예

다음 예에서는 Sales 계정과 2014년 예산 시나리오를 교차하는 셀에 입력할 수 없도록 합니다.

```
Sub NoInput
```

```
    HS.NoInput "S#Budget.Y#2014.A#Sales"
```

```
End Sub
```

## NoRound

이후의 모든 `Exp` 문에 대해 반올림을 해제합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 할당

### 팁:

Round 함수의 인수로 0을 입력하여 반올림을 해제할 수도 있습니다. 예를 들어 `HS.Round(0)`은 반올림을 해제합니다.

구문

```
HS.NoRound
```

반환 값  
없습니다.

## 예

다음 예에서는 SalesRound 계정의 셀에 삽입된 금액을 가장 가까운 소수 첫째 자리로 반올림한 다음 NoRound를 사용하여 SalesNoRound 계정의 셀에 삽입된 금액에 대해 반올림을 해제합니다.

```
HS.Round 0.1
```

```
HS.Exp "A#SalesRound" = "A#Sales"
```

```
HS.NoRound
```

```
HS.Exp "A#SalesNoRound" = "A#Sales"
```

## NumBase

현재 멤버나 지정된 멤버의 기본 멤버 수를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 주:

멤버는 1차 하위 구성요소가 없는 경우 기본 멤버입니다. 즉 트리 계층에서 분기의 끝에 있습니다.

## 구문

```
HS.<Object>.NumBase("Element")
```

```
HS.Node.NumBase("S#Scenario.Y#Year.P#Period.E#Entity")
```

```
HS.<Object>.NumBase("")
```

```
HS.Custom(Dimension).NumBase(Member)
```





주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

표 11-37 NumBase 함수 구문

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의 (사용자정의 차원 레이블)</li> <li>엔티티</li> <li>상위</li> <li>노드</li> </ul>
Element	선택한 객체에 따라 다음 차원 중 하나에서 적합한 멤버의 이름입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의</li> <li>엔티티</li> <li>상위</li> <li>노드</li> </ul> 계정 및 사용자정의 객체의 경우 멤버를 지정해야 합니다. 빈 문자열은 사용할 수 없습니다. 전체 차원의 기본 멤버 수를 가져오려면 다음 예와 같이 ALL을 따옴표로 묶어 지정합니다. iAcctBase = HS.Account.NumBase("ALL")
Scenario	적합한 시나리오 차원 멤버의 이름입니다.
Year	적합한 연도입니다.
Period	적합한 기간입니다.
Entity	적합한 엔티티 차원 멤버의 이름입니다.

Exp 함수에 NumBase 함수를 포함할 수도 있습니다. NumBase 함수를 포함하는 경우 다음 예와 같이 NumBase의 인수를 따옴표로 묶지 마십시오.

```
HS.Exp "A#AverageSales = A#Sales/HS.Entity.NumBase(Regional) "
```

#### 반환 값

기본 멤버 수를 식별하는 Long입니다. 노드의 경우 지정된 멤버의 활성 기본 요소 수를 가져옵니다.



주:

기본 엔티티가 분기에 두 번 표시되면 엔티티가 두 번 계산됩니다.

## 예

다음 예에서는 **Regional** 엔티티 아래에 있는 기본 엔티티의 평균 판매 금액을 저장하는 **SalesAlloc**라는 계정이 애플리케이션에 포함되어 있습니다. **SalesAlloc** 금액을 계산하기 위해 이 예에서는 **Sales** 계정의 금액을 **Regional** 아래의 기본 엔티티 수로 나눕니다.

```
If HS.Exp"A#SalesAlloc = A#Sales/HS.Entity.NumBase(Regional)" then
```

```
...
```

```
End If
```

```
If HS.Exp"A#SalesAlloc = A#Sales/  
HS.Custom("Prod").NumBase("TotalProducts")
```

```
...
```

```
End If
```

## Number

현재 기간 번호를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.View.PeriodNumber
```

### 반환 값

현재 기간 번호입니다.

## 예

다음 예에서는 현재 기간이 첫 번째 기간인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.View.PeriodNumber = 1 Then
```

```
...
```

```
End If
```

## NumChild

현재 차원 멤버나 지정된 멤버의 1차 하위 구성요소 수를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 주:

트리 계층에서 멤버의 한 레벨 아래에 있는 경우 멤버는 1차 하위 구성요소입니다. 지정된 객체보다 한 레벨 아래의 멤버만 계산됩니다.

## 구문

```
HS.<Object>.NumChild("Element")
```

```
HS.Node.NumChild("S#Scenario.Y#Year.P#Period.E#Entity")
```

```
HS.<Object>.NumChild("")
```

```
HS.Custom(Dimension).NumChild(Member)
```

### 주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

표 11-38 NumChild 함수 구문

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>계정</li> <li>사용자정의1-4</li> <li>사용자정의 (사용자정의 차원 레이블)</li> <li>엔티티</li> <li>노드</li> <li>상위</li> </ul>
Element	선택한 객체에 따라 다음 차원 중 하나에서 적합한 멤버의 이름입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의</li> <li>엔티티</li> <li>노드</li> <li>상위</li> </ul> 계정 및 사용자정의 객체의 경우 멤버를 지정해야 합니다. 빈 문자열은 사용할 수 없습니다. 전체 차원의 1차 하위 구성요소 수를 가져오려면 다음 예와 같이 ALL을 따옴표로 묶어 지정합니다. <pre>iAcctBase = HS.Account.NumChild("ALL")</pre>
Scenario	적합한 시나리오 차원 멤버의 이름입니다.
Year	적합한 연도입니다.
Period	적합한 기간입니다.
Entity	적합한 엔티티 차원 멤버의 이름입니다.

Exp 함수에 NumChild 함수를 포함할 수도 있습니다. NumChild 함수를 포함하는 경우 NumChild의 인수를 따옴표로 묶지 마십시오.

### 반환 값

1차 하위 멤버 수를 식별하는 문자열입니다. 노드의 경우 지정된 멤버의 활성 1차 하위 구성요소 수를 가져옵니다.

### 예

다음 예에서는 Regional 엔티티 바로 아래에 있는 엔티티의 평균 판매 금액을 저장하는 SalesChild라는 계정이 애플리케이션에 포함되어 있습니다. SalesChild 금액을 계산하기 위해 이 예에서는 Sales 계정의 금액을 Regional 바로 아래의 1차 하위 구성요소 수로 나눕니다.

```
HS.Exp "A#SalesChild = A#Sales/HS.Entity.NumChild(Regional)"
```

## NumCustom

애플리케이션에 대해 정의된 사용자정의 차원의 총수를 반환합니다. 예를 들어 사용자정의 차원을 5개 생성하고, 사용자정의 차원을 하나 삭제한 다음 사용자정의 차원을 2개 추가하는 경우 이 함수에 대해 반환되는 값은 6이어야 합니다. 이 값은 애플리케이션에 대해 정의된 사용자정의 차원의 총수입니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.NumCustom
```

### 반환 값

애플리케이션에 대한 사용자정의 차원 수입니다.

### 예

```
nCustoms = HS.NumCustom
```

## NumDescendant

현재 차원 멤버나 지정된 멤버의 하위 멤버 수를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 주:

트리 계층에서 상위보다 한 레벨 아래에 있는 경우 멤버는 하위 멤버입니다. 하위 멤버는 동일한 트리 분기 내에 있습니다.

예를 들어 이 계층에서 FosterCity와 Sunnyvale은 California와 UnitedStates의 하위 멤버입니다.

```

UnitedStates
├── California
│   ├── Sunnyvale
│   └── FosterCity

```

## 구문

```
HS.<Object>.NumDescendant ("Element")
```

```
HS.Node.NumDescendant ("S#Scenario.Y#Year.P#Period.E#Entity")
```

```
HS.<Object>.NumDescendant ("")
```

```
HS.Custom (Dimension) .NumDescendant (Member)
```



### 주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용합니다.

**표 11-39 NumDescendant 함수 구문**

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의 (사용자정의 차원 레이블)</li> <li>엔티티</li> <li>노드</li> <li>상위</li> </ul>
Element	선택한 객체에 따라 다음 차원 중 하나에서 적합한 멤버의 이름입니다. <ul style="list-style-type: none"> <li>계정</li> <li>Custom1...4</li> <li>사용자정의</li> <li>엔티티</li> <li>노드</li> <li>상위</li> </ul> 계정 및 사용자정의 객체의 경우 멤버를 지정해야 합니다. 빈 문자열은 사용할 수 없습니다. 전체 차원의 하위 멤버 수를 가져오려면 다음 예와 같이 ALL을 따옴표로 묶어 인수로 지정합니다. iAcctBase = HS.Account.NumDescendant ("ALL")
Scenario	적합한 시나리오 차원 멤버의 이름입니다.
Year	적합한 연도입니다.
Period	적합한 기간입니다.
Entity	적합한 엔티티 차원 멤버의 이름입니다.

**반환 값**

지정된 멤버의 하위 멤버 수입니다. 노드의 경우 지정된 멤버 아래의 활성 하위 엔티티 수입니다.

**주:**

하위 엔티티가 분기에 두 번 표시되면 엔티티가 두 번 계산됩니다.

**예**

다음 예에서는 France 엔티티에 하위 멤버가 없는 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Entity.NumDescendant("France") = 0 Then
```

```
...
```

```
End If
```

```
If HS.Custom("Product").NumDescendant("P3000-Phones") = 0 Then
```

```
...
```

```
End If
```

## NumPerInGen

처리되는 현재 기간에 대한 세대의 기간 수를 가져옵니다. 이 함수는 동적 SUB 함수에 사용할 수 있습니다.

**구문**

```
HS.Period.NumPerInGen
```

**반환 값**

뷰의 기간 수를 나타내는 하나의 값입니다.

**예**

```
HS.Dynamic "A#MarginPct=A#GrossMargin/HS.Period.NumPerInGen"
```

예를 들어 현재 기간이 4월이고 4월이 달력 파일(월별 세대)의 네 번째 세대에 있는 경우 월별 세대의 기간 수는 12입니다. 현재 기간이 Q2이고 달력 파일(분기별 세대)의 세 번째 세대에 있는 경우 기간 수는 4입니다.

월별 세대(네 번째 세대):

1월, 2월, 3월, 4월, 5월, 6월, 7월, 8월, 9월, 10월, 11월, 12월

이 세대의 기간 수로 12가 반환됩니다.

분기별 세대(세 번째 세대):

Q1, Q2, Q3, Q4

이 세대의 기간 수로 4가 반환됩니다.

반기별 세대(두 번째 세대):

HY1, HY2

이 세대의 기간 수로 2가 반환됩니다.

연도별 세대(첫 번째 세대):

연도

이 세대의 기간 수로 1이 반환됩니다.

## NumPeriods

지정된 시나리오의 빈도에 대해 정의된 기간 수를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.Scenario.NumPeriods("ScenarioName")
```

또는

```
HS.Scenario.NumPeriods(" ")
```

또는

```
HS.Scenario.NumPeriods(Var1)
```

### 반환 값

빈도의 기간 수를 나타내는 숫자 값입니다. 예를 들어 시나리오가 매월이면 기간 수로 12가 반환됩니다. 시나리오가 매분기이면 기간 수로 4가 반환됩니다.



**예**

다음 예에서는 실제 시나리오의 빈도에 대해 정의된 기간 수가 반환됩니다.

```
HS.Scenario.NumPeriods("Actual")
```

## OpenCellTextUnit

여러 셀에 대한 셀 텍스트를 검색합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합

전체 POV 대신, 셀 텍스트를 검색할 POV 부분(예: 시나리오 및 연도)을 지정할 수 있습니다. 페이지 차원 멤버(시나리오, 연도, 기간, 값, 엔티티)를 지정하지 않으면 현재 멤버에 대한 셀 텍스트만 반환됩니다. 서브큐브 차원(계정, ICP, 사용자정의)을 지정하지 않으면 모든 기준 멤버와 상위 멤버에 대한 셀 텍스트가 반환됩니다. 시나리오 및 연도 멤버를 지정하여 현재 서브큐브가 아닌 서브큐브에 대한 셀 텍스트를 검색할 수 있습니다. 기간, 값 및/또는 엔티티에 대한 멤버 목록을 지정하여 둘 이상의 서브큐브에 대한 셀 텍스트를 검색할 수도 있습니다. 셀 텍스트 레이블 검색과 차원 멤버나 셀 텍스트 레이블의 오름차순 또는 내림차순으로 셀 텍스트를 정렬할지 여부를 지정할 수 있습니다. 정렬 값을 지정하지 않으면 셀 텍스트 단위에서 정렬이 수행되지 않으며 데이터베이스에서의 순서에 따라 결과가 반환됩니다.

**구문**

```
Set CTU =  
HS.OpenCellTextUnit(POVExpression, Label(s), Sort_Dimension, Sort_Order)
```

여기서 *POVExpression*은 POV이고, *Label*은 셀 텍스트 레이블 none, one 또는 more이고, *Sort\_Order*는 Ascending 또는 Descending입니다.

**반환 값**

지정된 POV의 모든 셀 텍스트 및 레이블에 대한 정보를 반환합니다.

**예**

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014","", "", "")
```

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014.P{.[Base]}.A{.[Base]}", "", "Period", "Ascending")
```

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014.E{.[Base]}.P{.[Base]}", "CouponRate", "LABEL", "Ascending")
```

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014.E{.[Base]}.P{.[Base]}.A#Purchases", "", "Brands", "Descending")
```

여러 레이블에 대한 셀 텍스트를 검색하려는 경우 레이블을 심표로 구분된 값으로 입력할 수 있습니다. 레이블에 대한 값을 지정하지 않으면 모든 레이블이 검색됩니다.

### 예

```
Set CTU =
HS.OpenCellTextUnit("A#Sales", "Label-1,Label-2", "Label", "Descending")
```

## OpenDataUnit

통합, 계산 또는 환산하는 동안 처리할 데이터 단위를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합

### 구문

```
HS.OpenDataUnit(POVExpression)
```

여기서 *POVExpression*은 POV입니다. POV 표현식의 일부로 이 함수는 계정, ICP, C1, C2, C3 및 C4에 대해 사용자 정의 및 시스템 목록을 지원합니다.

### 반환 값

Sub Consolidate 서브루틴에 사용하면 데이터가 포함된 모든 레코드가 반환되지만 통합으로 플래그가 지정된 계정만 반환됩니다.

Sub Calculate 또는 Sub Translate 서브루틴에 사용하면 통합으로 플래그가 지정된 계정을 비롯하여 데이터가 포함된 모든 레코드가 반환됩니다.



#### 주:

IsConsolidated 속성이 True이면 계정이 통합됩니다.

### 예

```
Set DataUnit=HS.OpenDataUnit{"A{TotalRev.[Base]}.C1{C1Top.[Base]}.C2{MyC2List}.C3#[None]"}
```

## OpenDataUnitSorted

계산, 환산 또는 통합하는 동안 처리할 데이터 단위를 가져오며 데이터가 지정한 순서대로 정렬됩니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산

- 통합

### 구문

```
HS.OpenDataUnitSorted(POVExpression, <dimension to be sorted>, Ascending or Descending)
```

여기서, *POVExpression*은 POV 문자열이며, <*dimension to be sorted*>는 차원 이름 문자열이고 다음 6개 차원 중 하나여야 합니다.

"Account" 또는 "A", "ICP" 또는 "I", "Custom1" 또는 "C1", "Custom2" 또는 "C2", "Custom3" 또는 "C3", "Custom4" 또는 "C4".

계정, ICP, 사용자정의1, 사용자정의2, 사용자정의3, 사용자정의4 중 하나를 지정해야 합니다.

세번째 매개변수는 문자열 값("Ascending" 또는 "A", "Descending" 또는 "D")입니다.

### 예

```
Set DataUnit= HS.OpenDataUnitSorted("S#Actual.E#Group1", "C1", "Ascending")
```

## Owned

현재 처리되는 엔티티 쌍의 소유 엔티티를 가져옵니다. 이 함수는 지분 매입 규칙에 사용됩니다.

### 구문

```
HS.Entity.Owned
```

### 반환 값

소유 엔티티입니다.

### 예

```
Owned=HS.Entity.Owned
```

## Owner

현재 처리되는 엔티티 쌍의 소유자를 가져옵니다. 이 함수는 지분 매입 규칙에 사용됩니다.

### 구문

```
HS.Entity.Owner
```

### 반환 값

소유자 엔티티입니다.

### 예

```
Owner=HS.Entity.Owner
```

## PCon

현재 멤버나 지정된 멤버의 통합 지분을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.Node.PCon("S#Scenario.Y#Year.P#Period.E#Entity")
```

```
HS.Node.PCon("")
```



**주:**

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-40 PCon 함수 구문**

매개변수	설명
시나리오	적합한 시나리오 차원 멤버의 이름입니다.
연도	적합한 연도입니다.
기간	적합한 기간입니다.
엔티티	적합한 엔티티 차원 멤버의 이름입니다.

### 반환 값

멤버의 통합 지분입니다.

### 예

다음 예에서는 지정된 POV의 통합 지분을 가져옵니다.

```
Pcon = HS.Node.PCon("S#Actual.Y#2014.P#January.E#Regional.Connecticut")
```

## PEPU

EPU(지분 매입) 테이블에서 소유권의 백분율을 가져옵니다. 이 함수는 지분 매입 규칙에 사용됩니다.

**구문**

```
HS.PEPU("S#.Y#.P#", Owner, Owned)
```

**반환 값**

EPU 테이블의 소유 지분입니다.

**예**

```
HS.PEPU(S#Actual.Y#2014.P#Jan, Group, CT)
```

**또는**

```
""
```

```
HS.PEPU("", "", "")
```

기본 매개변수: 값이 비어 있으면 함수에서 현재 시나리오, 연도 및 기간의 엔티티 쌍에 대한 소유 지분이 반환됩니다.

**예**

```
Sub EquityPickUp()
```

```
Owned=Hs.Entity.Owned
```

```
OwnerCurrencyTotl=Hs.Entity.DefCurrency & "Total"
```

```
Hs.Clear "A#Inv.C4#EPU.I#" & Owned
```

```
Hs.Exp "A#Inv.C4#EPU.I#" & Owned & "=A#EQ.C4#C3Tot.I#[ICPTot].E#" &
```

```
Owned & ".V#" & OwnerCurrencyTotl & "*" & HS.PEPU (,,)
```

```
End Sub
```

## PeriodNumber

검색되는 데이터에 대한 뷰의 기간 수를 가져옵니다. 이 함수는 동적 SUB 함수에 사용할 수 있습니다.

**구문**

```
HS.View.PeriodNumber
```

## 반환 값

뷰의 기간 수를 나타내는 하나의 값입니다.

## 예

```
Hs.Dynamic "A#MarginPct=A#GrossMargin/HS.View.PeriodNumber"
```

기간	주기적	YTD	QTD	HYTD
1월	1	1	1	1
2월	1	2	2	2
3월	1	3	3	3
Q1	1	1	1	1
4월	1	4	1	4
5월	1	5	2	5
6월	1	6	3	6
Q2	1	2	1	2
HY1	1	1	1	1
7월	1	7	1	1
8월	1	8	2	2
9월	1	9	3	3
Q3	1	3	1	1
10월	1	10	1	4
11월	1	11	2	5
12월	1	12	3	6
Q4	1	4	1	2
HY2	1	2	2	1
연도	1	1	1	1

## PlugAcct

현재 계정 멤버나 지정된 계정의 플러그 계정을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.Account.PlugAcct ("Account")
```

```
HS.Account.PlugAcct ("")
```

여기서 *Account*는 적합한 계정 차원 멤버의 이름입니다.

 주:

빈 문자열("")을 사용하면 Sub Consolidate 서브루틴에 함수를 사용하는 경우에만 현재 멤버에 이 함수를 적용할 수 있습니다.

**반환 값**

멤버의 플러그 계정 이름을 지정하는 문자열입니다.

**예**

다음 예에서는 Sales 계정의 플러그 계정이 Plug1인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Account.PlugAcct("Sales") = "Plug1" Then
```

```
...
```

```
End If
```

## POwn

현재 멤버나 지정된 멤버의 최종 소유 지분을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.Node.POwn("S#Scenario.Y#Year.P#Period.E#Entity")
```

```
HS.Node.POwn("")
```

 주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

표 11-41 POwn 함수 구문

매개변수	설명
<i>Scenario</i>	적합한 시나리오 차원 멤버의 이름입니다.
<i>Year</i>	적합한 연도입니다.
<i>Period</i>	적합한 기간입니다.
<i>Entity</i>	적합한 엔티티 차원 멤버의 이름입니다.

**반환 값**

멤버의 소유 지분입니다.

**예**

다음 예에서는 지정된 POV의 소유 지분을 가져옵니다.

```
POwn = HS.Node.POwn("S#Actual.Y#2014.P#January.  
E#Regional.Connecticut")
```

## PVAForBalance

BALANCE 계정(ASSET 및 LIABILITY 계정)의 기본 환산 메소드를 결정합니다. 이 함수는 환산 중에 통화 변환에 대한 애플리케이션 기본값을 재정의하며 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.AppSettings.PVAForBalance
```

**반환 값**

BALANCE 계정이 주기적 값 환산 방법을 사용하면 True이고, BALANCE 계정이 환율 값 환산 방법을 사용하면 False인 부울 표현식입니다.



## 예

다음 예에서는 애플리케이션의 BALANCE 계정이 주기적 값 환산 방법을 사용하는 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.AppSettings.PVAForBalance = TRUE Then
```

```
...
```

```
End If
```

## PVAForFlow

FLOW 계정(REVENUE 및 EXPENSE 계정)의 기본 환산 메소드를 결정합니다. 이 함수는 환산 중에 통화 변환에 대한 애플리케이션 기본값을 재정의하며 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.AppSettings.PVAForFlow
```

### 반환 값

FLOW 계정이 주기적 값 환산 방법을 사용하면 True이고, FLOW 계정이 환율 값 환산 방법을 사용하면 False인 부울 표현식입니다.

## 예

다음 예에서는 애플리케이션의 FLOW 계정이 환율 값 환산 방법을 사용하는 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.AppSettings.PVAForFlow = FALSE Then
```

```
...
```

```
End If
```

## RateForBalance

BALANCE 계정(ASSET 및 LIABILITY 계정)에 사용할 기본 환산율 계정을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.AppSettings.RateForBalance
```

### 반환 값

BALANCE 계정에 사용할 기본 환산율이 포함된 환율 계정을 지정하는 문자열입니다.

### 예

다음 예에서는 BALANCE 계정의 기본 환산율 계정이 Rate1인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.AppSettings.RateForBalance = "Rate1" Then
```

```
...
```

```
End If
```

## RateForFlow

FLOW 계정(REVENUE 및 EXPENSE 계정)에 사용할 기본 환산율 계정을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.AppSettings.RateForFlow
```

### 반환 값

FLOW 계정에 사용할 기본 환산율이 포함된 환율 계정을 지정하는 문자열입니다.

**예**

다음 예에서는 FLOW 계정의 기본 환산율 계정이 Rate2인 경우 If 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.AppSettings.RateForFlow = "Rate2" Then
```

```
...
```

```
End If
```

## ReviewStatus

지정된 POV의 검토 상태를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.ReviewStatus("S#Scenario.Y#Year.P#Period.E#Entity.V#Value")
```

```
HS.ReviewStatus("")
```

 **주:**

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-42 ReviewStatus 함수 구문**

매개변수	설명
시나리오	적합한 시나리오 차원 멤버의 이름입니다.
연도	적합한 연도입니다.
기간	적합한 기간입니다.
엔티티	적합한 엔티티 차원 멤버의 이름입니다.
값	값 차원 멤버의 이름입니다.

**반환 값**

멤버의 검토 상태를 지정하는 문자열입니다. 적합한 검토 상태는 다음과 같습니다.

- 시작되지 않음
- 일차 통과
- 검토 레벨 1-10
- 제출됨
- 승인됨
- 게시됨
- 지원되지 않음

#### 예

다음 예에서는 지정된 POV의 검토 상태가 Submitted인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.ReviewStatus("") = "Submitted" Then
```

```
...
```

```
End If
```

## ReviewStatusUsingPhaseID

프로세스 관리 제출 단계 ID를 사용하여 지정된 POV의 검토 상태를 가져옵니다.

#### 구문

```
HS.ReviewStatusUsingPhaseID("S#Scenario.Y#Year.P#Period.E#Entity", n)
```

여기서 *n*은 프로세스 관리 제출 단계를 나타내는 정수입니다. 적합한 값은 1-9입니다.

#### 반환 값

[제출 단계 ID]를 사용하여 멤버의 검토 상태를 지정하는 문자열입니다. 적합한 검토 상태는 다음과 같습니다.

- 시작되지 않음
- 일차 통과
- 검토 레벨 1-10
- 제출됨
- 승인됨
- 게시됨
- 지원되지 않음

#### 예

```
HS.ReviewStatusUsingPhaseID("S#Actual.Y#2014.P#January.E#Connecticut", 3)
```

# Round

Exp 함수의 데이터를 반올림합니다. 인수에서 반올림 정도를 지정합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합

## 팁:

계산 규칙에 반올림을 다양하게 적용해야 하는 경우 Round가 포함된 명령문을 여러 개 포함할 수 있습니다.

## 구문

```
HS.Round(Unit)
```

여기서 *Unit*는 반올림할 요소입니다. 값이 1이면 가장 가까운 정수로 반올림됩니다. 값이 0.1이면 가장 가까운 소수 첫째 자리로 반올림됩니다. 값이 0이면 반올림이 해제됩니다.

이 인수에 0을 지정하면 계산 규칙의 모든 후속 Exp 함수에 대해 반올림이 해제됩니다. 이 구문은 HS.NoRound: HS.Round(0)를 사용하는 것과 같습니다.

## 주의:

계정의 NumDecimalPlaces 속성은 소수점 오른쪽에 표시될 수 있는 최대 자릿수를 결정합니다. Round 함수는 이 속성을 재정의하지 않습니다.

## 반환 값

없습니다.

**예**

다음 예에서는 SalesRound 계정에 삽입된 금액을 가장 가까운 소수 첫째 자리로 반올림한 다음 NoRound를 사용하여 SalesNoRound 계정의 셀에 삽입된 금액에 대해 반올림을 해제합니다.

```
HS.Round(0.1)
```

```
HS.Exp"A#SalesRound = A#Sales"
```

```
HS.NoRound
```

```
HS.Exp"A#SalesNoRound = A#Sales"
```

## Scale

지정된 통화의 스케일을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.Currency.Scale("Currency")
```

```
HS.Currency.Scale(Var1)
```

**표 11-43 Scale 함수 구문**

매개변수	설명
<i>Currency</i>	적합한 통화의 이름입니다.
<i>Var1</i>	VisualBasic 변수입니다.

**반환 값**

지정된 통화의 스케일을 나타내는 숫자입니다(0-9). 소수점 위치를 식별하여 통화에서 금액이 표시되고 저장되는 단위를 지정합니다. 반환 값은 다음과 같습니다.

- 0 = 단위
- 1 = 십
- 2 = 백
- 3 = 천

- 4 = 만
- 5 = 십만
- 6 = 백만
- 7 = 천만
- 8 = 억
- 9 = 십억

### 예

다음 예에서는 프랑스 프랑(FF)의 스케일이 3인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Currency.Scale("FF") = 3 Then
```

```
...
```

```
End If
```

## SecurityAsPartner

엔티티나 상위 멤버를 내부거래 파트너로 사용할 때 지정된 엔티티나 상위 멤버에 지정되는 보안 클래스를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.Entity.SecurityAsPartner("Entity")
```

```
HS.Entity.SecurityAsPartner("")
```

```
HS.Entity.SecurityAsPartner(Var1)
```

```
HS.Parent.SecurityAsPartner("Entity")
```



#### 주:

현재 엔티티 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

표 11-44 SecurityAsPartner 함수 구문

매개변수	설명
<i>Entity</i>	적합한 엔티티 차원 멤버의 이름입니다.
<i>Var1</i>	VisualBasic 변수입니다.

**반환 값**

ICP로 사용할 때 엔티티나 상위 멤버에 지정되는 보안 클래스가 포함된 문자열입니다.

**예**

다음 예에서는 France를 내부거래 파트너로 사용할 때 France의 보안 클래스가 Class1인 경우 If...Then 문과 If...End 문 사이의 명령문이 실행됩니다.

```
If HS.Entity.SecurityAsPartner("France") = "Class1" Then
```

```
...
```

```
End If
```

```
If HS.Parent.SecurityAsPartner("France") = "Class1" Then
```

```
...
```

```
End If
```

## SecurityClass

지정된 멤버의 보안 클래스를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당



**구문**

```
HS.<Object>.SecurityClass("Element")
```

```
HS.<Object>.SecurityClass("")
```

```
HS.<Object>.SecurityClass(Var1)
```

```
HS.Custom(Dimension).SecurityClass("Element")
```

**주:**

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-45 SecurityClass 함수 구문**

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>• 계정</li> <li>• 시나리오</li> <li>• 엔티티</li> <li>• 상위</li> <li>• Custom1...4</li> <li>• 사용자정의 (사용자정의 차원 레이블)</li> </ul>
Element	선택한 객체에 따라 다음 차원 중 하나에서 적합한 멤버의 이름입니다. <ul style="list-style-type: none"> <li>• 계정</li> <li>• 시나리오</li> <li>• 엔티티</li> <li>• Custom1...4</li> <li>• 사용자정의</li> </ul>
Var1	VisualBasic 변수입니다.

**반환 값**

지정된 멤버에 지정된 보안 클래스의 이름입니다.

**예**

다음 예에서는 Cash 계정에 지정된 보안 클래스가 Class1인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Account.SecurityClass("Cash") = "Class1" Then

    ...

End If

If HS.Custom(Dimension).SecurityClass(Member) Then

    ...

End If
```

## SetCellTextWithLabel

지정된 POV 및 셀 텍스트 레이블의 셀 텍스트 정보를 업데이트합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

계산

**구문**

```
HS.SetCellTextWithLabel("POVExpression", "CellTextLabel",
"CellTextString")
```

여기서, **POVExpression**은 멤버의 조합이고 **CellTextLabel**은 기본 셀 텍스트 레이블 ([Default])이거나 로드된 적합한 셀 텍스트 레이블 중 하나입니다. **CellTextString**은 작성할 텍스트 문자열입니다. 계정, ICP 및 사용자정의 POV 멤버를 지정해야 합니다. 시나리오, 연도, 기간, 뷰, 엔티티 및 값은 기본적으로 현재 멤버로 지정됩니다.

**반환 값**

없음

**예**

```
HS.SetCellTextWithLabel "A#Asset.I#[ICP
None].C1#[None].C2#[None].C3#[None].C4#[None]", "Rating", "AAA"
```

## SetData

개별 레코드를 설정합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산

- 환산

### 구문

```
HS.SetData lView, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
dData, bAddToExistingData
```

**표 11-46 SetData 함수 구문**

매개변수	설명
<i>lView</i>	0 = 시나리오 뷰 1 = 주기적 2 = YTD
<i>lAccount</i>	데이터를 설정할 계정의 ID 번호입니다.
<i>lICP</i>	데이터를 설정할 ICP의 ID 번호입니다.
<i>lCustom1...4</i>	데이터를 설정할 사용자정의 차원의 ID 번호입니다.
<i>dData</i>	설정할 데이터 값입니다.
<i>bAddToExistingData</i>	True = 데이터 누계 False = 데이터 바꾸기

### 반환 값

없습니다.

### 예

```
HS.SetData 2, 002, , , , , 25000, TRUE
```

## SetDataWithPOV

노드나 통화 큐브에 데이터를 삽입합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산

### 구문

```
HS.SetDataWithPOV POV, dData, bAddToExistingDataInCache
```

**표 11-47 SetData 함수 구문**

매개변수	설명
<i>POV</i>	적합한 POV
<i>dData</i>	설정할 데이터 값입니다.
<i>bAddToExistingData</i>	True = 데이터 누계 False = 데이터 바꾸기

**반환 값**

없습니다.

**예**

```
HS.SetDataWithPOV "W#YTD.A#Asset.I#[ICP
None].C1#None.C2#None.C3#None.C4#None",25000,TRUE
```

## SubmissionGroup

차원 멤버의 프로세스 관리 제출 그룹을 가져옵니다.

**구문**

```
HS.<Dimension>.SubmissionGroup(Dimension member)
```

여기서 *Dimension*은 계정, 사용자정의1...4 또는 사용자정의 차원 중 하나입니다.

```
HS.Account.SubmissionGroup(Account)
```

```
HS.Custom1.SubmissionGroup(Custom1)
```

```
HS.Custom2.SubmissionGroup(Custom2)
```

```
HS.Custom3.SubmissionGroup(Custom3)
```

```
HS.Custom4.SubmissionGroup(Custom4)
```

```
HS.Custom(Custom Dimension
Label).SubmissionGroup((CustomDimensionLabel))
```

**반환 값**

차원 멤버의 제출 그룹을 나타내는 정수입니다. 적합한 값은 1-99입니다.

**예**

```
HS.Account.SubmissionGroup("Sales")
```

```
HS.Custom("Product").SubmissionGroup("P3000-Phones")
```

## SupportsProcessManagement

시나리오에서 프로세스 관리를 지원하는지 여부를 확인합니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.Scenario.SupportsProcessManagement ("Scenario")
```

```
HS.Scenario.SupportsProcessManagement ("")
```



**주:**

현재 시나리오에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

```
HS.Scenario.SupportsProcessManagement (Var1)
```

**표 11-48 SupportsProcessManagement 함수 구문**

매개변수	설명
<i>Scenario</i>	적합한 시나리오입니다.
<i>Var1</i>	시나리오 멤버를 나타내는 VBScript 변수입니다.

### 반환 값

시나리오에 프로세스 관리를 사용할 수 있으면 True이고, 그렇지 않으면 False인 부울입니다.

### 예

다음 예에서는 실제 시나리오에 프로세스 관리를 사용할 수 있는 경우 If 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Scenario.SupportsProcessManagement ("Actual") = "TRUE" then
```

```
...
```

```
End IF
```

## SupportsTran

내부거래 트랜잭션을 지원하는 애플리케이션 계정을 지정합니다. 이 함수는 트랜잭션 규칙에만 사용할 수 있습니다.

### 구문

```
HS.SupportsTran "POVExpression"
```

여기서 *POVExpression*은 계정, 사용자정의1-4, 시나리오, 엔티티 및 연도 멤버의 조합입니다.

### 반환 값

없습니다.

### 예

```
HS.SupportsTran "S#ActMon.A#RecltIC.Cl#Closing"
```

## SwitchSign

대변이 현재 사용자정의 멤버에 대해 차변으로 전환되는지 또는 지정된 사용자정의 멤버에 대해 차변으로 전환되는지를 확인합니다. 이 함수는 다음 규칙을 사용하여 차변/대변 기호를 전환합니다.

- ASSET에서 LIABILITY
- LIABILITY에서 ASSET
- EXPENSE에서 REVENUE
- REVENUE에서 EXPENSE
- BALANCE에서 FLOW
- FLOW에서 BALANCE

이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

## 구문

```
HS.<Object>.SwitchSign("Member")
```

```
HS.<Object>.SwitchSign("")
```

```
HS.Custom(Dimension).SwitchSign(Member)
```



### 주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-49 SwitchSign 함수 구문**

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>• Custom1</li> <li>• Custom2</li> <li>• Custom3</li> <li>• Custom4</li> <li>• Custom</li> </ul>
멤버	적합한 사용자정의 차원 멤버의 이름입니다.

### 반환 값

대변이 사용자정의 멤버에 대해 차변으로 전환되면 True이고, 대변과 차변이 전환되지 않으면 False인 부울 표현식입니다.

## 예

다음 예에서는 대변과 차변이 전환되는 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Custom1.SwitchSign("") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Product").SwitchSign("P3000-Phones") = TRUE Then
```

```
...
```

```
End If
```

## SwitchType

계정 유형이 현재 사용자정의 멤버에 대해 전환되는지 또는 지정된 사용자정의 멤버에 대해 전환되는지를 확인합니다. 이 함수는 다음 규칙을 사용하여 사용자정의 차원 멤버의 계정 유형을 변경합니다.

- ASSET에서 EXPENSE
- EXPENSE에서 ASSET
- LIABILITY에서 REVENUE
- REVENUE에서 LIABILITY
- BALANCE에서 FLOW
- FLOW에서 BALANCE

이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.<Object>.SwitchType ("Member")
```

```
HS.<Object>.SwitchType ("")
```





주:

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-50 SwitchType 함수 구문**

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>• Custom1</li> <li>• Custom2</li> <li>• Custom3</li> <li>• Custom4</li> <li>• Custom</li> </ul>
멤버	적합한 사용자정의 차원 멤버의 이름입니다.

**반환 값**

계정 유형이 사용자정의 멤버에 대해 전환되면 True이고, 계정 유형이 전환되지 않으면 False인 부울 표현식입니다.

**예**

다음 예에서는 계정 유형이 현재 사용자정의1 멤버에 대해 전환되는 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```

If HS.Custom1.SwitchType("") = TRUE Then
    ...

End If

If HS.Custom("Product").SwitchType("P3000-Phones") = TRUE Then
    ...

End If
  
```

## Trans

연 누계 방법을 사용하여 통화를 환산합니다. 이 함수는 환산 규칙에 사용할 수 있습니다.

**구문**

```

HS.Trans("DestPOV","SourcePOV","Rate1","Rate2")
  
```

**표 11-51 Trans 함수 구문**

매개변수	설명
<i>DestPOV</i>	대상 POV입니다. 대상은 계정, 사용자정의1...4 및 ICP 멤버의 조합일 수 있습니다. 시스템은 지정되지 않은 각 차원에 대해 모든 적합한 멤버에 씁니다. 지정된 각 차원에 대해서는 지정된 멤버에만 씁니다.
<i>SourcePOV</i>	소스 POV입니다. 소스는 차원의 조합일 수 있습니다. 계정, 사용자정의1...4 및 ICP 차원을 지정하지 않은 경우 대상 멤버와 동일한 멤버가 사용됩니다. 시나리오, 연도, 기간 및 엔티티 차원을 지정하지 않은 경우 현재 멤버가 사용됩니다. 값 차원을 지정하지 않은 경우 <EntityCurrTotal> 멤버가 사용됩니다. 소스 POV가 비어 있는 경우 대상 POV가 소스 POV로 사용됩니다.
<i>Rate1-2</i>	환율입니다. 환율은 상수, 환율 계정 또는 특정 셀일 수 있습니다. 환율 계정은 엔티티 및 [None] 엔티티에 대한 입력입니다. 기본 환산 프로세스에 대한 자세한 내용은 <a href="#">기본 환산</a> 을 참조하십시오.

**반환 값**

없습니다.

**예**

다음 예에서는 Rate1 계정의 환율을 사용하여 연 누계 방법으로 Sales 계정을 환산합니다.

```
HS.Trans("A#Sales", "A#LastYearSales", "A#Rate1", "")
```

## TransPeriodic

주기적 방법을 사용하여 통화를 환산합니다. 이 함수는 환산 규칙에 사용할 수 있습니다.

**구문**

```
HS.TransPeriodic("DestPOV", "SourcePOV", "Rate1", "Rate2")
```

**표 11-52 TransPeriodic 함수 구문**

매개변수	설명
<i>DestPOV</i>	대상 POV입니다. 대상은 계정, 사용자정의1-4 및 ICP 멤버의 조합일 수 있습니다. 시스템은 지정되지 않은 각 차원에 대해 모든 적합한 멤버에 씁니다. 지정된 각 차원에 대해서는 지정된 멤버에만 씁니다.

표 11-52 (계속) TransPeriodic 함수 구문

매개변수	설명
<i>SourcePOV</i>	소스 POV입니다. 소스는 차원의 조합일 수 있습니다. 계정, 사용자정의 1, 사용자정의 2, 사용자정의 3, 사용자정의 4 및 ICP 차원을 지정하지 않은 경우 대상 멤버와 동일한 멤버가 사용됩니다. 시나리오, 연도, 기간 및 엔티티 차원을 지정하지 않은 경우 현재 멤버가 사용됩니다. 값을 지정하지 않은 경우 EntityCurrTotal 멤버가 사용됩니다. 소스가 비어 있는 경우 대상이 소스로 사용됩니다.
<i>Rate1-2</i>	환율입니다. 환율은 상수, 환율 계정 또는 특정 셀일 수 있습니다. 환율 계정은 엔티티 및 None 엔티티에 대한 입력입니다. 기본 환산 프로세스에 대한 자세한 내용은 <a href="#">기본 환산</a> 을 참조하십시오.

**반환 값**

없음

**예**

다음 예에서는 Rate1 계정의 환율을 사용하여 주기적 방법으로 Sales 계정을 환산합니다.

```
HS.TransPeriodic("A#Sales", "A#LastYearSales", "A#Rate1", "")
```

## UD1...3

현재 멤버나 지정된 멤버의 UserDefined1...3 속성에 저장된 텍스트를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

**구문**

```
HS.<Object>.UD1...3(strElement)
```

```
HS.<Object>.UD1...3("")
```

```
HS.Custom(Dimension).UD1(Member)
```

**주:**

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-53 UD1...3 함수 구문**

매개변수	설명
<Object>	다음 객체 키워드 중 하나입니다. <ul style="list-style-type: none"> <li>계정</li> <li>엔티티</li> <li>상위</li> <li>시나리오</li> <li>Custom1...4</li> <li>사용자정의</li> </ul>
요소	선택한 객체에 따라 다음 차원 중 하나에서 적합한 멤버의 이름입니다. <ul style="list-style-type: none"> <li>계정</li> <li>엔티티</li> <li>상위</li> <li>시나리오</li> <li>Custom1...4</li> <li>사용자정의</li> </ul>

**반환 값**

멤버에 대해 저장된 사용자 정의 텍스트가 포함된 문자열입니다.

**예**

다음 예에서는 UD1 계정의 사용자 정의 텍스트가 History인 경우 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.Account.UD1(strAccount) = "History" Then
```

```
...
```

```
End If
```

```
If HS.Custom("Product").UD1("P3000-Phones") = "History" Then
```

```
...
```

```
End If
```

## ValidationAccount

애플리케이션의 검증 계정을 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.AppSettings.ValidationAccount
```

### 반환 값

애플리케이션의 검증 계정 이름을 지정하는 문자열입니다.

### 예

애플리케이션의 검증 계정이 MyAccount이면 If...Then 문과 End If 문 사이의 명령문이 실행됩니다.

```
If HS.AppSettings.ValidationAccount = "MyAccount" Then
```

```
...
```

```
End If
```

## ValidationAccountEx

프로세스 관리 제출 단계의 검증 계정을 가져옵니다.

### 구문

```
HS.AppSettings.ValidationAccountEx(n)
```

여기서 *n*은 프로세스 관리 제출 단계를 나타내는 정수입니다. 적합한 값은 1-9입니다.

### 반환 값

프로세스 관리 제출 단계의 확인 계정 이름을 지정하는 문자열입니다.

### 예

다음 예에서는 제출 단계 5에 대해 정의된 검증 계정이 반환됩니다.

```
HS.AppSettings.ValidationAccountEx(5)
```

## XBRLTags

지정된 계정 멤버에 지정된 XBRL 태그를 가져옵니다. 이 함수는 다음 유형의 규칙에 사용할 수 있습니다.

- 계산
- 환산
- 통합
- 할당

### 구문

```
HS.Account.XBRLTags ("Account")
```

```
HS.Account.XBRLTags ("")
```



**주:**

현재 멤버에 이 함수를 적용하려면 빈 문자열("")을 사용하십시오.

**표 11-54 XBRLTags 함수 구문**

매개변수	설명
계정	적합한 계정입니다.

### 반환 값

지정된 계정의 XBRL 태그를 지정하는 문자열입니다.

# 12

## 사용자정의 함수

이 섹션에서는 Oracle Hyperion Financial Management에서 관리 보고 함수와 계획 함수에 사용할 수 있는 내부 HS 사용자정의 함수를 보여줍니다. 함수에는 설명, 함수 유형, 구문, 예 및 샘플 스크립트가 포함되어 있습니다.

**표 12-1 관리 보고 함수**

사용자정의 함수	설명	구문	함수 유형	Hyperion Enterprise의 해당 항목
평균	재무 평균을 계산합니다.	Average (POV, Periods)	함수	AVE A12
Cumulative	이전 기간의 금액을 누계합니다.	Cumulative (POV, View, NumPeriod)	함수	CUM CTD YTD
Difference	현재 금액과 기초 금액의 차액을 계산합니다.	Difference (POV, View)	함수	DIF DFB
DSO	매출 회전 일수를 계산합니다.	DSO (DSO, Debtor, Sales, DIP)	프로시저	프로시저
Opening	기초 잔액을 이월합니다.	Opening (POV, View)	함수	OPE BASE BASEFLOW
Rate	상대 환율을 구합니다.	Rate (ExchangeRate, Triangulation Currency)	함수	CrossRate

**표 12-2 계획 함수**

사용자정의 함수	설명	매개변수	함수 유형
Units_Rates	단위 * 비율(C=A*B)	Unit_Rates (Description, Units, Rates)	프로시저
Custom_Alloc	사용자정의 차원에서 할당합니다.	Custom_Alloc (Destination, Source, Factor, FactorN, FactorD, Elimination)	프로시저
Increase_Decrease	계정을 백분율만큼 늘리거나 줄입니다.	Increase_Decrease (Destination, Source, Factor, Scale, Inverse)	프로시저
Pro_Rata_Ratio	두 계정 간의 비율입니다.	Pro_Rata_Ratio (Destination, SourceN, SourceD)	프로시저

표 12-2 (계속) 계획 함수

사용자정의 함수	설명	매개변수	함수 유형
Spread	연도의 모든 기간에 총 금액을 분산합니다.	Spread (Destination, Source, Factor, FactorN, FactorD, Temp, Per)	프로시저

## 관리 보고 함수

이 섹션에서는 사용 가능한 관리 보고 사용자정의 함수를 보여 줍니다.

### 평균

여러 기간에 대한 계정의 평균 값을 계산합니다.

#### 반환 값

HS.EXP 함수의 일부로 사용할 정확한 표현식을 나타내는 문자열을 반환합니다.

#### 구문

Average (*PointOfView*, *Periods*)

표 12-3 Average 함수 구문

매개변수	적합한 값
PointOfView	계정, 사용자정의1...4, ICP 멤버 등의 적합한 조합입니다(예: "A#CASH.C1#[None].I#[ICP Top]"). 플로우 유형 계정의 경우 함수에서 주기적 값만 평균 계산에 사용합니다.
기간	다음 값 중 하나여야 합니다. YTD - 현재 연도의 기간 1부터 누적 데이터의 평균을 계산하려면 연 누계 옵션을 지정합니다. 주기적 - 현재 연도에서 현재 기간과 바로 이전 기간만 평균 계산에 사용하려면 주기적 옵션을 지정합니다. 첫 번째 기간의 경우 이 값이 소스와 같습니다. [임의의 양의 정수] - 평균을 계산할 기간 수를 지정합니다. 매월 범주의 연속 연도 평균을 구하려면 12를 지정합니다.

#### 자세한 설명

이 함수는 지정한 개수의 이전 기간에 대한 계정의 평균 값을 계산합니다. 소스가 잔액 유형 계정인 경우 평균이 입력된 데이터를 기반으로 합니다. 소스가 플로우 유형 계정인 경우 평균이 주기적 데이터만 기반으로 합니다.

Average 값은 *Periods* 매개변수를 기준으로 다르게 파생됩니다.



- *Periods* 매개변수가 YTD인 경우 평균 값은 현재 기간까지 현재 연도의 모든 기간의 합계를 현재 기간 수로 나눈 값입니다.
- *Periods* 매개변수가 주기적인 경우 평균 값은 현재 기간과 바로 이전 기간의 합계를 2로 나눈 값입니다. 현재 기간이 연도의 첫 번째 기간인 경우 평균 값이 소스 값과 같습니다.
- *Periods* 매개변수가 숫자인 경우 평균 값은 현재 기간과 지정한 기간 수의 각 이전 기간의 합계를 지정한 개수로 나눈 값입니다.

예

SALES 계정은 Average 사용자정의 함수에서 사용된 *Periods* 매개변수에 따라 2014년 1월, 2월 및 3월에 대해 다음 값을 반환합니다. 처리 중인 시나리오에 대해 설정된 기본 뷰는 YTD입니다.

표 12-4 Average 함수 예

계정	Oct2013	Nov2013	Dec2013	Jan2014	Feb2014	Mar2014
A#Sales	9,000	10,500	11,700	800	1,900	3,200
Average ("A#Sales", "YTD")	N/A	N/A	N/A	800	950	1,067
Average ("A#Sales", "Periodic")	N/A	N/A	N/A	800	950	1,200
Average ("A#Sales", "3")	N/A	N/A	N/A	1,167	1,033	1,067

**샘플 스크립트**

```
' sample statement written in the calling routine
```

```
Sub Calculate()
```

```
Hs.Exp "A#AVG_SALES = "& Average("A#Sales", "12")
```

```
End Sub
```

```
' programming of the AVERAGE function
```

```
FUNCTION Average(strPOV, strPERIOD)
```

```
  DIM nPERIOD
```

```
  DIM strCUM
```

```
  DIM i
```

```
  strPOV = UCASE(strPOV)
```

```
  strPERIOD = UCASE(strPERIOD)
```

```
  IF strPERIOD = "PERIODIC" THEN
```

```
    IF HS.PERIOD.ISFIRST = TRUE THEN
```

```
      nPERIOD = 1
```

```
    ELSE
```

```
      nPERIOD = 2
```

```
    END IF
```

```
  ELSEIF strPERIOD = "YTD" THEN
```

```
    nPERIOD = HS.PERIOD.NUMBER()
```

```
  ELSEIF CINT(strPERIOD) > 0 THEN
```

```
    nPERIOD = CINT(strPERIOD)
```

```
  ELSE
```

## Cumulative

지정한 계정에 대한 이전 기간 값의 합계를 계산합니다.

### 반환 값

HS.EXP 함수의 일부로 사용할 정확한 표현식을 나타내는 문자열을 반환합니다.

### 구문

Cumulative (*PointOfView*, *View*, *NumPeriod*)

**표 12-5 Cumulative 함수 구문**

매개변수	적합한 값
PointOfView	계정, 사용자정의1...4, ICP 멤버 등의 적합한 조합입니다(예: "A#CASH.C1#[None].I#[ICP Top]").
뷰	다음 값 중 하나여야 합니다. "" (큰따옴표) - 처리 중인 시나리오에 대해 정의된 기본 뷰(YTD 또는 주기적)를 기반으로 합니다. YTD - 사용자가 시나리오에 대해 설정된 기본 뷰를 재정의하는 연 누계 옵션을 지정합니다. 주기적 - 시나리오에 대해 설정된 기본 뷰를 재정의하는 주기적 옵션을 지정합니다.
NumPeriod	현재 기간부터 시작하여 누계할 현재 시나리오의 기간 수를 나타내는 정수입니다. NumPeriod가 0 또는 음수이면 함수가 현재 연도의 시작부터 합산됩니다.

### 자세한 설명

이 함수는 지정된 기간의 합계나 지정된 계정에 대한 연 누계 합계를 계산합니다. 기본적으로 누계 데이터 뷰는 시나리오입니다. 하지만 플로우 유형 계정에 대해 이 기본값을 재정의할 수 있습니다.

- *View* 매개변수가 YTD이면 함수가 연 누계 값을 누계합니다.
- *View* 매개변수가 주기적이면 함수가 주기적 값을 누계합니다.
- *View* 매개변수가 공백("")이면 함수가 시나리오 기본 뷰를 사용하여 데이터를 누계합니다.

### 예

CASH 계정은 Cumulative 함수에서 사용된 *Number* 매개변수에 따라 2014년 1월, 2월 및 3월에 대해 다음 값을 반환합니다.

SALES 계정은 Cumulative 함수에서 사용된 *View* 및 *Number* 매개변수에 따라 2014년 1월, 2월 및 3월에 대해 다음 값을 반환합니다. 처리 중인 시나리오에 대해 설정된 기본 뷰는 YTD입니다.

표 12-6 Cumulative 함수 예

계정	Oct2013	Nov2013	Dec2013	Jan2014	Feb2014	Mar2014
A#Cash	1,000	1,500	1,200	800	1,100	1,300
Cumulative("A#Cash","0")	N/A	N/A	N/A	800	1,900	3,200
Cumulative("A#Cash", "3")	N/A	N/A	N/A	3,500	3,100	3,200
A#Sales	9,000	10,500	11,700	800	1,900	3,200
Cumulative("A#Sales", "0")	N/A	N/A	N/A	800	2,700	5,900
Cumulative("A#Sales", "Periodic", "0")	N/A	N/A	N/A	800	1,900	3,200
Cumulative("A#Sales", "Periodic", "3")	N/A	N/A	N/A	3,500	3,100	3,200

## 샘플 스크립트

```
' sample statement written in the calling routine
```

```
Sub Calculate()
```

```
HS.EXP "A#TOT_Cash =" &Cumulative("A#Cash","",0)
```

```
End Sub
```

```
' programming of the Cumulative function
```

```
Function Cumulative(StrPov, StrVIEW, nPERIOD)
```

```
DIM strCUM
```

```
DIM i
```

```
IF nPERIOD <= 0 THEN
```

```
nPERIOD = HS.PERIOD.NUMBER() - 1
```

```
ELSE
```

```
nPERIOD = nPERIOD - 1
```

```
END IF
```

```
IF strVIEW = "" THEN
```

```
strVIEW = HS.SCENARIO.DEFAULTVIEW("")
```

```
END IF
```

```
strPOV = UCASE(strPOV)
```

```
strVIEW = UCASE(strVIEW)
```

```
IF strVIEW = "PERIODIC" THEN
```

```
strVIEW = ".W#PERIODIC"
```

```
ELSEIF strVIEW = "YTD" THEN
```

## Difference

현재 기간 값과 기초 값의 차액을 계산합니다.

### 반환 값

HS.EXP 함수의 일부로 사용할 정확한 표현식을 나타내는 문자열을 반환합니다.

### 구문

Difference (PointOfView, View)

**표 12-7 Difference 함수 구문**

매개변수	적합한 값
PointOfView	계정, 사용자정의1...4, ICP 멤버 등의 적합한 조합입니다(예: "A#CASH.C1#[None].I#[ICP Top]").
뷰	다음 값 중 하나여야 합니다. ""(큰따옴표) - 처리 중인 시나리오에 대해 정의된 기본 뷰(YTD 또는 주기적)를 기반으로 합니다. YTD - 시나리오에 대해 설정된 기본 뷰를 재정의하는 연 누계 옵션을 지정합니다. 주기적 - 시나리오에 대해 설정된 기본 뷰를 재정의하는 주기적 옵션을 지정합니다.

### 자세한 설명

이 함수는 현재 기간 값과 기초 값의 차액을 계산합니다. (현재 - 기초)

기초 값은 함수에 전달된 View 매개변수를 기준으로 다르게 파생됩니다.

- View 매개변수가 YTD이면 기초 값이 이전 연도의 마지막 기간에서 파생됩니다.
- View 매개변수가 주기적이면 기초 값이 현재 연도의 이전 기간에서 파생됩니다. 현재 기간이 연도의 첫 번째 기간이면 기초 값이 이전 연도의 마지막 기간에서 검색됩니다.
- View 매개변수가 공백(" ")이면 기초 값이 시나리오의 기본 데이터 뷰를 기반으로 합니다.

### 예

CASH 계정은 Difference 함수에서 사용된 View 매개변수에 따라 2014년 1월, 2월 및 3월에 대해 다음 값을 반환합니다. 처리 중인 시나리오에 대해 설정된 기본 뷰는 YTD입니다. Difference 함수는 현재 기간 값에서 기초 값을 뺍니다.

**표 12-8 Difference 함수 예**

계정	Dec2013	Jan2014	Feb2014	Mar2014
A#Cash	900	1,200	1,100	1,500
Difference("A#Cash", N/A ""		300	200	600

표 12-8 (계속) Difference 함수 예

계정	Dec2013	Jan2014	Feb2014	Mar2014
Difference("A#Cash", N/A "YTD")		300	200	600
Difference("A#Cash", N/A "Periodic")		300	-100	400

**샘플 스크립트**

```
' sample statement written in the calling routine

Sub Calculate()

Hs.Exp = "A#DiffCash" & Difference("A#Cash", "YTD")

End Sub

' programming of the DIFFERENCE function

FUNCTION DIFFERENCE(strPOV,strVIEW)

IF strVIEW = "" THEN

strVIEW = HS.SCENARIO.DEFAULTVIEW ("" )

END IF

strPOV = UCASE(strPOV)
strVIEW = UCASE(strVIEW)

IF strVIEW = "PERIODIC" THEN

DIFFERENCE = ""&" "&strPOV"-& strPOV & ".P#PRIOR" &"")""

ELSEIF strVIEW = "YTD" THEN

DIFFERENCE = "" ("&strPOV &"-" & strPOV & ".Y#PRIOR.P#LAST" &"") ""

ELSE

EXIT FUNCTION

END IF

END FUNCTION
```



## DSO - 매출 회전 일수

소모 메소드를 사용하여 현재 기간 채무자의 영업 일수를 계산합니다.

### 반환 값

이 루틴은 현재 기간 거래 채무자 수치에 포함된 영업 일수의 금액을 나타내는 단일 값을 계산합니다. 포함된 DSO 서브루틴에서는 다음과 같이 가정합니다.

- 채무자와 판매가 모두 양수입니다.
- 루틴에서 HS.GETCELL 함수를 사용하기 때문에 제공된 매개변수가 완전히 정의된 POV(예: 계정/C1/C2/C3/C4/ICP)입니다.
- 이 루틴은 가능한 가장 이전 시점까지의 일수를 계산합니다. 하지만 임의 기간의 주기적 판매 값이 음수 또는 0이면 중지됩니다.

### 구문

```
CALL DSO (strDSO,strDEBTOR,strSALES,strDIP)
```

**표 12-9 DSO 함수 구문**

매개변수	적합한 값
strDSO	사용자정의 및 내부거래 차원으로 완전히 정의된 계정입니다. 이 계정은 계산 대상입니다.
strDEBTOR	사용자정의 및 내부거래 차원으로 완전히 정의된 계정입니다. 이 계정은 현재 기간 거래 채무자의 소스입니다.
strSALES	사용자정의 및 내부거래 차원으로 완전히 정의된 계정입니다. 이 계정은 판매 소스입니다. 빈도 참조를 구체적으로 제외합니다.
strDIP	사용자정의 및 내부거래 차원으로 완전히 정의된 계정입니다. 이 계정은 기간 일수의 소스입니다. [None] 엔티티에 있는 것으로 가정됩니다.

### 자세한 설명

이 루틴은 현재 기간에 대한 Debtors 계정(매개변수 2) 및 Sales 계정(매개변수 3)의 값을 사용하고 비교합니다. 두 값 중 하나가 0 또는 음수이면 계산이 중지됩니다. 현재 기간부터 역순으로 Debtors 값이 누적 판매 값을 초과하는 각 연속 기간에 대해 루틴이 Period 계정(매개변수 4)에 일수로 지정된 해당 기간의 일수를 실행 합계에 더합니다.

채무자 값이 이런 방식으로 모두 소모된 경우 최종 기간의 일수가 주기적 판매 값에 대한 만료되지 않은 채무자 비율로 계산됩니다.

최종적으로, 이 루틴은 실행 합계를 Destination 계정(매개변수 1)에 게시합니다.

### 예

이 예에서는 표시된 달의 총 회전 일수를 계산합니다.

표 12-10 DSO 함수 예

Month	채무자	판매 기간	달의 일수	DSO 공식	총 DSO
9월	12,000	2,500	30	100%	30
8월	N/A	1,750	31	100%	31
7월	N/A	2,250	31	100%	31
6월	N/A	2,500	30	100%	30
5월	N/A	2,000	31	100%	31
4월	N/A	2,250	30	2000/2250	26.7
합계	N/A	N/A	N/A	N/A	179.7

## 샘플 스크립트

```
' Use within the calculation section:
' 1. Standard use

CALL DSO("A#DSO","A#TradeDebtors.C1#AllAges.C2#[None].I#[ICP
Top","A#TotalSales.C1#[None].C2#AllProducts.I#[ICP Top]", "A#DIP")

' 2. Use with a common custom dimension

set vPRODUCT = ARRAY("C2#PRODUCT1","C2#PRODUCT2", .... , "C2#PRODUCTn")

FOR EACH iITEM IN vPRODUCT

CALL DSO(" " A#DSO." "&iITEM," " A#TradeDebtors.C1#AllAges.I#[ICP
Top]." "&iITEM," " A#TotalSales.C1#[None].I#[ICP Top]." "&iITEM," "
A#DIP")

NEXT

' Actual script of Sub-routine

SUB DSO(strDSO,strDEBTOR,strSALES,strDIP)

DIM vTEST
DIM vDSO
DIM vCOUNT
```

```
DIM vXS_1
DIM vXS

HS.CLEAR(strDSO)

vTEST = HS.GETCELL(strDEBTOR) * HS.GETCELL(strSALES&"".W#Periodic"") *
HS.GETCELL(strDIP&"".E#[None]"")

' checks if any of the parameters are zero (uses principle of X * 0 = 0)

IF vTEST = 0 THEN

EXIT SUB

ELSE

vDSO = 0
vCOUNT = 0
vXS_1 = HS.GETCELL(strDEBTOR)
vXS = vXS_1 - HS.GETCELL(strSALES&"".W#Periodic"")

' ensures that periodic sales are not negative or zero

WHILE vXS > 0 AND vXS_1 > vXS

vDSO = vDSO + HS.GETCELL(strDIP&"".E#[None].P#CUR-" " &vCOUNT)
vCOUNT = vCOUNT + 1
```

```
vXS_1 = vXS
vXS = vXS - HS.GETCELL(strSALES&"".W#Periodic.P#CUR-"" &vCOUNT)

WEND

IF vXS = vXS_1 THEN

vCOUNT = vCOUNT - 1

END IF

vDSO = vDSO + (vXS_1 / HS.GETCELL(strSALES&"".W#Periodic.P#CUR-""
&vCOUNT)*HS.GETCELL(strDIP&"".E#[None].P#CUR-"" &vCOUNT))

IF vDSO < 0 THEN

vDSO = 0

END IF

END IF

HS.EXP strDSO &""& vDSO

END SUB
```

## Opening

지정한 완전히 정의된 계정(계정/C1/C2/C3/C4/ICP)의 기초 값을 검색합니다.

### 반환 값

이 함수는 HS.EXP 함수의 일부로 사용할 정확한 표현식을 나타내는 문자열을 반환합니다.

### 구문

Opening (*PointOfView*, *View*)

표 12-11 Opening 함수 구문

매개변수	적합한 값
PointOfView	계정, 사용자정의1...4, ICP 멤버 등의 적합한 조합입니다(예: "A#CLOSE.C1#[None].I#[ICP Top]").
뷰	다음 값 중 하나여야 합니다. ""(큰따옴표) - 처리 중인 시나리오에 대해 정의된 기본 뷰(YTD 또는 주기적)를 기반으로 합니다. YTD - 시나리오에 대해 설정된 기본 뷰를 재정의하는 연 누계 옵션을 지정합니다. 주기적 - 시나리오에 대해 설정된 기본 뷰를 재정의하는 주기적 옵션을 지정합니다.

자세한 설명

이 함수는 지정한 계정의 기초 값을 계산합니다. 기초 값은 View 매개변수를 기준으로 다르게 파생됩니다.

- View 매개변수가 YTD이면 기초 값이 이전 연도의 마지막 기간에서 파생됩니다.
- View 매개변수가 주기적이면 기초 값이 현재 연도의 이전 기간에서 파생됩니다. 현재 기간이 연도의 첫 번째 기간이면 기초 값이 이전 연도의 마지막 기간에서 검색됩니다.
- View 매개변수가 공백(" ")이면 기초 값이 시나리오의 기본 데이터 뷰를 기반으로 합니다.

예

FA\_COST 계정은 Opening 함수에서 사용된 View 매개변수에 따라 2014년 1월, 2월 및 3월에 대해 다음 값을 반환합니다. 처리 중인 시나리오에 대해 설정된 기본 뷰는 YTD입니다.

표 12-12 Opening 함수 예

계정	Dec2013	Jan2014	Feb2014	Mar2014
A#FA_COST	900	1,200	1,100	1,500
Opening("A#FA_COST\A		900	900	900
Opening("A#FA_COST\A "YTD")		900	900	900
Opening("A#FA_COST\A "Periodic")		900	1,200	1,100

**샘플 스크립트**

```
' sample statement written in the calling routine

Sub Calculate()

Hs.Exp "A#Open_FA_Cost = " & Opening("A#FA_Cost", "YTD")

End Sub

' programming of the OPENING function

FUNCTION OPENING(strPOV,strVIEW)

IF strVIEW = "" THEN

strVIEW = HS.SCENARIO.DEFAULTVIEW ( "")

END IF

strPOV = UCASE(strPOV)
strVIEW = UCASE(strVIEW)

IF strVIEW = "PERIODIC" THEN

OPENING = strPOV & ".P#PRIOR"

ELSEIF strVIEW = "YTD" THEN

OPENING = strPOV & ".Y#PRIOR.P#LAST"

ELSE

EXIT FUNCTION

END IF

END FUNCTION
```

## Rate

상위 멤버와 하위 멤버 간의 상대 환율을 계산하고 이 값을 승수로 반환합니다.

### 반환 값

이 함수는 대체로 환산 섹션에서 HS.EXP 함수의 일부로 사용할 값을 반환합니다.

### 구문

Rate (*ExchangeRate*, *TriangulationCurrency*)

**표 12-13 Rate 함수 구문**

매개변수	적합한 값
ExchangeRate	사용자정의 또는 내부거래 차원을 참조하지 않고 계정 문자열로 지정된 "CurrencyRate" 유형의 기본 계정입니다(예: "A#EOP_RATE").
TriangulationCurrency	문자열 또는 큰따옴표(" ")로서 적합한 통화 레이블입니다. 통화를 지정할 때는 사용자정의 차원을 참조할 필요가 없습니다.

### 자세한 설명

- 이 함수는 상위 멤버와 하위 멤버 간의 상대 환율을 계산하고 이 값을 승수로 반환합니다. 값은 *TriangulationCurrency* 매개변수를 기준으로 계산됩니다.
- TriangulationCurrency* 매개변수가 적합한 통화 레이블이면 상호 환율이 이 통화를 기반으로 합니다.
- TriangulationCurrency* 매개변수가 공백(" ")이면 이 함수는 먼저 적합한 직접 환율을 검색하고, 직접 환율이 없는 경우 애플리케이션 통화에 대해 삼각 분할을 사용합니다.
- 환율 값을 찾을 수 없으면 함수에서 1이 반환됩니다.

이 표에서는 데이터 검색 방법과 검색 순서를 보여 줍니다. 순서는 (1)과 같이 괄호 안의 숫자로 표현됩니다. 각 경우에서 먼저 하위 엔티티에서 검색되고, 데이터를 찾을 수 없으면 "[None]" 엔티티에서 검색됩니다.

다음 표에서는 하위 멤버 또는 상위 멤버의 통화가 삼각 분할 통화와 같거나, 삼각 분할이 공백인 경우 애플리케이션 통화와 같습니다.

**표 12-14 환율 예 - 삼각 분할 통화가 같음**

사용자정의 2 차원 환율		사용자정의 1 차원 환율	
		1차 하위	상위
사용자정의 2 차원 환율	1차 하위		(2)
상위	(1)		

다음 표에서는 삼각 분할이 지정되었으며 하위 통화 또는 상위 통화와 다릅니다.

**표 12-15 환율 예 - 삼각 분할 통화가 서로 다름**

사용자정의 2 차원 환율		사용자정의 1 차원 환율	
		1차 하위	상위
사용자정의 2 차원 환율	1차 하위		삼각 분할 (2)
상위			
삼각 분할		(1)	

다음 표에서는 삼각 분할이 지정되지 않았으며 애플리케이션 통화가 하위 통화 및 상위 통화와 다릅니다.

**표 12-16 환율 예 - 삼각 분할이 지정되지 않음**

사용자정의 2 차원 환율		사용자정의 1 차원 환율	
		1차 하위	상위
사용자정의 2 차원 환율	1차 하위		애플리케이션 (4)
상위	(1)	(2)	
애플리케이션		(3)	

**예**

애플리케이션 통화가 유로이고, C2#EURO에 대해 [None] 엔티티에 입력된 다음 환율을 사용하여 프랑스 하위 엔티티를 미국 상위 엔티티로 환산해야 합니다.

**표 12-17 Rate 함수 예**

Rate	기초 환율	기말 환율
C1#FFR	0.16000	0.16500
C1#USD	1.15862	1.15785

다음 함수는 기초 잔액 계정에 상대 종료 환율과 기초 환율의 차액을 곱합니다. 이 함수는 로컬 및 애플리케이션 통화 간에 환산이 일치하지 않는 경우 이동 분석을 계산할 때 유용합니다.

```
HS.EXP "A#FXO = A#OPEN * (" & RATE("A#EOP_RATE"," ") & "-" & RATE("A#OPE_RATE"," ") & ")"
```

이전 예에서 하위 멤버에 대한 OPEN 계정의 값이 FFR 10,000,000이면 미국 상위 FXO 계정의 값은 USD 44,102 [10,000,000 \* (0.165 / 1.15785 - 0.16 / 1.15862)]가 됩니다.



## 샘플 스크립트

```
' sample statement written in the calling routine
```

```
SUB TRANSLATE()
```

```
HS.TRANS "A#FXO", "A#FXO", "A#EOP_RATE", ""  
HS.EXP "A#FXO = A#OPEN * (" & RATE("A#EOP_RATE", " ") & "-" &  
RATE("A#OPE_RATE", " ") &")"
```

```
END SUB
```

```
' programming of the RATE function
```

```
FUNCTION RATE(sRATE, sTRI)
```

```
DIM sCCUR, sPCUR, sACUR, bRET, retValue, s3rdCUR  
DIM i
```

```
sRATE = UCASE(sRATE)  
sTRI = UCASE(sTRI)  
sCCUR = UCASE(HS.ENTITY.DEFCURRENCY(""))  
sPCUR = UCASE(HS.VALUE.CURRENCY)
```

```
sACUR = UCASE(HS.APPSETTINGS.CURRENCY)
retValue = 0

' check whether there is a triangulation specified, or if
triangulation or application currencies are the same as either parent
or child and set up the select case

IF sTRI = sCCUR OR sTRI = sPCUR OR (sTRI = " " AND (sACUR = sCCUR OR
sACUR = sPCUR)) THEN

i = 1

ELSEIF sTRI <> " " THEN

i = 2

ELSE

i = 3

END IF

SELECT CASE i

CASE 1

' bRET is a boolean that returns true if data is found. First search
the child...
' ...then search the [None] entity

bRET = GETVALUECP(".V#<Entity Currency>",retValue,sRATE,sCCUR,sPCUR)

IF NOT bRET THEN

bRET = GETVALUECP(".E#[None]",retValue,sRATE,sCCUR,sPCUR)

END IF

CASE 2
```

```
' use a dynamic parameter name for ease of writing the triangulation
```

```
checks

s3rdCUR = sTRI

bRET = GETVALUE3(".V#<Entity Currency>",retValue,sRATE,sCCUR,SPCUR,s3rdCUR)

IF NOT bRET THEN

bRET = GETVALUE3(".E#[None]",retValue,sRATE,
sCCUR,SPCUR,s3rdCUR)

END IF

CASE 3

' this case is used when the 2nd parameter is blank and is the most complex.

' first check direct rates in the child..

' ... then check triangulation against application currency in the child

' then check direct rates in [None].

'... finally check triangulation in [None]

s3rdCUR = sACUR

bRET = GETVALUECP(".V#<Entity Currency>",retValue,sRATE,sCCUR,SPCUR)

IF NOT bRET THEN

bRET = GETVALUE3(".V#<Entity Currency>",retValue,sRATE,sCCUR,SPCUR,s3rdCUR)

IF NOT bRET THEN

bRET = GETVALUECP(".E#[None]",retValue,sRATE,sCCUR,SPCUR)

IF NOT bRET THEN
```

```
sRATE, sCCUR, sPCUR, s3rdCUR)

END IF

END IF

END IF

END SELECT

IF bRET THEN

RATE = retValue

ELSE

RATE = 1

END IF

END FUNCTION

FUNCTION GETVALUECP (sENTITY, sVALUE, sRATE, sCCUR, sPCUR)

' this sub-function is used when comparing direct rates between child
and parent

GETVALUECP = FALSE

' check if data exists for direct rate child to parent. If it does
return it.
' if no direct child to parent rate check for indirect parent to child
```

```
rate...
' return the inverse of the indirect rate.

IF HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & sPCUR & sENTITY) <> 0 THEN

sVALUE = CDBL(HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & sPCUR & sENTITY))

GETVALUECP = TRUE

ELSEIF HS.GETCELL(sRATE & ".C1#" & sPCUR & ".C2#" & sCCUR & sENTITY) <> 0
THEN

sVALUE = CDBL(1 / HS.GETCELL(sRATE & ".C1#" & sPCUR & ".C2#" & sCCUR &
sENTITY))

GETVALUECP = TRUE

END IF

END FUNCTION

FUNCTION GETVALUE3(sENTITY,sVALUE,sRATE,sCCUR,sPCUR,s3rdCUR)

' this sub-function is used when triangulating
' check if data exists for direct rate child to triangulation...
' ... if it does return the direct relative rate child to parent...
' if no direct child to triangulation rate check for indirect triangulation
```

```
to child rate...
' ... return the inverse of the indirect relative rates.

GETVALUE3 = FALSE

IF HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & s3rdCUR & sENTITY) <>
0 THEN

sVALUE = CDBL(HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & s3rdCUR &
sENTITY) / HS.GETCELL(sRATE & ".C1#" & sPCUR & ".C2#" & s3rdCUR &
sENTITY))

GETVALUE3 = TRUE

ELSEIF HS.GETCELL(sRATE & ".C1#" & s3rdCUR & ".C2#" & sCCUR & sENTITY)
<> 0 THEN

sVALUE = CDBL(HS.GETCELL(sRATE & ".C1#" & s3rdCUR & ".C2#" & sPCUR &
sENTITY) / HS.GETCELL(sRATE & ".C1#" & s3rdCUR & ".C2#" & sCCUR &
sENTITY))

GETVALUE3 = TRUE

END IF

END FUNCTION
```

## 비즈니스 규칙 함수

이 섹션에서는 비즈니스 규칙 사용자정의 함수를 보여 줍니다.

### Custom\_Alloc

이 함수는 Factor POV를 할당의 기초로 사용하여 Source POV를 Destination POV에 할당하고 할당된 총 금액을 Elimination POV에 역 게시하는 옵션을 제공합니다. 이 함수는 사용자정의 차원 할당용으로 설계되었습니다.

#### 반환 값

반환 값이 없습니다.

## 구문

```
Custom_Alloc(Destination, Source, Factor, FactorN, FactorD, Elimination)
```

**표 12-18 Custom\_Alloc 함수 구문**

매개변수	적합한 값
Destination	계정, ICP 및 사용자정의 1-4 멤버의 적합한 조합인 적합한 대상 POV입니다.
소스	차원 멤버의 적합한 조합인 적합한 소스 POV입니다. <i>Source</i> 는 할당할 금액입니다.
Factor	적합한 소스 POV입니다. <i>Factor</i> 는 할당 요소를 저장하는 데 사용되는 계정입니다.
FactorN	적합한 소스 POV입니다. <i>FactorN</i> 은 할당의 기초로 사용되는 분자 요소입니다.
FactorD	적합한 소스 POV입니다. <i>FactorD</i> 는 할당의 기초로 사용되는 분모 요소입니다.
Elimination	적합한 소스 POV입니다. <i>Elimination</i> 은 빈 문자열("")일 수 있으며, 이 경우 이 매개변수가 무시됩니다. <i>Elimination</i> 매개변수가 설정된 경우 <i>Destination POV</i> 에 게시된 금액에 -1을 곱하여 <i>Elimination POV</i> 에 게시합니다.

## 자세한 설명

이 함수는 Factor POV를 할당의 기초로 사용하여 Source POV를 Destination POV에 할당하고 할당된 총 금액을 Elimination POV에 역 게시하는 옵션을 제공합니다. 이 함수는 사용자정의 차원 할당용으로 설계되었습니다.

*Factor* 매개변수는 *FactorN*을 *FactorD*로 나눈 결과를 저장합니다. 요소에서 현재 엔티티보다 오래된 엔티티를 참조할 수 있게 하려면 이 매개변수가 필요합니다.

Source POV의 엔티티가 상위 멤버인 경우 하위 멤버 레벨에서 계산을 실행하기 전에 상위 멤버를 통합해야 합니다. 상위 통화가 하위 통화와 다른 경우 하위 멤버 레벨에서 계산을 실행하기 전에 관련된 모든 통화의 환산도 실행해야 합니다.

호출 루틴에서 변수를 설정하여 Custom\_Alloc 함수로 전달하는 것이 좋습니다. 이 함수는 Destination, Source, Factor, FactorN, FactorD 및 Elimination POV를 정의합니다. 또한 호출 루틴의 변수 이름을 Custom\_Alloc 함수와 동일하게 설정하는 것이 좋습니다.

*Elimination* 매개변수는 빈 문자열("")일 수 있으며, 이 경우 이 매개변수가 무시됩니다. *Elimination* 매개변수가 설정된 경우 Destination POV에 게시된 금액에 -1을 곱하여 Elimination POV에 게시합니다.

## 예

총 판매에 대한 제품 판매의 비율을 기준으로 제품에 Telephone 계정이 할당됩니다. 할당된 금액의 역수가 Allocations 계정에 게시됩니다.

표 12-19 Custom\_Alloc 함수 예

계정	Jan2014	Feb2014	Mar2014
A#Telephone.C1#[None]	100	300	400
A#Sales.C1#Product1	1000	1000	1000
A#Sales.C1#Product2	1000	2000	3000
A#Sales.C1#TotalProducts	2000	3000	4000
Custom_Alloc (A#Telephone.C1#[None], "A#Telephone.C1#[None]", "A#Factor", A#Sales.C1#TotalProducts, "A#Sales.C1#Product1", "A#Sales.C1#Product2", "A#ProductAllocations.C1#[None]")	N/A	N/A	N/A
A#Factor.C1#Product1	0.50	0.33	0.25
A#Factor.C1#Product2	0.50	0.66	0.75
A#Telephone.C1#Product1	50	100	100
A#Telephone.C1#Product2	50	200	300
A#ProductAllocations.C1#[None]	100	-300	-400

CUSTOM\_ALLOC 함수에서 반환되는 결과는 다음과 같습니다.

```
HS.EXP "A#Factor = A#Sales / A#Sales.C1#TotalProducts"
```

```
HS.EXP "A#Telephone = A#Telephone.C1#[None] * A#Factor"
```

```
HS.EXP "A#Allocations.C1#[None] = (A#Telephone.C1#[None] * -1)"
```

### 샘플 스크립트

이 스크립트에는 다음 정보가 들어 있습니다.

- 호출 루틴 내에 작성된 샘플 명령문
- 호출 루틴에서 설정되어 Custom\_Alloc 함수로 전달된 변수
- Custom\_Alloc 함수와 동일하게 설정된 호출 루틴의 변수 이름

```
Sub Calculate()
```

```
Dim Destination
Dim Source
Dim Elimination
Dim Factor
Dim FactorN
Dim FactorD
```



```
Dim C1list
Dim Clitem

C1list = HS.Custom1.List("Alloc")

For Each Clitem in C1list

Source = "A#Telephone.C1#[None]"
Destination = "A#Telephone.C1#" & Clitem
Factor = "A#Factor.C1#" & Clitem
FactorN = "A#Sales.C1#" & Clitem
FactorD = "A#Sales.C1#TotalProducts"
Elimination = "A#ProductAllocations.C1#" & Clitem

Call Custom_Alloc(Destination,Source,Factor,FactorN,
FactorD,Elimination)

Next

End Sub

' Beginning of the Custom_Alloc function

Sub Custom_Alloc(Destination,Source,FactorN,FactorD,
Elimination)

HS.Clear Factor

HS.Exp Factor & " = " & FactorN & "/" & FactorD

HS.EXP Destination & " = " & Source & " * " & Factor

If Elimination <> "" Then

HS.EXP Elimination & " = " & Source & " * -1 * " & Factor

End If

End Sub
```

## Increase\_Decrease

이 함수는 Destination POV를 백분율 Factor만큼 늘리거나 줄입니다. 백분율 Factor는 Source POV, VBScript 상수 또는 VBScript 변수에서 가져올 수 있습니다.

### 반환 값

반환 값이 없습니다.

### 구문

```
Increase_Decrease(Destination, Source, Factor, Scale, Inverse)
```

**표 12-20 Increase\_Decrease 함수 구문**

매개변수	적합한 값
Destination	계정, ICP 및 사용자정의 1-4 멤버의 적합한 조합인 적합한 대상 POV입니다.
소스	차원 멤버의 적합한 조합인 적합한 소스 POV입니다. <i>Source</i> 는 할당할 금액입니다.
Factor	적합한 소스 POV, 상수 또는 변수입니다.
Scale	정수 값 1 또는 100입니다. Factor를 Scale로 나눕니다.
Inverse	True 또는 False입니다. True이면 Factor의 부호가 반대로 바뀝니다. 이 매개변수를 사용하여 Factor가 양수로 저장되는 감소(또는 그 반대)를 생성할 수 있습니다. False이면 Factor의 저장된 부호가 증가 또는 감소를 결정하는 데 사용됩니다.

### 자세한 설명

이 함수는 Destination POV를 백분율 Factor만큼 늘리거나 줄입니다. 백분율 Factor는 Source POV, VBScript 상수 또는 VBScript 변수에서 가져올 수 있습니다.

일반적으로 Source POV는 Destination POV와 같지만 다를 수도 있습니다.

*Scale* 매개변수는 필요한 경우 요소의 스케일을 줄이는 데 사용됩니다. 이 매개변수는 Source POV에서 요소를 가져오고 스케일되지 않은 형태(예: 50%가 0.50이 아니라 50으로 저장됨)로 저장하는 경우에 적용됩니다.

*Inverse* 매개변수는 요소의 부호를 반대로 바꾸는 데 사용됩니다. 이 매개변수는 Source POV에서 요소를 가져오고 절대 수로 저장하는 경우에 적용됩니다. *Inverse* 매개변수가 True로 설정된 경우 요소에 -1을 곱합니다. *Inverse* 매개변수가 false로 설정된 경우에는 요소에 -1을 곱하지 않습니다.

### 예

이 예에서는 Telephone 계정이 10%만큼 증가합니다.

표 12-21 Increase\_Decrease 함수 예

계정	Jan2014	Feb2014	Mar2014
A#Telephone	100	300	400
A#Factor/C1[None]	10	10	10
	N/A	N/A	N/A
Increase_Decrease("A#Telephone", "A#Telephone", "A#Factor.C1#[None]",100,False)			
A#Telephone	110	330	440

INCREASE\_DECREASE 함수에서 반환되는 결과는 다음과 같습니다.

```
HS.EXP "A#Telephone = A#Telephone * (1+ (A#Factor.C1#[None]/100))"
```

#### 샘플 스크립트

- 호출 루틴 내에 작성된 샘플 명령문
- 호출 루틴에서 설정되어 Increase\_Decrease 함수로 전달된 변수
- Increase\_Decrease 함수와 동일하게 설정된 호출 루틴의 변수 이름

```
Sub Calculate()

    Dim Destination
    Dim Source
    Dim Factor
    Dim Scale
    Dim Inverse

    Destination = "A#Telephone"
    Source = "A#Telephone"
    Factor = "A#Factor.C1#[None]"
```

```
Scale = "100"
Inverse = False

Call Increase_Decrease(Destination,Source,Factor,Scale,
Inverse)

End Sub

' Beginning of the Increase_Decrease function

Sub Increase_Decrease(Destination,Source,Factor,Scale,Inverse)

If Inverse = False Then

HS.EXP Destination & " = " & Source & " *

(1 + (" & Factor & " / " & Scale & "))"

Else

HS.EXP Destination & " = " & Source & " *

(1 + ((" & Factor & " * -1) / " & Scale & ))"

End If

End Sub
```

## Pro\_Rata\_Ratio

이 함수는 두 Source POV 간의 비율을 계산합니다( $C = A / B$ ).

### 반환 값

반환 값이 없습니다.

### 구문

```
Pro_Rata_Ratio(Destination,SourceN,SourceD)
```

**표 12-22 Pro\_Rata\_Ratio 함수 구문**

매개변수	적합한 값
Destination	계정, ICP 및 사용자정의 1-4 멤버의 적합한 조합인 적합한 대상 POV입니다.
SourceN	차원 멤버의 적합한 조합인 적합한 소스 POV입니다. <i>SourceN</i> 은 비율 계산의 분자입니다.
SourceD	적합한 소스 POV입니다. <i>SourceD</i> 는 비율 계산의 분모입니다.

**자세한 설명**

이 함수는 두 Source POV 간의 비율을 계산합니다( $C = A / B$ ).

호출 루틴에서 변수를 설정하여 Pro\_Rata\_Ratio 함수로 전달하는 것이 좋습니다. 이 함수는 Destination, SourceN 및 SourceD POV를 정의합니다. 또한 호출 루틴의 변수 이름을 Pro\_Rata\_Ratio 함수와 동일하게 설정하는 것이 좋습니다.

상위 멤버에 대해 기본적으로 가중 평균 비율이 계산되지 않습니다. 상위 멤버 값은 하위 멤버 값의 합산으로 표시됩니다. 이 경우 상위 멤버에 대해 수학적으로 부정확한 값이 생성됩니다. 따라서 Ratio 계정에 대해 합산을 해제하는 것이 좋습니다.

**예**

MarginPct 계정은 GrossMargin/TotalRevenues 값을 반환합니다.

**표 12-23 Pro\_Rata\_Ratio 함수 예**

계정	Jan2014	Feb2014	Mar2014
A#GrossMargin	1000	100	750
A#TotalRevenues	2000	400	1000
	0.50	0.25	0.75
Pro_Rata_Ratio("A#GrossMargin", "#TotalRevenues")			

PRO\_RATA\_RATIO 함수에서 반환되는 결과는 다음과 같습니다.

```
HS.EXP "A#MarginPct = A#GrossMargin / A# TotalRevenues"
```

**샘플 스크립트**

이 스크립트에는 다음 정보가 들어 있습니다.

- 호출 루틴 내에 작성된 샘플 명령문
- 호출 루틴에서 설정되어 Pro\_Rata\_Ratio 함수로 전달된 변수

- Pro\_Rata\_Ratio 함수와 동일하게 설정된 호출 루틴의 변수 이름

```

Sub Calculate()

Dim Destination 'Destination POV
Dim SourceN     'Source Numerator POV
Dim SourceD     'Source Denominator POV

Destination = "A#MarginPct"
SourceN     = "A#GrossMargin"
SourceD     = "A#TotalRevenues    "

Call Pro_Rata_Ratio(Destination,SourceN,SourceD)

End Sub

' Beginning of the Pro_Rata_Ratio function

Sub Pro_Rata_Ratio(Destination,SourceN,SourceD)

HS.EXP Destination & " = " & SourceN & " / " & SourceD

End Sub

```

## Spread

이 함수는 프로파일 계정에 정의된 프로파일(예: 수익 프로파일, 4-4-5 등)을 기준으로 소스 계정의 단일 기간 값(예: P#[Year])을 대상 계정의 모든 기간에 할당합니다.

### 반환 값

반환 값이 없습니다.

### 구문

```
Spread(Destination,Source,Factor,FactorN,FactorD,Temp,Per)
```

**표 12-24 Spread 함수 구문**

매개변수	적합한 값
Destination	계정, ICP 및 사용자정의 1-4 멤버의 적합한 조합인 적합한 대상 POV입니다.

표 12-24 (계속) Spread 함수 구문

매개변수	적합한 값
소스	차원 멤버의 적합한 조합인 적합한 소스 POV입니다. Source POV에는 단일 기간이 포함되어야 합니다(예: P#[Year]). 단일 기간 금액은 분산할 금액입니다.
Factor	적합한 소스 POV입니다. Factor는 할당 요소를 저장하는 데 사용되는 계정입니다.
FactorN	적합한 소스 POV입니다. FactorN은 분산 할당의 기초로 사용되는 분자 요소입니다.
FactorD	적합한 소스 POV입니다. FactorD는 분산 할당의 기초로 사용되는 분모 요소입니다.
Temp	적합한 대상 계정입니다. Temp는 일시적으로 Source 값을 저장하는 계정입니다.
Per	시간 프레임에서 첫 번째 기간의 이름을 정의하는 기간 문자열(예: "1월")입니다. Temp 값은 첫 번째 기간에 저장되며 계산에서 이 값을 참조하려면 해당 매개변수가 필요합니다.

### 자세한 설명

이 함수는 Profile POV에 정의된 프로파일(예: 수익 프로파일, 4-4-5 등)을 기준으로 Source POV의 단일 기간 값(예: P#[Year])을 Destination POV의 모든 기간에 할당합니다.

시간 기반 할당은 특히 예산 편성 애플리케이션에 적합합니다. 이 경우 먼저 전체 연도에 대해 금액이 입력된 다음 적합한 프로파일을 기준으로 나중에 금액이 기간에 할당됩니다.

Source POV에는 단일 기간이 포함되어야 합니다. 기간은 일반적으로 P#[Year]이지만 P#January 등의 단일 기간일 수도 있습니다.

Source POV의 값은 계산 시 임시 계정에 저장됩니다. 임시 계정은 일반적으로 소스 및 대상 계정이 동일한 계정이기 때문에 필요합니다. 이 경우 계산이 1 기간에서 다음 기간으로 진행되면 P#[Year]의 값이 변경됩니다. 따라서 모든 기간에 대해 값을 참조할 수 있으려면 먼저 값을 저장해야 합니다.

호출 루틴에서 변수를 설정하여 Spread 함수로 전달하는 것이 좋습니다. 이 함수는 Destination, Source, Profile, Temp 및 Period1 매개변수를 정의합니다. 또한 호출 루틴의 변수 이름을 Spread 함수와 동일하게 설정하는 것이 좋습니다.

### 예

Telephone 계정의 Year 값은 4-4-5 분기별 비율을 사용하여 전체 기간에 할당됩니다.

SPREAD 함수에서 반환되는 결과는 다음과 같습니다.

```
HS.EXP "A#TempTelephone.C1#[None] = A#Telephone.C1#[None].P#[Year]" (Where
Period.Number = 1)
```

```
HS.EXP "A#Telephone.C1#[None] = A#TempTelephone P#January *
E.Globals.A#Profile445.C1#[None].P#Cur /
E.Globals.A#Profile445.C1#[None].P#[Year]"
```

### 샘플 스크립트

이 스크립트에는 다음 정보가 들어 있습니다.

- 호출 루틴 내에 작성된 샘플 명령문
- 호출 루틴에서 설정되어 Spread 함수로 전달된 변수
- Spread 함수와 동일하게 설정된 호출 루틴의 변수 이름

```
Sub Calculate()
```

```
Dim Destination  
Dim Source  
Dim Factor  
Dim FactorN  
Dim FactorD  
Dim Temp  
Dim Per
```

```
Source = "A#Telephone.C1#[None].P#[Year]"  
Destination = "A#Telephone.C1#[None]"  
Factor = "A#Factor.C1#[None]"  
FactorN = "E#Globals.A#Profile445.C1#[None].P#CUR"  
FactorD = "E#Globals.A#Profile445.C1#[None].P#[Year]"
```



```
Temp = "A#TempTelephone.C1#[None]"
Per = "January"

Call Spread(Destination,Source,Factor,
FactorN,FactorD,Temp,Per)

End Sub

' Beginning of the Spread function

Sub Spread(Destination,Source,Factor,FactorN,FactorD,Temp,Per)

If HS.Period.Number = 1 Then

HS.Exp Temp & " = " & Source

End If

HS.Clear Factor
HS.EXP Factor & " = " & FactorN & " / " & FactorD

HS.Clear Destination
HS.EXP Destination & " = " & Temp & ".P#" & Per & " * " & Factor

End Sub
```

## Units\_Rates

이 함수는 두 Source POV의 곱을 계산합니다( $C = A * B$ ).

### 반환 값

반환 값이 없습니다.

### 구문

```
Units_Rates(Destination,Units,Rates)
```

**표 12-25 Units\_Rates 함수 구문**

매개변수	적합한 값
Destination	계정, ICP 및 사용자정의 1-4 멤버의 적합한 조합인 적합한 대상 POV입니다.
Units	차원 멤버의 적합한 조합인 적합한 소스 POV입니다.
Rates	적합한 소스 POV입니다.

**자세한 설명**

이 함수는 두 Source POV의 곱을 계산합니다( $C = A * B$ ). 호출 루틴에서 변수를 설정하여 Units\_Rates 함수로 전달하는 것이 좋습니다. 이 함수는 Destination, Units 및 Rates POV를 정의합니다. 또한 호출 루틴의 변수 이름을 Units\_Rates 함수와 동일하게 설정하는 것이 좋습니다.

**예**

Sales 계정은 UnitsSold \* Price 값을 반환합니다.

**표 12-26 판매 단위 함수 예**

계정	Jan2014	Feb2014	Mar2014
A#UnitsSold	1000	2000	5000
A#Price	1.25	1.00	0.50
	1250	2000	2500
Units_Rates("A#UnitsSold",A#Price)			

UNITS\_RATES 함수에서 반환되는 결과는 다음과 같습니다.

```
HS.EXP "A#Sales = A#UnitsSold * A#Price"
```

**샘플 스크립트**

이 스크립트에는 다음 정보가 들어 있습니다.

- 호출 루틴 내에 작성된 샘플 명령문
- 호출 루틴에서 설정되어 Units\_Rates 함수로 전달된 변수

- Units\_Rates 함수와 동일하게 설정된 호출 루틴의 변수 이름

```
Sub Calculate()  
  
Dim Destination  
Dim Units  
Dim Rates  
  
Destination = "A#Sales"  
Units = "A#UnitsSold"  
Rates = "A#Price"  
  
Call Units_Rates(Destination,Units,Rates)  
  
End Sub  
  
' Beginning of the Units_Rates function  
  
Sub Units_Rates(Destination,Units,Rates)  
  
HS.EXP Destination & " = " & Units & " * " & Rates  
  
End Sub
```

## Calculation Manager를 사용하여 규칙 생성

Oracle Hyperion Calculation Manager 모듈은 Oracle Hyperion Financial Management에 대한 계산 규칙을 생성할 공통 사용자 인터페이스를 제공합니다. 그래픽 플로우를 통해 계산 과정을 더 잘 이해할 수 있으며 그래픽 뷰와 VB 스크립트 뷰를 전환할 수 있습니다. Calculation Manager에서는 모든 계산 규칙을 유지 관리하고 애플리케이션 간에 규칙을 공유하기 위해 중앙 저장소를 제공합니다. 계산 규칙을 가져오고, 내보내고, 인쇄할 수 있으며 네비게이션이 용이하도록 사용자정의 폴더를 생성할 수 있습니다.

### Calculation Manager 보안 역할

이러한 역할은 Oracle Hyperion Financial Management의 Oracle Hyperion Calculation Manager 액세스에 사용할 수 있습니다.

- 규칙 관리자 - Calculation Manager에서 지정된 애플리케이션에 대해 규칙 객체, 템플릿 및 변수 생성, 수정, 삭제 및 규칙 세트 검증, 배포와 같은 태스크를 수행할 수 있습니다.
- 규칙 디자이너 - 규칙 객체를 생성하고 이러한 객체를 수정 또는 삭제할 수 있습니다.
- 규칙 조회자 - 규칙 객체를 표시 및 검증할 수 있습니다.

Oracle Hyperion Enterprise Performance Management Workspace에서 Calculation Manager에 액세스하려면 규칙 관리자, 규칙 디자이너 또는 규칙 조회자 보안 역할이 있어야 합니다.

규칙을 배포하려면 규칙 관리자 보안 역할이 있어야 합니다.

보안 역할에 대한 자세한 내용은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

### Calculation Manager의 애플리케이션 작업

애플리케이션을 생성하는 경우 VBScript 규칙을 로드하거나 Oracle Hyperion Calculation Manager를 사용하여 규칙을 디자인하고 Oracle Hyperion Financial Management 애플리케이션에 배포할 수 있습니다.

#### 주:

Calculation Manager에 규칙을 배포한 후 VBScript 규칙을 로드하려고 하면 Calculation Manager 규칙을 덮어쓴다는 메시지가 표시됩니다. [확인]을 눌러 계속하거나 [취소]를 누를 수 있습니다.

Calculation Manager의 통합 폴더를 열면 애플리케이션 목록이 영문자순으로 표시됩니다.

애플리케이션 폴더를 확장하여 규칙 세트, 규칙, 공식, 스크립트 및 템플릿을 볼 수 있습니다. Calculation Manager 사용에 대한 자세한 내용은 *Oracle Hyperion Calculation Manager 디자이너 가이드* 또는 온라인 도움말을 참조하십시오.

## 규칙을 Calculation Manager로 마이그레이션

이전 릴리스의 기존 VBScript 규칙(.rle) 파일이 있는 경우 해당 파일을 임포트하여 Oracle Hyperion Calculation Manager로 마이그레이션할 수 있습니다. *Oracle Hyperion Calculation Manager 디자이너 가이드*의 "규칙 마이그레이션" 섹션을 참조하십시오.

## 함수 선택기의 VB 함수 지원

Oracle Hyperion Calculation Manager 함수 선택기에서는 다음과 같은 VB 함수를 지원합니다.



주:

UI에서는 선택할 수 없지만 스크립트 구성요소에서 다른 VB 함수를 사용할 수도 있습니다.

### 배열 함수

함수	설명
배열	배열이 포함된 변수를 반환합니다.
필터	필터 기준에 따라 문자열 배열의 서브셋이 포함된 0 기반 배열을 반환합니다.
조인	배열에 있는 많은 하위 문자열로 구성된 문자열을 반환합니다.
LBound	배열의 지정된 차원에 사용할 수 있는 가장 낮은 아래 첨자를 반환합니다.
분할	지정된 개수의 하위 문자열이 포함된 0 기반 1차원 배열을 반환합니다.
UBound	배열의 지정된 차원에 사용할 수 있는 가장 높은 아래 첨자를 반환합니다.

### 날짜 및 시간 함수

함수	설명
Date	현재 시스템 날짜를 반환합니다.
DateAdd	지정된 시간 간격이 추가된 날짜를 반환합니다.
DateDiff	두 날짜 사이의 간격 수를 반환합니다.
DatePart	지정된 날짜의 지정된 부분을 반환합니다.
DateSerial	지정된 연도, 월, 일의 날짜를 반환합니다.
Day	일을 나타내는 1에서 31(포함) 사이의 숫자를 반환합니다.
Month	월을 나타내는 1에서 12(포함) 사이의 숫자를 반환합니다.

함수	설명
MonthName	지정된 월의 이름을 반환합니다.

수학

함수	설명
Abs	지정된 수의 절대값을 반환합니다.
Fix	지정된 수의 정수 부분을 반환합니다.
Int	지정된 수의 정수 부분을 반환합니다.

String

함수	설명
InStr	다른 문자열 내에서 한 문자열의 첫 번째 발생 위치를 반환합니다. 문자열의 첫 문자에서 검색이 시작됩니다.
InStrRev	다른 문자열 내에서 한 문자열의 첫 번째 발생 위치를 반환합니다. 문자열의 첫 문자에서 검색이 시작됩니다.
LCase	지정된 문자열을 소문자로 변환합니다.
Left	문자열 왼쪽에서 지정된 개수의 문자를 반환합니다.
Len	문자열의 문자 수를 반환합니다.
Mid	문자열에서 지정된 개수의 문자를 반환합니다.
Right	문자열 오른쪽에서 지정된 개수의 문자를 반환합니다.
StrComp	두 문자열을 비교하고 비교 결과를 나타내는 값을 반환합니다.
Trim	문자열의 왼쪽과 오른쪽에서 모두 공백을 제거합니다.
UCase	지정된 문자열을 대문자로 변환합니다.

## Financial Management에 대한 특수 VB 스크립트 함수

이러한 특수 함수는 Oracle Hyperion Financial Management에서 Oracle Hyperion Calculation Manager UI의 배열과 루프를 처리하기 위해 구현되었습니다.

### Range

값	루프 변수	VBScript 생성
@Range(1-50)	i	Dim i(50) I(1)=1 I(2)=2 I(3)=3 .. I(50)=50

값	루프 변수	VBScript 생성
@Range(5-10)	i	Dim i(6) I(1)=5 I(2)=6 I(3)=7 I(4)=8 I(5)=9 I(6)=10
@Range(1,3-5,7-9)	i	Dim i(8) I(1)=1 I(2)=3 I(3)=4 I(4)=5 I(5)=7 I(6)=8 I(7)=9

### For/ForStep

값	루프 변수	VBScript 생성	설명
@For(2,10)	Item	For Item=2 to 10	새로운 @ForLoop @For(from, to)
@ForStep(2,10,2)	Item	For Item =2 to 10 step 2	새로운 @ForStep 루프 함수 @ForStep(from, to, step) 역 단계가 필요한 경우 단계 앞에 음수 기호(-)를 추가합니다. 예: @ForStep(2,10,-2)

### ExitFor

@ExitFor - 루프 종료

다음과 같은 VB 스크립트 명령문이 생성됩니다.

For each *element* in group

[*statements*]

### Exit For

*statements*]

Next [*element*]

또는

For *counter*=start To end [Step *step*]

[*statements*]

### Exit For

[statements]

Next [counter]

### ExitSub

@ExitSub - 규칙 종료

다음과 같은 VB 스크립트 명령문이 생성됩니다.

Sub name [(x,y)]

statements]

### ExitSub

[statements]

EndSub

### ReDim

하나 이상의 동적 배열 변수 차원을 변경하고 저장영역 공간을 재할당합니다. 선택적 Preserve 키워드를 사용하여 차원을 변경할 때 배열 내용을 그대로 유지할 수 있습니다.

{VarArrayX(5)} = @Redim

{VarArrayXY(5,9)} = @Redim

다음과 같은 VB 명령문이 생성됩니다.

Redim VarArrayX(5)

Redim VarArrayXY(5,9))

### RedimPreserve

{VarArrayXY(5)} = @RedimPreserve

또는

{VarArrayXY(5,9)} = @RedimPreserve

또는

{VarArrayXY(5,{i})} = @RedimPreserve

다음과 같은 VB 명령문이 생성됩니다.

RedimPreserve VarArrayX(5)

또는

RedimPreserve VarArrayXY(5,9)

또는

RedimPreserve VarArrayXY(5,i)



# 14

## 내부거래 트랜잭션 관리

### 참조:

- [내부거래 트랜잭션 설정](#)
- [원인 코드 관리](#)
- [내부거래 트랜잭션 모니터](#)
- [엔티티 잠금 및 잠금 해제](#)
- [내부거래 트랜잭션 요약 보기](#)

## 내부거래 트랜잭션 설정

내부거래 트랜잭션은 조직의 두 엔티티 간에 수행되는 트랜잭션입니다. Oracle Hyperion Financial Management를 사용하면 계정과 사용자정의 차원 전체에서 내부거래 트랜잭션 세부정보를 추적하고 조정할 수 있습니다. 내부거래 트랜잭션 모듈에서는 내부거래 계정 차이를 효율적으로 식별, 보고 및 조정하는 방법을 제공합니다.

[내부거래 트랜잭션 관리] 페이지를 사용하여 내부거래 트랜잭션을 처리합니다. 트랜잭션 처리에 대한 자세한 내용은 *Oracle Hyperion Financial Management 사용자 가이드*를 참조하십시오.

내부거래 트랜잭션을 입력하려면 다음 설정 절차를 먼저 완료해야 합니다.

- 내부거래 트랜잭션에 대한 기간을 엽니다. [내부거래 기간 열기](#)를 참조하십시오.
- 내부거래 트랜잭션 규칙을 정의하고 애플리케이션에 로드합니다. HS.SupportsTran 함수는 내부거래 트랜잭션을 지원하는 애플리케이션에서 계정, 시나리오 및 사용자정의 차원을 지정합니다. [규칙 파일 생성 및 규칙 로드](#)를 참조하십시오.
- 일치 허용오차를 결정합니다. [일치 허용오차 설정](#)을 참조하십시오.
- 트랜잭션이 [일치하지 않음] 상태인 이유를 나타내는 원인 코드를 정의합니다. [원인 코드 추가](#)를 참조하십시오.
- 통화 환율을 입력합니다. [통화 정의](#)를 참조하십시오.

## 내부거래 기간 열기

내부거래 트랜잭션을 입력, 로드 또는 처리하려면 먼저 트랜잭션 기간을 열어야 합니다. 기간에는 [열리지 않음], [열림] 또는 [닫힘] 상태가 지정될 수 있습니다. 기간의 기본 상태는 [열리지 않음]입니다. 기간을 열고 트랜잭션을 입력한 후에만 [닫힘]으로 변경할 수 있습니다. [열리지 않음]으로 돌아갈 수는 없습니다.

각 기간에 대해 [게시 전에 일치/검증을 수행하시겠습니까?] 옵션을 설정하고 [자동 일치] 및 [수동 일치] 프로세스에 적용되는 일치 허용 오차를 지정할 수 있습니다. [게시 전에 일치/검증을 수행하시겠습니까? 옵션 설정 및 일치 허용오차 설정](#)을 참조하십시오.

내부거래 기간을 열려면 [애플리케이션 관리자] 보안 역할이 있어야 합니다.

기간을 열려면 다음을 수행합니다.

1. **통합, 내부거래, 기간** 순으로 선택합니다.

2. 시나리오에서 기간의 시나리오를 선택합니다.
3. 연도에서 기간의 연도를 선택합니다.
4. 열려는 각 기간을 선택합니다.
5. **선택 사항:** 기간에 대한 트랜잭션 ID 허용 오차 금액이나 백분율, 계정 허용 오차 금액 또는 수동 허용 오차 금액을 입력합니다.
6. **게시 전에 일치/검증을 수행하시겠습니까?** 열에서 다음 옵션 중 하나를 선택합니다.
  - 시스템에서 트랜잭션을 게시하기 전에 일치 상태를 확인해야 하는 경우 **예** 또는 **제한**을 선택합니다.
  - 시스템에서 일치 상태를 검사할 필요가 없는 경우 **아니요**를 선택합니다.
7. **기간 열기를 누르거나 작업, 기간 열기** 순으로 선택합니다.  
기간의 현재 상태가 [열림]으로 변경됩니다.
8. **선택사항:** 기간에 대한 설정을 저장하려면 **기간 설정 저장**을 누르거나 **작업, 기간 설정 저장** 순으로 선택합니다.



**팁:**

변경사항을 저장하지 않고 기간을 원래 상태로 복원하려면 **복원**을 누르거나 **작업, 복원** 순으로 선택합니다.

## 일치 허용오차 설정

자동 일치 및 수동 일치 프로세스에 대해 기간별 일치 허용 오차를 설정할 수 있습니다. 계정 및 수동 일치 허용 오차 금액을 설정할 수 있습니다. 트랜잭션 ID(TID) 허용 오차의 경우 금액, 백분율 또는 둘 다를 지정할 수 있습니다.

트랜잭션 ID에 대해 백분율을 입력하면 엔티티 트랜잭션 합계와 파트너 트랜잭션 합계 사이의 더 적은 금액이 사용되며 백분율이 금액에 적용되어 허용 오차 금액을 생성합니다.

예를 들어 TID가 123인 엔티티 A에서의 트랜잭션 세 개가 있고 이러한 트랜잭션의 합계가 1000이라고 가정합니다. TID가 123인 엔티티 B에는 5개의 트랜잭션이 있고 합계가 1020입니다. 엔티티 합계와 파트너 합계 사이의 차이는 20입니다. 하지만 허용 오차를 3%로 지정하면 더 적은 합계의 3%가 계산되므로(1000 \* 3%) 30이 생성됩니다. 이 값을 차이와 비교하면 허용 오차 내에 있으므로 트랜잭션이 일치하는 것으로 간주됩니다.

백분율과 함께 금액(예: 15)을 입력하면 백분율 금액과 입력된 금액을 비교하여 가장 작은 금액이 비교로 사용됩니다. 이 예에서 엔티티 합계와 파트너 합계의 차이는 20이고 백분율 허용 오차는 30이지만 금액 허용 오차는 15입니다. 이 값은 허용 오차 내에 있지 않으므로 트랜잭션이 일치하지 않습니다.

TID 금액과 백분율을 비워 둘 수도 있습니다. 두 값 중 하나가 0이거나 둘 다 비어 있으면 트랜잭션 차이가 0인 트랜잭션만 일치합니다.

계정 허용 오차 및 수동 일치 허용 오차의 경우 일치 허용 오차는 애플리케이션 통화로 표시되고 입력한 금액은 애플리케이션 통화의 스케일 요소로 표시됩니다. 일치 프로세스에서 시스템은 각 트랜잭션을 애플리케이션 통화로 변환하고 합계 차이 금액을 기간에 설정된 일치 허용 오차와 비교합니다. 비교는 단위로 수행됩니다.

TID/RID 일치 프로세스에서 TID 허용 오차 값과 비교할 때 트랜잭션이 애플리케이션 통화로 환산되지는 않습니다.

## 게시 전에 일치/검증을 수행하시겠습니까? 옵션 설정

각 기간에 대해 [게시 전에 일치/검증을 수행하시겠습니까?] 옵션을 설정할 수 있습니다. [일치/검증] 옵션은 시스템에서 트랜잭션이 게시되기 전에 트랜잭션의 일치 상태를 검사해야 하는지 여부를 정의하고 엔티티가 잠기거나 기간이 닫히기 전에 수행해야 하는 검증 유형을 정의합니다.

[게시 전에 일치/검증] 옵션을 선택한 경우 게시하려면 먼저 트랜잭션을 일치시키거나 원인 코드를 지정해야 합니다. 기간을 닫거나 엔티티를 잠그려면 먼저 원인 코드와 함께 모든 일치 트랜잭션 또는 불일치 트랜잭션을 게시해야 합니다.

[제한] 옵션을 선택한 경우 트랜잭션을 먼저 일치시켜야 게시할 수 있지만, 게시되지 않은 트랜잭션이 있는 엔티티를 잠그거나 기간을 닫을 수 있습니다.

**표 14-1** 게시 전에 일치/검증을 수행하시겠습니까? 옵션 값 설명

일치/검증 옵션 값	설명
아니요	모든 트랜잭션을 게시할 수 있습니다.
예	다음 조건 중 하나가 충족되면 트랜잭션을 게시할 수 있습니다. <ul style="list-style-type: none"> <li>트랜잭션이 [일치함] 상태인 경우</li> <li>[일치하지 않음] 상태의 트랜잭션에 적합한 원인 코드가 포함되어 있는 경우</li> </ul> 원인 코드가 없는 대응 취소된 트랜잭션이나 일치하지 않는 트랜잭션은 게시할 수 없습니다. 또한 시스템에서 기간이 닫히거나 엔티티가 잠기기 전에 일치하는 트랜잭션이나 원인 코드가 포함된 일치하지 않는 트랜잭션이 모두 게시되었는지 검사합니다.
제한	[게시 전에 일치/검증을 수행하시겠습니까?] 옵션을 [제한]으로 설정한 경우 다음 조건 중 하나가 충족되면 트랜잭션을 게시할 수 있습니다. <ul style="list-style-type: none"> <li>트랜잭션이 [일치함] 상태인 경우</li> <li>[일치하지 않음] 상태의 트랜잭션에 적합한 원인 코드가 포함되어 있는 경우</li> </ul> 원인 코드가 없는 대응 취소된 트랜잭션이나 일치하지 않는 트랜잭션은 게시할 수 없습니다. 게시되지 않은 트랜잭션이 있는 엔티티를 잠그거나 기간을 닫을 수 있습니다.

## 내부거래 기간 닫기

내부거래 트랜잭션을 처리한 후 기간을 닫아 트랜잭션이 수정되지 않도록 할 수 있습니다. [게시 전에 일치/검증] 옵션을 선택한 경우 원인 코드와 함께 모든 일치 트랜잭션과 불일치 트랜잭션을 게시해야 합니다. 기간이 이후 트랜잭션에 대해 잠긴 경우에도 해당 기간에 대한 트랜잭션을 보고 보고서를 실행할 수 있습니다.

내부거래 기간을 닫으려면 [애플리케이션 관리자] 보안 역할이 있어야 합니다.

기간을 닫으려면 다음을 수행합니다.

1. 통합, 내부거래, 기간 순으로 선택합니다.

2. **시나리오** 목록에서 기간을 달을 시나리오를 선택합니다.
3. **연도** 목록에서 기간을 달을 연도를 선택합니다.
4. 달을 기간을 선택합니다.
5. **기간 달기**를 누르거나 **작업, 기간 달기** 순으로 선택합니다.

## 원인 코드 관리

애플리케이션에서 내부거래 트랜잭션을 생성하면 기본 일치 상태인 [대응 취소함]이 지정됩니다. 자동 일치 프로세스를 수행하는 동안 일치 상태가 [일치함] 또는 [일치하지 않음]으로 업데이트됩니다.

트랜잭션이 [일치하지 않음] 상태인 이유를 나타내는 원인 코드를 정의할 수 있습니다. 예를 들어 파트너 엔티티의 송장이 누락되었거나 파트너가 입력한 금액이 부정확하기 때문일 수 있습니다. 애플리케이션에 대해 적합한 원인 코드 목록을 정의하면 사용자가 내부거래 트랜잭션을 입력할 때 목록에서 선택하여 코드 중 하나를 지정할 수 있습니다.

기간에 대해 [게시 전에 일치/검증을 수행하시겠습니까?] 옵션을 선택하면 [일치함] 상태의 트랜잭션이나 적합한 원인 코드가 포함된 [일치하지 않음] 상태의 트랜잭션을 게시할 수 있습니다.

애플리케이션에 대해 원인 코드를 추가, 편집 또는 삭제할 수 있습니다.

다음 절차를 참조하십시오.

- [원인 코드 추가](#)
- [원인 코드 편집](#)
- [원인 코드 삭제](#)

## 원인 코드 추가

일치하지 않는 애플리케이션 트랜잭션의 원인 코드 목록을 생성할 수 있습니다. 사용자는 트랜잭션에 대해 원인 코드 중 하나를 지정할 수 있습니다.

원인 코드를 수동으로 추가하거나, 트랜잭션을 로드하는 동안 원인 코드를 로드할 수 있습니다. 트랜잭션 로드에 대한 자세한 내용은 *Oracle Hyperion Financial Management 사용자 가이드*를 참조하십시오.

원인 코드를 추가하려면 다음을 수행합니다.

1. **통합, 내부거래, 원인 코드** 순으로 선택합니다.
2. **행 추가**를 누르거나 **작업, 행 추가** 순으로 선택합니다.
3. **이름**에서 원인 코드의 레이블을 입력합니다.

### 주:

레이블에는 최대 20자를 사용할 수 있습니다. 공백은 한 문자로 계산됩니다. 앰퍼샌드(&), 별표(\*), 백슬래시(\), 쉼표(,), 중괄호({ }), 슬래시(/), 하이픈(-), 숫자 기호(#), 마침표(.), 더하기 기호(+) 및 세미콜론(;) 문자는 부적합합니다.

4. **선택 사항:** 설명에서 원인 코드에 대한 설명을 입력합니다.



**주:**  
설명에는 최대 40자를 사용할 수 있습니다.

5. **저장**을 누르거나 **작업, 저장** 순으로 선택합니다.

## 원인 코드 편집

원인 코드를 생성한 후 코드 설명을 편집할 수 있습니다. 원인 코드 레이블은 편집할 수 없습니다.

원인 코드를 편집하려면 다음을 수행합니다.

1. **통합, 내부거래, 원인 코드** 순으로 선택합니다.
2. 원인 코드 목록에서 편집할 원인 코드를 선택합니다.
3. **설명** 열에서 설명을 편집하고 **확인**을 누릅니다.

## 원인 코드 삭제

애플리케이션의 원인 코드 목록에서 더 이상 필요하지 않은 원인 코드를 삭제할 수 있습니다.

원인 코드를 삭제하려면 다음을 수행합니다.

1. **통합, 내부거래, 원인 코드** 순으로 선택합니다.
2. 원인 코드 목록에서 삭제할 원인 코드를 선택합니다.
3. **행 삭제** 또는 **모두 삭제**를 누르거나 **작업, 행 삭제** 또는 **모두 삭제** 순으로 선택합니다.
4. **예**를 눌러 원인 코드를 삭제합니다.

## 내부거래 트랜잭션 모니터

[내부거래 트랜잭션 모니터] 기능을 사용하여 내부거래 트랜잭션 일치 프로세스를 모니터합니다. 한 기간에 다수의 내부거래 트랜잭션을 시스템에 입력하면 모든 트랜잭션이 성공적으로 입력 및 대응되었는지 확인하기 위해 일치 프로세스에 오랜 시간이 걸릴 수 있습니다. 모든 트랜잭션이 동시에 입력되지는 않으므로 관리자가 일치 프로세스를 모니터해야 합니다. [내부거래 트랜잭션 모니터] 기능을 사용하면 내부거래 트랜잭션 프로세스를 시작한 내부거래 파트너를 쉽게 확인할 수 있습니다.

내부거래 트랜잭션을 모니터하려면 [내부거래 트랜잭션 관리자] 보안 역할이 지정되어야 합니다. 이 역할이 지정되면 프로세스 상태, 잠금 상태, 엔티티 세부정보, 내부거래 트랜잭션 요약을 보고 전자메일 알림을 수행할 수 있습니다.

[내부거래 트랜잭션 모니터] 페이지에는 프로세스 상태 및 잠금 상태와 함께 내부거래 엔티티 목록이 표시됩니다. 엔티티, 프로세스 상태 또는 잠금 상태별로 표시를 필터링할 수 있습니다. 엔티티는 [내부거래 트랜잭션 모니터 세부정보] 정보에 대한 링크입니다. 엔티티를 누르면 [일치함], [일치하지 않음] 또는 [대응 취소함]과 같은 상태별로 게시된 트랜잭션 수와 게시 취소된 트랜잭션 수가 표시됩니다.

표 14-2 내부거래 트랜잭션에 대한 엔티티 상태

상태	설명
시작되지 않음	엔티티에 POV의 시나리오, 연도 및 기간에 대한 내부거래 트랜잭션이 없습니다.
시작됨	엔티티에 POV의 시나리오, 연도 및 기간에 대한 내부거래 트랜잭션이 하나 이상 있습니다.
잠금 불가능	기간에 대해 [게시 전에 일치/검증]을 선택했지만 원인 코드와 함께 일치 트랜잭션 또는 불일치 트랜잭션을 게시하지 않은 경우에는 엔티티가 [잠금 불가능] 상태입니다.

목록에서 엔티티를 눌러 엔티티에 대해 게시 및 게시 취소된 트랜잭션의 총 개수를 일치 상태별로 분류하여 볼 수 있습니다.

예를 들어 엔티티 A 옆에 있는 더하기(+) 기호를 눌러 확장하면 페이지에 엔티티 A 상태별로 게시된 트랜잭션과 게시 취소된 트랜잭션 수가 표시됩니다. [대응 취소함] 열에서 게시 취소된 트랜잭션의 값을 누르면 시스템이 [내부거래 트랜잭션 관리] 페이지로 연결되며 엔티티 A에 대해 [대응 취소함] 상태인 게시 취소된 트랜잭션을 표시하는 필터링이 선택됩니다. 동시에 여러 엔티티를 확장하여 해당 세부정보를 볼 수 있습니다.

[내부거래 트랜잭션 모니터] 페이지에서 엔티티에 대한 전자메일 알림을 보낼 수 있습니다. *Oracle Hyperion Financial Management 사용자 가이드*를 참조하십시오.

내부거래 트랜잭션을 모니터하려면 다음을 수행합니다.

1. **통합, 내부거래, 모니터** 순으로 선택합니다.
2. POV 막대에서 시나리오, 연도 및 기간을 선택합니다.
3. **표시 옵션** 목록에서 다음 옵션 중 하나를 선택합니다.
  - 레이블을 사용하여 엔티티 정보를 보려면 **레이블**을 선택합니다.
  - 설명을 사용하여 엔티티 정보를 보려면 **설명**을 선택합니다.
  - 레이블과 설명을 사용하여 엔티티 정보를 보려면 **모두**를 선택합니다.
4. **필터** 목록의 **엔티티**에서 상태를 모니터할 엔티티를 입력하거나 찾아봅니다.

 **주:**

[엔티티]를 비워 두면 목록에 있는 모든 엔티티가 반환됩니다.

5. **선택 사항:** 활성 엔티티만 표시하려면 **활성 항목만 표시**를 선택합니다.
6. **선택 사항:** 프로세스 상태별로 트랜잭션 목록을 필터링하려면 다음 트랜잭션 유형을 하나 이상 선택합니다.
  - 시작되지 않음
  - 시작됨
7. **선택 사항:** 상태별로 트랜잭션 목록을 필터링하려면 다음 트랜잭션 유형을 하나 이상 선택합니다.
  - 잠금 가능

- 잠금 불가능
  - 잠금
8. **선택사항:** 프로세스 또는 잠금 상태를 기준으로 트랜잭션 목록을 정렬하려면 열 제목을 누르고 **오름차순** 또는 **내림차순**을 선택합니다.
  9. 엔티티 옆에 있는 더하기 기호(+)를 누르거나 컨텍스트 메뉴에서 **세부정보 표시**를 선택하여 내부거래 트랜잭션 모니터 세부정보를 표시합니다.
  10. **내부거래 트랜잭션 모니터 세부정보** 상자에서 상태 열 중 하나의 값을 누릅니다. 선택한 필터 기준을 사용한 [내부거래 트랜잭션 관리] 페이지가 표시됩니다.
  11. **선택사항:** 엔티티에 대한 전자메일 알림을 보내려면 [내부거래 트랜잭션 모니터] 페이지에서 엔티티를 선택하고 **전자메일 알림**을 누르거나, 컨텍스트 메뉴에서 **전자메일 알림**을 선택하거나, **작업, 전자메일 알림** 순으로 선택합니다.

## 엔티티 잠금 및 잠금 해제

시나리오, 연도 및 기간에 대해 엔티티에 잠금을 적용하여 엔티티의 내부거래 트랜잭션에 대한 추가 수정을 방지할 수 있습니다. 해당 기간에 대해 [게시 전에 일치/검증] 옵션을 선택한 경우 엔티티를 잠그기 전에 먼저 원인 코드와 함께 모든 일치 트랜잭션과 불일치 트랜잭션을 게시해야 합니다.

### 주:

트랜잭션 잠금 상태는 데이터 잠금 상태와는 다릅니다. 데이터 잠금 상태에 대한 자세한 내용은 *Oracle Hyperion Financial Management 사용자 가이드*를 참조하십시오.

[내부거래 트랜잭션 관리자] 보안 역할이 지정된 경우 프로세스 상태, 잠금 상태, 엔티티 세부정보, 내부거래 트랜잭션 요약 보고 전자메일 알림을 수행할 수 있습니다.

엔티티가 잠기면 새 내부거래 트랜잭션을 입력할 수 없습니다. 기존 트랜잭션을 삭제하거나 변경할 수도 없습니다. 잠긴 엔티티에는 트랜잭션을 게시하거나 게시 취소할 수 없으며, 잠긴 엔티티와 관련된 트랜잭션의 일치 상태를 업데이트할 수도 없습니다. 엔티티의 일치 상태를 업데이트할 수 없으므로 파트너 엔티티가 잠겨 있지 않아도 파트너가 트랜잭션을 일치시킬 수 없습니다.

예를 들어 엔티티 A가 잠겨 있다고 가정합니다. 엔티티 A에 대해 더 이상 트랜잭션을 입력할 수 없으며, 엔티티에 대해 게시나 일치를 수행할 수도 없습니다. 엔티티 B가 잠겨 있지 않으므로 엔티티 B와 파트너 엔티티 A의 내부거래 트랜잭션은 여전히 사용할 수 있습니다. 하지만 엔티티 B를 엔티티 A와 일치시키려고 하면 시스템에서 엔티티 A의 일치 상태를 업데이트할 수 없으므로 프로세스가 실패합니다.

제출 단계를 사용하는 경우 모든 단계가 [게시됨] 상태가 될 때까지 엔티티를 잠그면 안 됩니다.

엔티티를 잠그려면 다음을 수행합니다.

1. **통합, 내부거래, 모니터** 순으로 선택합니다.
2. **시나리오**에서 엔티티의 시나리오를 선택합니다.
3. **연도**에서 엔티티의 연도를 선택합니다.
4. **기간**에서 엔티티의 기간을 선택합니다.
5. 잠글 엔티티를 선택합니다.
6. **잠금**을 누르거나, 컨텍스트 메뉴에서 **잠금**을 선택하거나, **작업, 잠금** 순으로 선택합니다.

 **팁:**

엔티티 잠금을 해제하려면 잠금을 해제할 엔티티를 선택하고 **잠금 해제**를 누르거나, 컨텍스트 메뉴에서 **잠금 해제**를 선택하거나, **작업, 잠금 해제** 순으로 선택합니다.

여러 기간을 잠금 해제하려면 **Ctrl** 키를 사용하여 기간을 강조 표시한 후 **잠금 해제**를 선택합니다. **Shift** 키를 사용하여 강조 표시하면 여러 기간을 잠금 해제할 수 없습니다.

## 내부거래 트랜잭션 요약 보기

모든 내부거래 트랜잭션의 상태 요약을 보고, 필요한 경우 엔티티를 필터링하여 특정 엔티티를 표시할 수 있습니다.

내부거래 트랜잭션 요약을 보려면 다음을 수행합니다.

1. **통합, 내부거래, 모니터** 순으로 선택합니다.
2. 엔티티의 시나리오, 연도 및 기간을 선택합니다.
3. [엔티티] 필터를 사용하여 트랜잭션 요약을 보려는 엔티티를 하나 이상 선택합니다.
4. **요약**을 누르거나, 컨텍스트 메뉴에서 **요약**을 선택하거나, **작업, 요약** 순으로 선택합니다.
5. 요약 보기가 완료되면 **확인**을 누릅니다.



## 프로세스 관리 제출 단계 관리

프로세스 관리는 재무 데이터의 검토 및 승인 관리입니다. 검토를 위해 데이터가 특정 시나리오, 연도, 기간, 엔티티 및 값 차원의 데이터 조합인 프로세스 단위로 구성됩니다. 검토 주기 중에 프로세스 단위에 대해 승인 요청, 제출, 승인, 거부, 게시 등의 작업을 수행합니다.

프로세스 관리에서 데이터 프로세스 단위를 여러 제출 단계로 나눈 후 데이터 서브세트로 작업할 수 있습니다. 검토하는 동안 전체 프로세스 단위 대신 프로세스 단위의 각 단계를 승인 요청할 수 있습니다. 이렇게 하면 검토 프로세스를 적용하기 위한 추가 시나리오가 필요 없게 됩니다.

프로세스 관리 제출 단계를 관리하려면 UseSubmissionPhase 애플리케이션 속성을 활성화해야 하며, 애플리케이션 관리자 또는 검토 수퍼바이저여야 합니다.

### 제출 단계 정의

검토 프로세스 요구사항은 기간마다 다를 수 있습니다. 예를 들어 월별 마감 주기에는 1월과 2월의 대차대조표 및 수익/손실 데이터에 대한 단일 단계 검토 프로세스가 필요할 수 있습니다. 3월과 같은 분기별 달의 경우 분기별 달기 검토 프로세스에 대차대조표 및 이익/손실 데이터와 보충 데이터에 대한 다단계 제출 주기가 필요할 수 있습니다.

검토 프로세스 요구사항이 시나리오마다 다를 수도 있습니다. 예를 들어 실제 시나리오의 경우 검토를 위해 제출하는 데 대차대조표 및 수익/손실 계정만 필요할 수 있습니다. 예산 시나리오의 경우 모든 계정이 필요할 수 있고, 예측 시나리오의 경우 수익/손실 계정과 보충 데이터만 필요할 수 있습니다.

검토 프로세스에 최대 9개의 제출 단계를 사용할 수 있습니다. 예를 들어 실제 시나리오에서는 1단계에서 대차대조표 및 이익/손실 계정을 검토하도록 제출하고, 2단계에서 보충 데이터를 제출할 수 있습니다. 예산 시나리오에서는 1단계에서 내부거래 데이터를 제출하고, 2단계에서 대차대조표 및 이익/손실 계정을 제출하고, 3단계에서 보충 데이터를 제출할 수 있습니다.

제출 단계를 설정하려면 다음 태스크를 수행합니다.

- 제출 단계를 사용할 애플리케이션 및 차원 메타데이터 속성을 설정합니다. [애플리케이션 설정 정의](#)를 참조하십시오.
  - UseSubmissionPhase 애플리케이션 속성을 Y로 설정합니다. 기본적으로 제출 단계 설정은 비활성화되어 있으며, 애플리케이션에서 단계별 제출을 사용하려면 이 속성을 설정해야 합니다.
  - SupportSubmissionPhaseForAccounts, SupportSubmissionPhaseForCustom 또는 SupportSubmissionPhaseForICP 속성을 필요에 따라 설정합니다. 프로세스 관리를 사용할 차원(계정, 사용자정의 및 ICP 멤버)을 정의해야 합니다. 예를 들어 애플리케이션에 계정에 의한 제출이 필요하고 사용자정의 또는 내부거래(ICP) 차원에 대한 제출은 필요하지 않은 경우 SupportSubmissionPhaseForAccounts 속성을 선택할 수 있습니다. 차원을 하나 이상 활성화해야 합니다.
- 검증 계정을 지정합니다. [애플리케이션 설정 정의](#)를 참조하십시오.

검증 계정은 프로세스 제어에서 및 데이터 잠금을 위해 사용됩니다. 검증 계정 금액이 0이어야 데이터를 승격, 승인 또는 잠글 수 있습니다.

프로세스 제어에서 또는 데이터 잠금을 위해 검증을 사용하지 않으려면 검증 계정을 비워둡니다.

단계별 제출을 사용하는 경우 각 제출 단계에 대해 별도의 검증 계정을 지정할 수 있습니다. 애플리케이션에 최대 9개의 제출 단계를 사용할 수 있습니다. 단계별 제출을 사용하는 애플리케이션의 경우 검증 계정 설정은 1단계의 검증 계정을 식별합니다. 검증 계정 2-9는 나머지 단계를 식별합니다.

- 제출 그룹을 정의하고 차원 멤버에 제출 그룹을 지정합니다. [제출 그룹 설정](#)을 참조하십시오.
- 제출 단계에 제출 그룹을 지정합니다. [단계에 제출 그룹 지정](#)을 참조하십시오.

## 제출 그룹 설정

각 애플리케이션에 대해 제출 그룹에 포함할 차원 멤버를 결정해야 합니다. 예를 들어 제출 그룹 1에 대해 현금 계정과 투자 계정을 정의할 수 있습니다.

메타데이터 파일에서 제출 그룹에 대해 SubmissionGroup=0 - 99 또는 <공백> 속성을 설정합니다.

기본적으로는 값이 비어 있습니다. <공백> 값은 기본적으로 값 1로 설정됩니다.

제출 그룹을 0으로 설정하면 계정이 검토 프로세스에 포함되지 않습니다.

여러 차원이 단계별 제출에 사용되는 경우 시스템에서 차원 멤버의 최대 그룹 지정을 기준으로 셀 제출 그룹 지정을 결정합니다. 제출 그룹을 지정하기 전에 필요한 모든 그룹 지정을 고려해야 합니다.

예 1:

Account=2

C1=1

C2=2

C3=1

C4=1

ICP=1

이 차원의 최대 제출 그룹 수가 2이므로 셀의 제출 그룹 값은 2입니다.

예 2:

Account=1

C1=3

C2=2

C3=5

C4=1

ICP=3

이 차원의 최대 제출 그룹 수가 5이므로 셀의 제출 그룹 값은 5입니다.

## 제출 그룹 및 단계 예

단계에 대해 애플리케이션 및 차원 멤버 속성을 설정하고 차원 멤버의 제출 그룹을 정의한 후 각 제출 단계에 계정의 제출 그룹을 지정할 수 있습니다. 프로세스 관리를 지원하는 시나리오에만 지정이 적용됩니다. 하지만 시나리오 및 기간별로 지정해야 합니다.

다음 예에서는 샘플 제출 그룹 및 해당 그룹의 제출 단계 지정을 보여 줍니다.

계정	Submission Group
HistData	0
Cash	1
Invest	1
ICRec	2
ICPay	2
Liability	3
Equity	3
Revenue	4
Expense	4
SuppData1	5
SuppData2	5
Headcount	6
MiscData	6

C1(제품)	Submission Group
[None]	1
Golf Balls	7
Tennis Balls	8
Soccer Balls	9

기존 계정은 상위 계정에서 제출 그룹을 상속하지 않고, 상위 계정은 1차 하위 구성요소의 제출 그룹을 사용하지 않습니다. 각 계정에 제출 그룹을 지정해야 합니다. 이 예에서 HistData 계정에는 제출 그룹 0이 지정되었으므로 계정에 검토 프로세스가 필요하지 않습니다.

다음 예에서는 실제 시나리오에 대한 기간별 제출 그룹 지정을 보여 줍니다.

**표 15-1 예: 제출 단계에 제출 그룹 지정**

기간	제출 단계 1	제출 단계 2	제출 단계 3
1월(단일 단계)	1, 2, 3, 4	N/A(보충 데이터 및 제품 데이터 필요 없음)	N/A(보충 데이터 및 제품 데이터 필요 없음)
2월	1, 2, 3, 4	N/A	N/A
3월(복수 단계)	2	1, 3, 4, 7, 8, 9	5, 6
4월	1, 2, 3, 4	N/A	N/A
5월	1, 2, 3, 4	N/A	N/A
6월(복수 단계)	2	1, 3, 4, 7, 8, 9	5, 6
7월	1, 2, 3, 4	N/A	N/A
8월	1, 2, 3, 4	N/A	N/A
9월(복수 단계)	2	1, 3, 4, 7, 8, 9	5, 6
10월	1, 2, 3, 4	N/A	N/A
11월	1, 2, 3, 4	N/A	N/A
12월(복수 단계)	2	1, 3, 4, 7, 8, 9	5, 6

### 1월 - 단일 단계 지정

1월 마감의 경우, 이 예에서는 프로세스 관리가 제출 단계 1에는 필요하지만 제출 단계 2와 3에는 필요하지 않습니다. 이 마감 주기는 월별 단기 마감 주기이므로 내부거래, 대차 대조표 및 수익/손실 데이터(그룹 1, 2, 3 및 4)가 동일한 제출 단계에서 제출됩니다. 보충 데이터가 필요하지 않습니다.

### 3월 - 다단계 지정

3월 분기별 마감 중에는 프로세스 관리에서 여러 단계에 걸쳐 데이터 제출을 사용합니다.

제출 단계 1에는 내부거래 데이터가 필요하므로 YCRec 및 ICPay 계정의 데이터가 포함됩니다. 이 예에서는 제출 그룹 2가 지정된 모든 계정입니다.

3월의 제출 단계 2에는 대차 대조표 계정과 수익/손실 계정(Cash, Invest, Liability, Equity, Revenue 및 Expense 계정)이 포함됩니다. 이 예에서는 제출 그룹 1, 3 또는 4가 지정된 모든 계정입니다.

3월의 제출 그룹 3에는 보충 데이터가 포함되며 SuppData1, SuppData2, Headcount 및 MiscData와 같은 보충 데이터 계정이 포함됩니다. 이 예에서는 제출 그룹 5와 6이 지정된 모든 계정입니다.

## 단계에 제출 그룹 지정

각 제출 단계에 제출 그룹을 지정할 수 있습니다. 프로세스 관리를 지원하는 시나리오에만 지정이 적용됩니다. 그룹을 지정하지 않으면 해당 그룹 내의 차원 멤버에 프로세스 관리가 적용되지 않습니다.

프로세스 관리 제출 그룹을 지정하려면 애플리케이션 관리자 또는 검토 수퍼바이저여야 합니다.

시나리오 및 기간별로 단계에 제출 그룹을 지정합니다. 제출 단계에 대해 그룹을 하나 이상 입력할 수 있으며, 여러 그룹을 지정하는 경우 심표를 구분자로 사용합니다(예: 1, 5, 6, 8, 9). 그룹 범위를 지정할 수 있습니다. 예를 들어 그룹 1, 2, 3, 4, 5, 7 및 8을 제출 단계에 지정하려면 1-5, 7-8을 지정할 수 있습니다. 심표를 사용하여 일정한 범위의 그룹을 하나 이상 입력하는 경우 데이터를 제출하고 새로 고치면 범위에 포함되는 그룹이 표시됩니다. 예를 들어 1, 2, 3, 4는 1-4로 표시됩니다.

적합한 그룹은 1에서 99 사이입니다. 제출 단계 1의 기본값은 모든 그룹을 나타내는 키워드 ALL입니다. 이 지정을 변경하기 전에는 모든 그룹이 제출 단계 1에 속합니다.

같은 그룹을 동일한 기간의 여러 단계에 지정할 수는 없습니다. 예를 들어 단계 1과 단계 2에 그룹 3을 모두 지정할 수 없으므로 단계 1에 그룹 2-5를 지정하고 단계 2에 그룹 3과 8을 지정할 수는 없습니다. 제출 그룹은 동일한 기간 내에서 하나의 단계에만 지정할 수 있습니다. 제출 그룹이 이미 지정된 단계에 제출 그룹을 지정하려고 하면 오류 메시지가 표시됩니다. 한 제출 단계의 모든 그룹을 다른 제출 단계로 이동하면 지정된 시나리오와 기간에 대해 원래 단계가 시스템에서 완전히 지워집니다.

제출 단계 지정을 건너뛸 수 있습니다. 예를 들어 2단계에 그룹을 지정하지 않고 제출 단계 1과 3에 그룹을 지정할 수 있습니다. 제출 단계에 지정되지 않은 그룹은 검토 프로세스의 일부로 간주되지 않습니다. 적절한 보안 클래스 액세스 권한이 있는 모든 사용자는 검토 레벨 보안을 확인하지 않고도 이러한 차원 멤버를 사용할 수 있습니다. 지정되지 않은 셀은 데이터 입력 전에 프로세스 관리를 위해 시작할 필요가 없습니다.

제출 단계에 제출 그룹을 지정하려면 다음을 수행합니다.

#### 1. 통합, 데이터, 제출 단계 순으로 선택합니다.

기본적으로 모든 단계가 표시됩니다. 오른쪽의 [옵션] 창에서 숨기려는 단계를 지울 수 있습니다.

#### 2. 시나리오를 변경하려면 POV에서 시나리오 차원을 누르고 시나리오를 선택한 다음 확인을 누릅니다.

#### 3. 각 단계 열에 해당 제출 단계의 그룹을 입력하고 **Enter** 키를 누릅니다.

- 여러 그룹을 입력하려면 심표를 구분자로 사용합니다.
- 그룹 범위를 지정하려면 대시를 구분자로 사용합니다.
- 모든 그룹을 나타내려면 ALL을 지정합니다.

#### 4. 제출을 누르거나 작업, 제출 순으로 선택하여 데이터를 저장합니다.

#### 5. 새로고침을 누르거나 작업, 새로고침 순으로 선택하여 데이터베이스를 새로고칩니다.

## 지정되지 않은 제출 그룹 보기

[프로세스 제어]에서 어떤 단계에도 지정되지 않은 제출 그룹을 볼 수 있습니다. 제출 단계가 지정되지 않은 제출 그룹은 검토 프로세스에 포함되지 않습니다. 지정되지 않은 그룹을 보면 검토 프로세스에서 실수로 그룹이 생략되었는지 여부를 확인할 수 있습니다.

어떤 제출 단계에도 지정되지 않은 차원 멤버에 지정된 그룹과 어떤 차원 멤버에도 지정되지 않은 제출 단계에 지정된 그룹이 표시됩니다. 차원 멤버나 단계에 지정되지 않은 그룹은 표시되지 않습니다.

예를 들면, 애플리케이션은 1-10까지 지정된 제출 그룹을 갖습니다.

기간	제출 단계 1	제출 단계 2	제출 단계 3
1월	1,2,3,4	5,7	8,9
2월	1,2,3,4	5-8	N/A
3월	2	1,3,4	5,6
4월	1,2,3,4	5,6	8
5월	1,2,3,4	5-8	N/A
6월	2	1,3,4	5,6
7월	1,2,3,4	N/A	N/A
8월	1,2,3,4	N/A	N/A
9월	2	1,3,4	5,6
10월	1,2,3,4	N/A	N/A
11월	1,2,3,4	NA	N/A
12월	2	1,3,4	5,6

앞의 예에서 1월에 대해 지정되지 않은 그룹을 표시하는 옵션을 선택하면 다음 그룹이 지정되지 않은 그룹으로 표시됩니다.

기간	지정되지 않은 그룹
1월	6,10

지정되지 않은 제출 그룹을 보려면 다음을 수행합니다.

1. **통합, 데이터, 제출 단계** 순으로 선택합니다.  
오른쪽 창에는 [지정되지 않은 그룹] 정보가 표시됩니다.
2. 기간에 대한 행을 선택하고 **지정되지 않은 그룹** 창에서 **새로고침**을 누릅니다.

# 16

## 전자메일 알림 관리

내부거래 트랜잭션에 대해, 그리고 프로세스 관리 검토 프로세스 중에 전자메일 알림을 사용할 수 있습니다. 전자메일 알림은 시스템의 주요 이벤트 또는 데이터 변경을 강조 표시하는 데 도움이 됩니다. 예를 들어 내부거래 트랜잭션이 일치하지 않아 일치시켜야 한다거나 프로세스 단위가 다음 승인 요청 레벨에 대해 준비되었다는 전자메일 알림을 보낼 수 있습니다.

전자메일 알림은 표준 SMTP를 사용하여 전송되므로 인터넷 전자메일을 지원하는 모든 전자메일 시스템에서 알림을 사용할 수 있습니다. 알림을 사용하려면 Financial Management 구성 유틸리티를 실행할 때 SMTP 서버 이름을 지정하고 전자메일 설정을 구성해야 합니다. *Oracle Enterprise Performance Management System 설치 및 구성 가이드*를 참조하십시오.

알림 프로세스에서는 LDAP, MSAD 또는 Native Directory와 같은 인증 파일에 저장된 전자메일 주소를 사용합니다.

전자메일 알림을 보내거나 받으려면 먼저 애플리케이션에서 사용자 및 데이터 보안 권한을 설정해야 합니다. 알림에 대해 시나리오와 엔티티에 지정된 보안 클래스가 전자메일 알림을 지원해야 하며, 사용자에게 전자메일 알림을 받을 수 있는 보안 역할이 지정되어야 합니다. 보안 역할 설정에 대한 자세한 내용은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

## 프로세스 관리 알림 설정

프로세스 제어의 상태 변경에 따라 전자메일 알림을 생성하도록 프로세스 제어 모듈을 설정할 수 있습니다. First Pass, Review Level 1-10, Submitted, Approved 또는 Published 작업에 대해 알림을 설정할 수 있습니다.

프로세스 단위가 Not Started 레벨에 있는 경우 또는 사인오프 작업에 대해서는 전자메일 알림이 생성되지 않습니다.

프로세스 관리 전자메일 알림을 설정하려면 다음을 수행합니다.

1. 프로세스 단위의 시나리오에서는 메타데이터 파일에서 SupportsProcessManagement 메타데이터 속성을 A로 설정하여 알림을 허용합니다.

### 주:

이 속성을 활성화하면 검토 프로세스를 수행하는 동안 시나리오에서 전자메일을 받을 수 있는 보안 권한이 있는 사용자에게 전자메일 알림이 생성됩니다.

2. [프로세스 제어를 위한 전자메일 알림 수신] 보안 역할에 사용자를 지정합니다.
3. 사용자에게 프로세스 단위의 시나리오와 엔티티에 지정된 보안 클래스에 대한 [모두] 또는 [승인 요청] 액세스 권한을 지정하고 각 보안 클래스에 대한 알림을 추가합니다.
4. 보안 클래스의 [보안 액세스] 섹션에서 [알림 지원] 옵션을 Y(Yes)로 설정하여 알림을 활성화합니다. (예: [기본값];User1@NativeDirectory;All;Y).

모든 기준을 충족하는 사용자는 전자메일 알림을 받습니다.

**표 16-1 프로세스 관리 사용자 역할 및 알림 통지**

작업 전후의 프로세스 단위 레벨	통지를 받는 프로세스 관리 사용자 역할
일차 통과	엔티티에 대해 [모두] 또는 [승인 요청] 액세스 권한이 있는 사용자가 통지를 받습니다.
검토 레벨 1-10	검토자 레벨의 검토자와 제출자 역할이 통지를 받습니다. 예를 들어 Review Level 1의 경우 검토자 1과 제출자 역할이 통지를 받습니다.
제출됨	검토 슈퍼바이저 역할이 통지를 받습니다. 이 역할이 할당된 사용자만 제출된 프로세스 단위를 승인할 수 있습니다.
승인됨	검토자 1 - 검토자 10과 제출자 역할이 통지를 받습니다.
게시됨	엔티티에 대해 [모두], [읽기] 또는 [승인 요청] 액세스 권한이 있는 사용자가 통지를 받습니다.

프로세스 검토 작업을 수행하면 설정된 보안 권한에 따라 적절한 사용자에게 자동으로 전자메일 알림이 생성됩니다. 작업을 수행한 사용자는 확인 전자메일도 받게 됩니다.

 주:

[애플리케이션 관리자] 역할이 할당된 사용자는 전자메일 알림을 받지 않습니다. [애플리케이션 관리자] 역할이 지정된 사용자가 전자메일 알림을 받으려면 관리자를 별도의 사용자로 설정하고 알림을 받는 역할을 지정합니다.

## 내부거래 알림 설정

전자메일 알림을 받을 수 있는 보안 권한이 지정된 사용자에 대해 내부거래 트랜잭션 전자메일 알림을 생성할 수 있습니다.

보안 역할 설정에 대한 자세한 내용은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

내부거래 트랜잭션 전자메일 알림을 설정하려면 다음을 수행합니다.

1. [내부거래를 위한 전자메일 알림 수신] 보안 역할에 사용자를 지정합니다.
2. 보안 클래스의 [보안 액세스] 섹션에서 [알림 지원] 옵션을 Y(Yes)로 설정하여 알림을 활성화합니다. (예: [기본값];User1@NativeDirectory;All;Y).

모든 기준을 충족하는 사용자는 내부거래 트랜잭션 및 내부 파트너 일치 보고서 모듈에서 전자메일 알림을 받습니다.

내부거래 트랜잭션의 전자메일 알림 생성에 대한 자세한 내용은 *Oracle Hyperion Financial Management 사용자 가이드*를 참조하십시오.



# A

## 구성 설정

설정 모듈에서는 Oracle Hyperion Financial Management 구성 설정을 보고 수정할 수 있습니다. 기본적으로 [설정] 테이블은 설치 시 권장되는 설정으로 채워집니다.

필요한 경우 설정 값을 변경할 수 있습니다. 예를 들어 메모리 사용에 따라 값을 변경하거나 성능을 향상시켜야 할 수도 있습니다. 이러한 값 업데이트에 대한 고려 사항은 [성능 최적화](#)를 참조하십시오.

특정 애플리케이션에 대한 설정을 재정의할 수 있습니다. 시스템 레벨 값을 변경하려면 Financial Management 관리자 보안 역할이 있어야 합니다. 애플리케이션 레벨 값을 변경하려면 Financial Management 관리자 및 애플리케이션 관리자 보안 역할이 둘 다 있어야 합니다.

설정 목록을 정렬하고 설정을 검색하며 이러한 설정을 Excel 워크시트로 내보낼 수 있습니다. 설정 재정의를 삭제할 수 있지만 시스템에서 생성된 설정은 삭제할 수 없습니다.

특정 애플리케이션 설정에 대한 런타임 값을 볼 수도 있습니다. [유효 설정 보기](#)를 참조하십시오.

다음 항목을 참조하십시오.

- [사용가능 구성 설정](#)
- [구성 설정 변경](#)
- [값 대체](#)
- [설정 테이블 표시 변경](#)
- [설정 검색](#)
- [유효 설정 보기](#)
- [설정 익스포트](#)
- [설정 삭제](#)

## 사용가능 구성 설정

표 A-1 구성 설정

설정	설명	값
AllowOverlappingConsolidation	이 사용자가 겹치기 통합 프로세스를 실행할 수 있도록 허용할지 여부입니다.	0 - 사용자가 겹치기 통합을 실행할 수 있도록 허용 안 함. 현재 실행 중인 겹치기 통합이 하나 있으면 사용자가 새 통합 프로세스를 시작할 수 없습니다. 1 - 사용자가 겹치기 통합을 실행할 수 있도록 허용. 현재 다른 겹치기 통합 프로세스가 실행 중인 경우 새 통합을 시작할지를 묻는 경고 메시지가 표시됩니다. 기본값: 1

표 A-1 (계속) 구성 설정

설정	설명	값
AutoClearDeadTasks	완료된 태스크를 시스템에서 자동으로 지울지 여부입니다.	0 - 완료된 태스크를 시스템에 유지 1 - 완료된 태스크를 시스템에서 지움 기본값: 1
AutoClearEAFlatfileTasksAfterSeconds	실행 중인 태스크에서 [데이터 추출] 태스크를 자동으로 지우고 애플리케이션 서버에서 파일을 삭제하기 전에 시스템에서 기다릴 시간(초)입니다	범위: 600초(10분) - 864,000초(10일) 기본값: 86,400초
BRCRowFetchSize	테이블의 모든 반복에 대한 행 가져오기 크기입니다. LCM 애플리케이션 스냅샷 익스포트에만 사용됩니다.	범위: 1000 -100000행 기본값: 5000
BRCThreadPoolCount	스레드 풀 크기 - LCM 애플리케이션 스냅샷 임포트 및 익스포트를 처리하는 데 사용되는 스레드 수	범위: 10 - 100개 스레드 기본값: 15
ConcurrentCalculationWaitTimeInSecond	최대 개수의 동시 계산이 실행 중인데 사용자가 새 계산을 실행하면 시스템은 시간이 초과될 때까지 대기했다가 POV 세부정보로 시간 초과에 대한 시스템 로그에 메시지를 기록합니다. 최대 개수의 동시 계산이 실행 중이고 ConcurrentCalculationWaitTimeInSecond = -1인데 사용자가 새 계산을 실행하면 시스템은 계산 중 하나가 완료될 때까지 제한 없이 대기합니다. 이 설정은 관리자가 애플리케이션 및 서버별로 변경할 수 있으나 15초 미만으로는 설정할 수 없습니다.	기본값: 30초

표 A-1 (계속) 구성 설정

설정	설명	값
ConsolidationMultiThreadingScheme	통합 프로세스가 실행되는 방법과 관련된 여러 설정에 사용됩니다. 다른 설정에 영향을 주지 않고 각 설정을 활성화하거나 비활성화할 수 있습니다. 두 개 이상의 옵션을 선택하고 선택한 옵션의 합으로 매개변수를 설정할 수 있습니다. 예를 들어 옵션 2와 4를 사용하려면 값을 6으로 설정합니다. 값이 0이면 세 가지 옵션이 하나도 활성화되지 않은 제한된 다중 스레딩의 레거시 통합 동작을 나타냅니다. 이 설정은 서버 및/또는 애플리케이션 레벨에서 재정의할 수 있습니다.	적합한 값: 1 - 숫자가 상위 엔티티에 통합되기 전에 통합 프로세스에서 최대 [상위 멤버] 값까지 모든 하위 엔티티를 계산하고 환산할 수 있게 합니다. 2 - 다중 스레드 상위 레벨 엔티티에 대한 통합 프로세스를 허용하여 대규모 통합의 성능을 훨씬 향상시킵니다. 4 - 모든 기본 레벨 엔티티에서 초기 계산을 건너뛵니다. 관련 상위 엔티티를 처리할 때 필요한 경우 이러한 엔티티가 계산됩니다. 8 - 모두 통합 및 데이터가 있는 항목 모두 통합 중 통합 프로세스에서 최종 오류의 영향을 받은 1차 하위만 통합할 수 있습니다. 기본값: 2
DataSize	테이블 구성요소의 가져오기 크기입니다.	범위: 25 - 2,000 기본값: 500
DefaultAdminPage	통합 관리를 위한 기본 시작 페이지입니다.	적합한 값: 시스템 메시지, 애플리케이션, 시스템 설정, 시스템 사용자, 시스템 관리, 태스크 감사, 데이터 감사, 태스크 플로우 관리, 태스크 플로우 상태, DSN 구성 페이지 중 하나 기본값: 시스템 메시지
DefaultColFetchSize	데이터 양식 및 데이터 그리드의 기본 열 가져오기 크기입니다.	범위: 25 - 2,000 기본값: 50
DefaultRowFetchSize	데이터 양식 및 데이터 그리드의 기본 행 가져오기 크기입니다.	범위: 25 - 2,000 기본값: 250

표 A-1 (계속) 구성 설정

설정	설명	값
DSStartupOption	<p>애플리케이션 시작 및 종료 시기를 판별합니다.</p> <p>이 설정은 글로벌로 설정하거나 애플리케이션 레벨, 서버 레벨 또는 둘 다에서 대체될 수 있습니다.</p>	<p>0 - 요청에 따라 애플리케이션이 시작됨, 첫번째 사용자가 애플리케이션을 열려고 시도하는 경우입니다. 최종 사용자가 애플리케이션에서 로그오프하면 애플리케이션이 종료됩니다. 애플리케이션을 필요할 때만 유지하려는 경우 이 설정이 유용합니다. 아카이브에 사용되는 애플리케이션이 항상 작동 중일 필요는 없습니다.</p> <p>1 - Java Server 시작 시 애플리케이션이 시작됩니다. 최종 사용자가 시스템을 로그오프한 후에도 애플리케이션은 계속 작동을 유지합니다. 항상 애플리케이션을 작동 상태로 유지하여 사용자가 애플리케이션을 열 때 대기 시간이 길지 않도록 하려는 경우 이 설정이 유용합니다.</p> <p>2 - 요청 시 애플리케이션이 시작됩니다. 최종 사용자가 시스템을 로그오프한 후에도 애플리케이션은 계속 작동을 유지합니다.</p> <p>기본값: 0</p>
EnableRulesLogging	<p>규칙 로깅을 사용할지 여부입니다. 규칙 이슈를 격리시키기 위해 디버깅 용도로 사용됩니다.</p>	<p>0 - 로깅 사용 안함</p> <p>1 - 로깅 사용</p> <p>기본값: 0</p>
EnableRulesProfiling	<p>규칙 프로파일링을 사용으로 설정합니다. 애플리케이션 수준에서 또는 전체적으로 값을 대체할 수 있습니다.</p>	<p>0 - 규칙 프로파일링 사용 안함</p> <p>1 - 규칙 프로파일링 사용</p> <p>기본값: 0</p>
EnableRunningTasksMaskUserNames	<p>타사 이외의 사용자가 볼 때 실행 중인 태스크에서 사용자 이름을 마스킹할지 여부입니다.</p>	<p>0 - 사용자 이름 마스킹 사용 안 함</p> <p>1 - 사용자 이름 마스킹 사용</p> <p>기본값: 0</p>

표 A-1 (계속) 구성 설정

설정	설명	값
IcmSystemReportTranslationMode	통화 환산이 필요한 내부거래 매칭 시스템 보고서를 실행하는 경우 환산된 값이 기본적으로 데이터베이스에 기록됩니다. 이 경우 보고서 실행 시간이 길어지고 데이터베이스 크기가 훨씬 증가할 수 있습니다. 이 설정을 통해 환산된 값의 처리 방법을 제어할 수 있습니다. 트랜잭션 모드 = 0(sub translate 수행 및 데이터 저장) 트랜잭션 모드 = 1(sub translate 수행 및 데이터 저장 안 함) 트랜잭션 모드 = 2(기본 환산을 사용하여 즉시 환산 수행, sub translate 수행 안 함 및 데이터 저장 안 함)	0 - 기존 환산 논리 사용. 규칙에 Sub Translate를 사용하고 변경사항을 데이터베이스로 커밋합니다. 1 - 즉시 규칙 사용. 규칙에 Sub Translate를 사용하지만 환산된 값을 데이터베이스에 쓰지 않습니다. 참고: CN 상태의 상위 엔티티도 환산됩니다. 2 - 즉시 기본 환산 사용. 기본 환산만 실행합니다(Sub Translate 없음). 변경사항이 데이터베이스에 기록되지 않습니다. 참고: CN 상태의 상위 엔티티는 환산되지 않습니다. 기본값: 0
MaxDataCacheSizeInMB	애플리케이션 서버가 설을 저장하는 데 할당하는 최대 메모리 (MB)입니다.	범위: 값은 500 이상이어야 합니다. 기본값: 4,500
MaxFileSelectionOnLoad	여러 파일 선택 컨트롤에서 선택할 수 있는 최대 파일 수입니다.	값은 1보다 크거나 같아야 합니다. 기본값: 10
MaxNumConcurrentCalculations	애플리케이션 서버당 허용되는 최대 동시 계산 수입니다. 이 설정은 관리자가 애플리케이션 서버별로 변경할 수 있습니다.	기본값: 8
MaxNumConcurrentConsolidations	애플리케이션 서버당 허용되는 최대 동시 통합 수입니다. 설정된 값을 초과하여 수행되는 모든 통합은 스케줄링된 통합으로 대기됩니다. 이 설정은 글로벌로 설정하거나 애플리케이션 서버 레벨에서 재정의할 수 있습니다.	범위: 범위에 대한 하드 한도는 없습니다. 값은 양수여야 합니다. 기본값: 8
MaxNumConcurrentConsolidationsPerClient	사용자 클라이언트당 허용되는 최대 동시 연결 수입니다. 설정된 값을 초과하여 수행되는 모든 통합은 스케줄링된 통합으로 대기됩니다.	기본값: 8
MaxNumCubesInRAM	FreeLRU 알고리즘에 대한 최대 서브큐브 수입니다.	범위: 값은 100 이상이어야 합니다. 기본값: 60,000
MaxNumDataRecordsInRAM	시스템에서 RAM에 저장할 최대 데이터 레코드 수입니다.	범위: 범위에 대한 하드 한도는 없습니다. 값은 양수여야 합니다. 기본값: 30,000,000

표 A-1 (계속) 구성 설정

설정	설명	값
MaxNumRetriesOfBaseLevelCalculation	<p>통합 프로세스가 기본 레벨 엔티티에서 계산을 재시도하는 최대 횟수입니다.</p> <p>이 설정을 사용하면 통합 프로세스 중 기준 레벨 엔티티 계산을 지정된 재시도 횟수까지 또는 해당 엔티티 상태가 OK가 될 때까지 여러 번 반복할 수 있습니다.</p> <p>하나의 기준 레벨 엔티티에서 실행 중인 규칙의 HS.ImpactStatus 함수가 이미 계산된 다른 엔티티에 영향을 주고 최상위 엔티티 상태가 OK가 아닌 경우 사용됩니다.</p> <p>예: California와 Connecticut은 USA 상위 엔티티로 롤업되는 기준 레벨 엔티티입니다. USA 계산 중 소계에 HS.ImpactStatus를 사용하는 경우 Connecticut 엔티티가 영향을 받습니다. 통합 실행 후에는 Connecticut 하위의 변경으로 인해 USA 상위 엔티티의 상태가 OK가 아닙니다. 이 설정이 2로 설정되면 시스템은 계산 프로세스를 두 번 재시도합니다. 두번째 계산 시도에서 California는 OK이고, Connecticut은 계산되어 USA로 롤업되고, USA 엔티티 상태는 OK입니다. 3개 또는 4개의 종속성 레벨이 있는 경우 필요한 횟수만큼 재시도하도록 설정을 조정할 수 있습니다.</p>	<p>범위: 1-4</p> <p>기본값: 0</p>
MinDataCacheSizeInMB	<p>최소 데이터 캐시 크기(MB)입니다. 이 값을 크게 설정하면 시스템별로 데이터 캐시 증가 시도 횟수를 줄여 메모리 조각화를 줄일 수 있습니다. 일반적으로 데이터 캐시는 요구에 따라 증가하며 한 번에 최대 25MB 증가합니다.</p>	<p>범위: 100-5,000</p> <p>기본값: 2,250</p>
NumConsolidationThreads	<p>통합당 허용되는 최대 통합 스레드 수입니다. 이 값을 줄이면 시스템의 리소스 사용이 제한되어 통합 성능이 느려집니다. 시스템의 Oracle Hyperion Financial Management에 사용할 수 있는 CPU 코어 수는 동일해야 합니다.</p>	<p>범위: 범위에 대한 하드 한도는 없습니다. 값은 1 이상이어야 합니다.</p> <p>기본값: 8</p>

표 A-1 (계속) 구성 설정

설정	설명	값
NumCubesLoadedBeforeCheckingLRU	FreeLRU 알고리즘을 트리거하기 위해 RAM에 로드되어야 하는 큐브 수입니다. FreeLRU는 두 조건 중 하나가 충족될 때 (NumMinutesBeforeCheckingLRU가 충족되거나 NumCubesLoadedBeforeCheckingLRU에 도달할 때) 트리거됩니다.	범위: 범위에 대한 하드 한도는 없습니다. 값은 양수여야 합니다. 기본값: 100
NumDataLoadsAllowed	애플리케이션 서버당 허용되는 최대 동시 데이터 로드 태스크 수입니다. 이 설정은 글로벌로 설정하거나 애플리케이션 서버 레벨에서 재정의할 수 있습니다.	범위: 범위에 대한 하드 한도는 없습니다. 값은 양수여야 합니다. 기본값: 8
NumEAExportsAllowed	애플리케이션 서버당 허용되는 최대 동시 데이터 추출 태스크 수입니다. 이 설정은 글로벌로 설정하거나 애플리케이션 서버 레벨에서 재정의할 수 있습니다.	범위: 범위에 대한 하드 한도는 없습니다. 값은 양수여야 합니다. 기본값: 8
NumEAThreads	데이터 추출당 스레드 수입니다.	범위: 범위에 대한 하드 한도는 없습니다. 값은 양수여야 합니다. 기본값: 8
NumMinutesBeforeCheckingLRU	FreeLRU 알고리즘의 제한을 확인하는 간격(분)입니다.	값은 1 이상이어야 합니다. 기본값: 15
NumReportsAllowed	보고서 유형당, 서버당, 애플리케이션당 허용되는 최대 동시 보고서 태스크 수입니다.	기본값: 8
NumThreadsToUseWhenUpdatingCubeStatusWhenWasChanged	애플리케이션 시스템을 변경할 때 상태를 업데이트할 때 사용하는 스레드 수입니다.	값은 1 이상이어야 합니다. 기본값: 16
SessionManagerTimeoutInMS	Financial Management의 세션 관리자에서 사용되는 제한 시간 (밀리초)입니다.	값은 60,000 이상이어야 합니다. 기본값: 1,200,000
OverrideUserFetchSizeWhenOpeningCube	데이터 양식 및 데이터 그리드에서 모든 사용자에게 가져오기 크기를 재정의할지 여부입니다	0 - 모든 사용자에게 가져오기 크기를 재정의하지 않음 1 - 모든 사용자에게 가져오기 크기 재정의 기본값: 0
WebformDoCalculateOnSubmit	사용자가 데이터 양식에서 [데이터 제출]을 누를 경우 데이터를 제출한 후 자동으로 계산할지 여부입니다.	0 - 자동으로 계산 안 함 1 - 자동으로 계산 기본값: 0

## 구성 설정 변경

설정을 변경하려면 관리자 보안 권한이 있어야 합니다. 시스템에서 제공하는 설정의 경우 값을 제외한 다른 열은 변경할 수 없습니다.

[참고] 열을 사용하여 값이 변경된 이유와 같은 설명을 입력할 수 있습니다. 행의 [참고] 아이콘 위로 마우스를 이동하면 설정에 대한 기존 참고 내용을 볼 수 있습니다.

설정이 수정되면 설정을 수정한 사용자의 이름과 설정이 수정된 날짜 및 시간을 표시하도록 [설정] 테이블이 업데이트됩니다.

파란색으로 표시된 설정은 사용자가 수정 가능한 설정이 기본값과 일치하지 않으며 값이 업데이트되었음을 나타냅니다.

구성 설정을 변경하려면 다음을 수행합니다.

1. **탐색, 관리, 통합 관리** 순으로 선택합니다.
2. **관리 태스크에서 시스템을 확장한 다음 설정을** 누릅니다.
3. [설정] 테이블에서 설정 값을 변경합니다.

#### **팁:**

원래 값으로 재설정하려면 **재설정**을 누르고 확인 프롬프트에서 **예**를 누릅니다.

4. **선택사항:** 설명을 입력하려면 행의 **참고** 아이콘을 누르고 설명을 입력한 다음 **확인**을 누릅니다.
5. **저장**을 누릅니다.
6. **새로고침**을 눌러 변경사항을 표시합니다.

## 값 대체

서버 및 애플리케이션에 대한 설정 기본값을 재정의할 수 있습니다. 설정은 글로벌 경우에는 재정의할 수 없습니다.

설정을 복사하려면 다음을 수행합니다.

1. [설정] 테이블에서 복사할 설정을 선택합니다.
2. **대체**를 누르거나, **작업, 대체** 순으로 선택하거나, 설정을 마우스 오른쪽 버튼으로 누르고 **대체**를 선택합니다.
3. [재정의] 대화상자에서 설정 값을 변경하고 설정이 적용되는 서버 및 애플리케이션을 선택합니다.
4. **선택사항:** 설명을 입력하려면 행의 **참고** 아이콘을 누르고 설명을 입력한 다음 **확인**을 누릅니다.
5. **적용 후 닫기**를 눌러 새 값을 적용합니다.  
사용자가 생성한 새 설정은 굵게 표시되며 테이블에서 선택됩니다.  
**선택 사항:** 다른 설정을 재정의하려면 **적용 후 다른 값 재정의**를 누릅니다.
6. **저장**을 누릅니다.

## 설정 테이블 표시 변경

설정 열을 정렬하고 열 순서를 재조정할 수 있습니다.

표시할 열을 선택하려면 다음을 수행합니다.



1. **설정** 목록에서 **보기**, **열** 순으로 누르고 표시할 열을 선택하거나 **모두 표시**를 선택합니다.
2. **선택사항:** 열을 표시하거나 숨기려면 **보기**, **열**, **열 관리** 순으로 선택하고 화살표 키를 사용하여 열을 [숨겨진 열] 또는 [표시되는 열] 목록으로 이동합니다.
3. **선택사항:** 열 정렬 순서를 변경하려면 머리글 아이콘을 누르고 **오름차순 정렬** 또는 **내림차순 정렬**을 선택합니다.

## 설정 검색

선택한 기준에 따라 설정을 검색할 수 있습니다. [설정 관리] 탭에서 설정을 적용할 설정 이름, 서버 및 애플리케이션과 업데이트된 날짜, 설정을 생성하거나 업데이트한 사용자를 검색할 수 있습니다.

[유효 설정] 탭에서는 런타임 시 서버에서 애플리케이션에 사용되는 값을 볼 수 있습니다. 이름, 서버 및 애플리케이션별로 설정을 검색할 수 있습니다.

설정을 검색하려면 다음을 수행합니다.

1. **검색** 필드에서 검색할 기준을 선택하거나 입력합니다.
2. 모든 기준과 일치시키려면 **모두**를 선택하고 선택한 기준과 일치시키려면 **임의**를 선택합니다.
3. **검색**을 누릅니다.



**팁:**

검색 설정을 재설정하려면 [검색] 옆의 **재설정** 버튼을 누릅니다.

## 유효 설정 보기

[유효 설정] 탭에서 애플리케이션 설정에 대한 런타임 값을 볼 수 있습니다.

유효 설정을 보려면 다음을 수행합니다.

1. **설정 관리**에서 서버 이름과 애플리케이션 이름을 선택합니다.
2. **유효 설정**을 누르거나 **작업**, **유효 설정** 순으로 선택합니다.  
유효 설정은 선택한 애플리케이션과 서버를 기반으로 합니다. 재정의된 기본값은 파란색으로 표시됩니다.
3. 기본 설정 목록으로 돌아가려면 **설정 관리**를 누르거나 **작업**, **설정 관리** 순으로 선택합니다.

## 설정 익스포트

설정 목록을 내보내어 Excel 워크시트에 저장할 수 있습니다.

설정을 내보내려면 다음을 수행합니다.

1. [설정 관리]에서 **익스포트**를 누르거나 **작업**, **익스포트** 순으로 선택합니다.  
현재 검색을 기반으로 행이 내보내집니다.
2. 다운로드 지침에 따라 파일을 Excel 워크시트로 다운로드합니다.

## 설정 삭제

재정의된 설정을 삭제할 수 있습니다. 시스템에서 생성된 설정은 삭제할 수 없습니다.

설정을 삭제하려면 다음을 수행합니다.

1. [설정] 테이블에서 삭제할 설정 행을 선택합니다.
2. **선택 항목 삭제**를 누르거나, **작업, 선택 항목 삭제** 순으로 선택하거나, 설정을 마우스 오른쪽 버튼으로 누르고 **선택 항목 삭제**를 선택합니다.
3. 확인 프롬프트가 나타나면 **예**를 누릅니다.

# B

## 성능 최적화

이 장에서는 Oracle Hyperion Financial Management 애플리케이션, 데이터베이스 관리 및 일반적인 운영 체제 개념을 잘 알고 있다고 가정합니다. 실제 구현 및 환경은 비즈니스 요구사항, Financial Management 데이터 세트, 네트워크 토폴로지 및 하드웨어 사용에 따라 달라집니다. 따라서 고유한 구현에 맞게 이러한 지침을 조정하는 방법을 고려해야 합니다. 모든 테스트 결과 및 성능 수치는 개념 조정을 설명하기 위한 예로만 사용되었습니다.

### ▲ 주의:

부적절한 설정 및 구성을 사용하면 Financial Management가 작동하지 않을 수도 있습니다.

조정 설정을 구현하기 전에 엔드 투 엔드 성능 테스트를 수행하여 기본 구성의 기준 성능 데이터를 가져오고 조정 설정을 증분 변경한 다음 성능 데이터를 수집하는 것이 좋습니다. 구성 변경 시 시스템 성능이 저하되지 않고 향상되는지 확인해야 합니다.

지원되는 플랫폼 및 구성요소는 Oracle Enterprise Performance Management System Supported Platform Matrix([Oracle EPM Supported Platform Matrix](#))를 참조하십시오.

## 성능 개요

### Oracle Hyperion EPM System Performance 소개

Oracle Hyperion EPM System 성능을 최대화하려면 모든 구성요소를 모니터, 분석 및 조정해야 합니다. 이 섹션에서는 성능을 모니터하는 데 사용할 수 있는 툴 및 Oracle Hyperion Financial Management의 성능을 최적화하는 기술에 대해 설명합니다.

## 공통 EPM 설치 디렉토리 참조

이 장에서는 다음 디렉토리 참조를 사용합니다.

### 미들웨어 홈

미들웨어 홈은 Oracle WebLogic Server 홈과 선택적으로 EPM Oracle 홈을 비롯한 하나 이상의 Oracle 홈으로 구성됩니다. 기본 설치 디렉토리는 Oracle\Middleware입니다. 이 장에서는 미들웨어 홈 위치를 `MIDDLEWARE_HOME`이라고 합니다.

### EPM Oracle 홈

Oracle 홈에는 특정 제품을 호스트하는 데 필요한 설치된 파일이 포함되며 미들웨어 홈의 디렉토리 구조 내에 있습니다. 기본 EPM Oracle 홈 위치는 `MIDDLEWARE_HOME\EPMSys11R1`입니다. 이 장에서는 EPM Oracle 홈 위치를 `EPM_ORACLE_HOME`이라고 합니다.

### EPM Oracle 인스턴스

또한 구성하는 동안 일부 제품은 구성 시 정의된 EPM Oracle 인스턴스에 구성요소를 배포합니다. EPM Oracle 인스턴스의 기본 위치는 `MIDDLEWARE_HOME\user_projects\epmsystem1`입니다. 이 장에서는 EPM Oracle 인스턴스 위치를 `EPM_ORACLE_INSTANCE`라고 합니다.

## Financial Management 레코드 및 서브큐브

Oracle Hyperion Financial Management의 레코드에는 지정된 차원 멤버 교차점에 대한 모든 기본 기간의 데이터가 포함됩니다.

서브큐브는 다음 차원의 동일한 멤버를 공유하는 모든 셀로 구성됩니다.

- 시나리오
- 연도
- 엔티티
- 값

통화 서브큐브 및 노드 서브큐브라는 두 가지 유형의 서브큐브가 있습니다. 이러한 유형의 서브큐브는 엔티티 및 값 차원 멤버를 사용하는 방식이 서로 다릅니다.

- 통화 서브큐브에는 해당 비노드 값 차원 멤버를 공유하는 셀이 포함됩니다. 통화 서브큐브의 경우 엔티티 멤버의 상위는 관련이 없습니다. 해당하는 비노드 값 차원 멤버는 다음과 같습니다.
  - 사용자 정의 통화 멤버. 사용자 정의 통화마다 3쌍의 값 차원 멤버가 하나씩 있습니다. 예를 들어 애플리케이션에 USD라는 통화가 있는 경우 이 통화에 대한 3쌍의 값 차원 멤버는 USD, USD Adjs 및 USD Total입니다. 3쌍은 세 개의 별도 서브큐브로 간주됩니다.
  - 엔티티의 기본 통화를 가리키는 3쌍. 3쌍은 <Entity Currency>, <Entity Curr Adjs> 및 <Entity Curr Total> 값 멤버로 구성됩니다.
  - [None] 값 멤버.

#### 주:

상위 엔티티 기본 통화를 가리키는 비노드 값 차원 멤버(<Parent Currency>, <Parent Curr Adjs> 및 <Parent Curr Total>)는 통화 서브큐브와 관련이 없습니다.

- 노드 서브큐브에는 공통 노드 값 차원 멤버를 공유하는 셀이 포함됩니다. 노드 서브큐브의 경우 상위 엔티티 및 1차 하위 엔티티 멤버를 모두 지정해야 합니다. 노드 값 차원 멤버는 다음과 같습니다.
  - [Contribution Total]
  - [Contribution Adjs]
  - [Contribution]
  - [Elimination]
  - [Proportion]
  - [Parent Total]
  - [Parent Adjs]

- [Parent]

서브큐브는 통합을 위한 기본 데이터 단위이므로 Financial Management의 여러 위치에서 데이터 이동 및 처리가 서브큐브를 기준으로 수행됩니다.

## Financial Management에 대한 조정 권장 사항

성능 조정 Financial Management는 반복 프로세스입니다.

 주:

특정 프로덕션 워크로드에 대해 조정을 수행해야 합니다. OATS(Oracle Application Testing Suite)와 같은 로드 생성 도구에서 워크로드가 생성될 때 조정을 수행할 수 있습니다.

이 섹션에서는 다음을 포함하여 성능 조정에 대한 빠른 시작을 제공하는 여러 영역에 대해 설명합니다.

- 운영 체제 매개변수 조정
- HTTP 서버 매개변수 조정
- Oracle 데이터베이스 매개변수 조정

 주:

이 목록은 성능 조정을 시작하는 데 유용한 도구지만 조정할 영역의 전체 목록은 아닙니다. 조정 시 성능이 향상될 수 있는 위치를 파악하려면 구현 내에서 특정 성능 이슈를 모니터 및 추적해야 합니다.

## 성능 문제 진단

성능 이슈가 발생할 경우 수정 조치를 취하기 전에 먼저 원인을 확인하는 것이 중요합니다. 포괄적인 문제 분석을 수행하기 전에는 성능 관련 매개변수 설정을 변경하거나 다른 조치를 취하지 않는 것이 좋습니다.

## 모니터링 툴 사용

모니터링 도구를 사용하여 진단 프로세스의 일부로 성능 데이터를 수집하는 것이 좋습니다. Oracle Hyperion Financial Management 애플리케이션 서버, 웹 서버, 데이터베이스 서버 및 네트워크 계층을 모니터링하면 유용한 성능 데이터를 얻을 수 있습니다.

Linux 환경에서는 OS Watcher Black Box와 같은 도구를 사용하여 운영 체제 및 Financial Management 프로세스를 모니터링할 수 있습니다.

Windows 환경에서는 Microsoft 성능 모니터링과 같은 도구를 사용하여 Financial Management 애플리케이션 프로세스 성능을 모니터링할 수 있습니다. Financial Management 애플리케이션과 관련된 카운터를 수집하도록 성능 모니터를 구성하는 단계는 다음에서 확인할 수 있습니다.

[성능 모니터 구성](#)

모니터할 성능 카운터는 다음과 같습니다.

하위 시스템	카운터	지침
메모리	메모리: 사용 가능한 시스템 페이지 테이블 항목	사용 가능한 시스템 페이지 테이블 항목이 8,000개 미만인 경우 경고. 사용 가능한 시스템 페이지 테이블 항목이 5,000개 미만인 경우 심각.
메모리	메모리: 사용가능 메가바이트	설치된 물리적 메모리의 20%-25% 이상이어야 합니다. 이 경우 페이지 지정 작업을 주의해서 모니터링합니다.
메모리	메모리: 페이지 읽기/초	이 카운터는 항상 1,000 미만이어야 합니다.
프로세서	프로세서: % 프로세서 시간: _합계	총 프로세서 활용은 약 70%-80% 미만이어야 합니다.
프로세서	프로세서: % 프로세서 시간: (N)	각 프로세서 인스턴스는 활용이 약 70%-80% 미만이어야 합니다.
디스크	물리적 디스크: 평균 디스크 초/전송	약 25ms 미만이어야 합니다. 일반 규칙: 평균 디스크 초/전송(디스크 대기 시간 카운터)이 25ms보다 훨씬 큰 경우 디스크 하위 시스템이 비정상이며 병목입니다. 이 카운터는 문제를 해결하는 방법을 알려주지 않으며 문제가 있는 것만 나타냅니다.
디스크	물리적 디스크: 평균 디스크 대기열 길이	평균은 디스크 스핀들 수보다 작아야 합니다. SAN을 사용하는 경우 이 카운터를 무시하고 대기 시간 카운터인 물리적 디스크\평균 디스크 초/읽기 및 물리적 디스크\평균 디스크 초/쓰기에 집중합니다.
디스크	물리적 디스크: 평균 디스크 초/읽기	평균 값은 20ms 미만이어야 합니다. 스파이크(최대값)는 50ms 이하여야 합니다.
디스크	물리적 디스크: 평균 디스크 초/쓰기	평균 값은 20ms 미만이어야 합니다. 스파이크(최대값)는 50ms 이하여야 합니다.
네트워크	네트워크 인터페이스: 바이트 합계/초	100-Mbps NIC(네트워크 인터페이스 카드)의 경우 6-7MB/초 미만이어야 합니다. 1000-Mbps NIC의 경우 60-70MB/초 미만이어야 합니다.
네트워크	네트워크 인터페이스: 패킷 아웃바운드 오류	항상 0이어야 합니다.

프로세스	카운터/지침
Oracle Hyperion Financial Management: 인스턴스: 1 JHsxServer 2. XFMDDataSource	<p>프로세스: % 프로세서 시간 - 프로세스 프로세서 활용은 90% 미만이어야 합니다.</p> <p>프로세스: 전용 바이트 - 특정 프로세스에 대해 전용으로 할당된 바이트를 보고합니다. 누출 프로세스의 경우 값이 증가합니다.</p> <p>프로세스: 작업 세트 - 프로세스에 할당된 공유 및 개인 바이트를 보고합니다. 누출 프로세스의 경우 값이 증가합니다.</p> <p>프로세스: 페이지 오류/초 - 프로세스로 인해 발생한 총 오류(하드 및 소프트 오류) 수를 보고합니다. 누출 프로세스의 경우 값이 증가합니다.</p> <p>프로세스: 페이지 파일 바이트 - 페이징 파일의 크기를 보고합니다. 메모리 누출 중 해당 값이 증가합니다.</p> <p>프로세스: 핸들 수 - 객체에 대해 열린 애플리케이션이 생성하는 핸들 수를 보고합니다. 핸들은 프로그램에서 액세스해야 하는 리소스를 식별하는 데 사용됩니다. 그러나 이 카운터 값은 메모리 누출 중에 증가하는 경향이 있습니다.</p> <p>프로세스: 가상 바이트 - 프로세스에서 사용 중인 가상 주소 공간의 현재 크기(바이트)입니다. perfmon 프로세스의 가상 바이트는 빠른 속도로 증가하고 메모리를 해제하지 않으므로 애플리케이션의 메모리 누출을 나타냅니다.</p> <p>프로세스: 최대 가상 바이트 - 프로세스에서 하나의 임의 시간에 사용한 가상 주소 공간의 최대 크기(바이트)입니다. perfmon 프로세스의 가상 바이트는 빠른 속도로 증가하고 메모리를 해제하지 않으므로 애플리케이션의 메모리 누출을 나타냅니다.</p> <p>프로세스: 페이지 지정되지 않은 풀 바이트 - 시스템 메모리(물리적 메모리) 영역인, 페이지 지정되지 않은 풀의 크기(바이트)입니다.</p>

## RDA(원격 진단 에이전트) 사용

RDA(원격 진단 에이전트)는 Perl 프로그래밍 언어로 작성된 엔진에 의해 실행되는 명령행 진단 스크립트 세트입니다. 수집된 데이터는 문제 진단에 도움이 되는 포괄적인 환경 정보를 제공합니다.

RDA 실행은 Oracle Hyperion Financial Management 애플리케이션의 서버큐브 크기를 결정하는 데 특히 유용할 수 있습니다. RDA는 My Oracle Support 웹 사이트를 통해 사용할 수 있습니다. 시작하려면 다음 기술 자료 문서를 참조하십시오.

[문서 1100612.1](#)

## 참조 애플리케이션 사용

참조 애플리케이션은 성능 이슈를 진단하는 데 사용되는 애플리케이션입니다. 참조 애플리케이션은 환경이 제대로 조정되었는지 확인하기 위해 내부 타이밍과 비교할 수 있는 정해진 개수의 태스크를 실행합니다. 특정 유형의 참조 애플리케이션을 실행하면 문제를 식별하는 데 유용할 수 있습니다. 그러나 Oracle Hyperion Financial Management의 모든 성능 특성을 보여줄 수 있는 하나의 애플리케이션은 없습니다. 다른 하나의 애플리케이션에서 최상의 성능을 유지하려면 보려면 및

성능 저하를 가장 것 같습니다. 관련된 매개변수에는 데이터 볼륨, 서버큐브당 레코드 배포, 엔티티 구조, 통화 수 등이 포함됩니다.

## 운영 체제 매개변수 조정

### Windows 매개변수 조정

Windows 플랫폼의 경우 일반적으로 기본 TCP/IP 설정을 사용해도 충분합니다. 그러나 로드가 심한 경우 *MaxUserPort* 및 *TcpTimedWaitDelay*를 조정해야 할 수도 있습니다. 이러한 매개변수는 애플리케이션에서 요청한 사용자 포트의 가용성을 결정합니다.

매개변수	기본값	제안 값
<b>TcpTimedWaitDelay</b> 이 매개변수는 애플리케이션이 TCP 연결을 닫은 후 OS가 포트를 회수하기 위해 기다리는 시간을 제어합니다. 기본값은 4분입니다. 사용자 로드가 심한 경우 이러한 제한을 초과하여 주소 사용 중 연결 예외가 발생할 수도 있습니다.  이 매개변수는 Windows 레지스트리의 HKLM\System\CurrentControlSet\Services\Tcpip\Parameters 위치에서 설정합니다. 값 이름: <i>TcpTimedWaitDelay</i> 값 유형: <b>DWORD</b> 데이터: 30(10진수)	240	30
<b>MaxUserPort</b> 값 이름: <i>MaxUserPort</i> 값 유형: <b>DWORD</b> 데이터: 65534(10진수)  Windows2008부터 언급된 포트 번호를 설정하려면 다음 명령을 실행해야 합니다. <pre>netsh int ipv4 set dynamicportrange protocol=tcp startport=10100 numberofports=55433 store=persistent</pre> 제공된 명령은 support.microsoft.com에 있는 다음 문서를 기반으로 하고 있습니다. 동적 포트 범위 Windows 설정	5000	65534



## 웹 서버 조정

Oracle Hyperion Financial Management 웹 서버 및 기타 EPM 구성요소에 대한 주요 조정 단계는 컴퓨터에 첫번째 EPM System 제품을 설치할 때 설치되는 EPM System Configurator에 포함되어 있으며 컴퓨터에 설치된 모든 제품을 구성하는 데 사용됩니다. 자세한 내용은 *Oracle Enterprise Performance Management System 설치 및 구성 가이드*를 참조하십시오.

### HFM 웹 조정

#### HFM 웹 시간 초과 매개변수

EPM 구성 파일	기본값	제안 값
<p><b>Weblogic 포함 웹 서버 플러그인</b> 다음 설정은 Oracle HTTP Server 또는 IIS를 사용하여 요청을 Weblogic으로 프록시할 때 적용됩니다. 사용되는 웹 서버 유형의 선택 항목은 EPM System Configurator에서 설정됩니다. 매개변수는 대소문자를 구분하며 수동으로 추가해야 합니다.</p>		
<p><b>WLIOTimeoutSecs</b> 플러그인이 WebLogic Server의 요청에 대한 응답을 기다리는 시간을 정의합니다.</p>	7,200	7,200
<p><b>WLSocketTimeoutSecs</b> 연결하는 동안 소켓의 시간 초과(초)를 설정합니다.</p>	600	600
<p><b>(OHS 웹 서버 플러그인)</b> EPM_ORACLE_INSTANCE\httpConfig\ohs\config\OHS\ohs_component\mod_wl_ohs.conf hfmadf 인스턴스에 적용됩니다. 예: &lt;LocationMatch ^/hfmadf&gt; SetHandler weblogic-handler WeblogicCluster&lt;WLCluster&gt;:&lt;포트&gt; WLIOTimeoutSecs 7200 Idempotent OFF WLSocketTimeoutSecs 600 &lt;/LocationMatch&gt;</p>		

EPM 구성 파일	기본값	제안 값
<b>(IIS 웹 서버 플러그인)</b>		
EPM_ORACLE_INSTANCE\httpConfig\VirtualHosts\hfmadf\iisproxy.ini		
WlForwardPath=/hfmadf		
PathTrim=/		
WebLogicHost=<호스트>		
WebLogicPort=<포트>		
KeepAliveEnabled=true		
KeepAliveSecs=20		
WLIOTimeoutSecs=3600		
Idempotent=OFF		
WLSocketTimeoutSecs=750		
<b>Weblogic</b>		
<b>잠긴 스레드(Weblogic Admin 콘솔)</b>		
장기 실행 태스크(예: 메타데이터 로드)은 HFM 애플리케이션 프로세스가 로드 파일을 처리 중일 때 중지된 것처럼 보일 수 있습니다. 이러한 이슈가 발생할 경우 잠긴 스레드 시간을 늘립니다.		
Weblogic 콘솔에서 다음 설정을 변경합니다.		
잠금 및 편집을 선택합니다.		
서버를 선택하고 HFMWeb(N)을 누릅니다.		
[조정] 탭을 선택합니다.		
잠긴 스레드 최대 시간을 1200으로 변경합니다.		
잠긴 스레드 타이머 간격을 1200으로 변경합니다.		
변경 활성화를 선택합니다.		
HFMWeb0 서버를 다시 시작해야 합니다.		
여기서 HFMWeb(N)은 배포된 HFM 웹 서버 수 및 업데이트 중인 서버에 따라 HFMWeb0, HFMWeb1 등입니다.		
잠긴 스레드 최대 시간	600	1200
잠긴 스레드 타이머 간격	60	1200

## HFM 웹 조정 매개변수

이 섹션에서는 기타 HFM 웹 UI 조정 매개변수를 다룹니다.

조정 매개변수 및 위치

매개변수	기본값	제안 값
<b>WebLogic</b>		
<b>HFM ADF 웹 애플리케이션 Java 힙 크기(Xms 및 Xmx)</b>		
HFM ADF 웹 애플리케이션의 관리되는 인스턴스 각각에 사용할 수 있는 힙 크기입니다.		
<b>Windows</b>		
HFM ADF 웹이 Windows 서비스로 설치되었습니다. 힙 매개변수는 다음 위치에서 JVMOptionXX 옵션을 편집하여 조정할 수 있습니다.		
HKLM\SOFTWARE\Hyperion Solutions\HFMWeb (N)\HyS9FinancialMangementWeb		
여기서 HFMWeb(N)은 배포된 HFM 웹 서버 수 및 업데이트 중인 서버 인스턴스에 따라 HFMWeb0, HFMWeb1 등입니다.		
압축 배포에서는 모든 EPM 제품에 대해 하나의 EPMSys(N) 웹 애플리케이션이 있으므로 HFM ADF 웹을 별도로 조정할 수 없습니다. 구성 옵션이 포함된 Windows 레지스트리의 위치는		
HKLM\SOFTWARE\Hyperion Solutions\EPMServer (N)\HyS9EPMServer입니다.		
Xms는 초기 힙 크기를 설정하며 Xmx와 동일한 크기로 설정되어야 합니다.	128m	8192m
<b>Windows</b>		
-Xms에 대한 JVMOptionXX 찾기		
데이터 유형: REG_SZ		
데이터 값: -Xms<크기>		
Xmx는 최대 힙 크기를 설정합니다. 32비트 시스템에서는 1.8GB 이하로 설정하고 64비트 시스템에서는 <i>사용가능</i> 물리적 메모리의 75% 이하로 설정하는 것이 좋습니다.	4096m	8192m
<b>Windows</b>		
-Xmx에 대한 JVMOptionXX 찾기		
데이터 유형: REG_SZ		
데이터 값: -Xmx<크기>		

## Financial Management 애플리케이션 조정

일반적으로 단일 애플리케이션 서버에서 한 번에 둘 이상의 Oracle Hyperion Financial Management 애플리케이션을 실행하면 모든 애플리케이션의 성능이 저하됩니다. 다른 애플리케이션이 유틸리티 상태인 경우에도 한 번에 네다섯 개 이상의 애플리케이션을 실행하려고 시도하지 마십시오. 유틸리티 애플리케이션을 실행하려면 데이터베이스 연결 및 CPU 시간이 필요하기 때문입니다.

### 일반적으로 조정되는 Financial Management 설정

구성 설정의 전체 목록은 이 가이드의 부록 A에서 확인할 수 있습니다. [구성 설정](#)을 참조하십시오.

#### MaxNumDataRecordsInRAM

일반적으로 MaxNumDataRecordsInRAM은 RAM에 유지할 레코드 수를 결정하기 때문에 가장 중요한 설정입니다. 메모리의 레코드 수는 두 조건 NumMinutesBeforeCheckingLRU 및 NumCubesLoadedBeforeCheckingLRU 중 하나에 도달할 때 확인됩니다. [사용가능 구성 설정](#)을 참조하십시오.

범위: 범위에 대한 하드 한도는 없습니다.

기본값: 30,000,000

#### 주:

RAM의 총 레코드 수가 이 값을 초과하면 FreeLRU 호출을 통해 메모리에서 레코드를 해제하여 Oracle Hyperion Financial Management 서버에 사용할 일부 메모리를 확보합니다. "FreeLRUCachesIfMoreRAMIsNeeded가 데이터 큐브를 해제했습니다."라는 정보 메시지가 로그에 기록됩니다.

#### MinDataCacheSizeInMB

이 값을 높게 설정하면 DataCache 증가 시도 횟수를 줄여 메모리 조각화를 줄일 수 있습니다. 일반적으로 DataCache는 요구에 따라 증가하며 한 번에 최대 25MB 증가합니다.

범위: 범위에 대한 하드 한도는 없습니다.

기본값: 2,000

#### MaxDataCacheSizeInMB

Oracle Hyperion Financial Management 애플리케이션 서버가 셀 값과 셀 상태를 저장하기 위해 할당하는 최대 메모리 크기를 제어합니다. 시스템에 더 많은 메모리가 필요한 경우 셀 값과 셀 상태가 LRU 논리에 따라 디스크로 페이지 아웃됩니다. 해결 방법은 캐시 크기를 늘리는 것입니다.

범위: 값은 500 이상이어야 합니다.

기본값: 4,500

 주:

페이지 지정으로 인해 통합 작업을 성능이 저하되는 경우 MaxDataCacheSizeInMB를 늘려 페이지 지정을 최소화할 수 있습니다. 시스템에서 불필요하게 셀을 디스크로 페이지 아웃하지 않도록 이 값은 "MaxNumDataRecordsInRAM"에서 허용된 총 메모리 사용을 초과해야 합니다. MaxDataCacheSizeInMB 설정이 너무 낮으면 데이터 레코드를 저장하고 페이지 지정을 시작할 메모리가 부족하여 시스템 성능이 저하됩니다.

## MaxNumCubesInRAM

이 설정은 지정된 시간에 메모리에 유지되는 큐브 수를 제어합니다. 메모리의 큐브 수는 두 조건 NumMinutesBeforeCheckingLRU 및 NumCubesLoadedBeforeCheckingLRU 중 하나에 도달할 때 확인됩니다. 이 설정을 낮추면 희소 애플리케이션(많은 엔티티가 있지만 엔티티당 레코드 수가 비교적 적은 애플리케이션)의 높은 메모리 사용을 줄일 수 있습니다. 이 설정은 통합 및 데이터 검색을 비롯한 모든 데이터 작업에 영향을 줍니다.

범위: 100-500,000

기본값: 60,000

## IcmSystemReportTranslationMode

통화 환산이 필요한 내부거래 매칭 시스템 보고서를 실행하는 경우 환산된 값이 기본적으로 데이터베이스에 기록됩니다. 이 경우 보고서 실행 시간이 길어지고 데이터베이스 크기가 훨씬 증가할 수 있습니다. 이 설정을 통해 환산된 값의 처리 방법을 제어할 수 있습니다.

범위: 0,1,2

기본값: 0

적합한 데이터 옵션:

- 0 - 기본 동작입니다. 환산된 값이 데이터베이스에 기록됩니다.
- 1 - 즉시 계산에서 규칙에 Sub Translate를 사용하지만 환산된 값이 데이터베이스에 기록되지 않습니다. 참고: CN 상태의 상위 엔티티도 환산됩니다.
- 2 - 즉시 계산에서 기본 환산만 실행합니다(Sub Translate 규칙 없음). 변경사항이 데이터베이스에 기록되지 않습니다. 참고: CN 상태의 상위 엔티티는 환산되지 않습니다.

## NumConsolidationThreads

이 설정은 애플리케이션 서버당 통합의 다중 스레드를 제어합니다.

범위: 범위에 대한 하드 한도는 없습니다. 값은 양수여야 합니다.

기본값: 8

 주:

이 값을 줄이면 시스템의 리소스 사용이 제한되어 통합 성능이 느려집니다. 값을 올리면 CPU 활용이 증가하며 다른 구성요소의 성능이 저하될 수도 있습니다.

 **팁:**

테스트를 수행하여 NumConsolidationThreads를 4개에서 8개로 늘릴 경우의 영향을 평가했습니다. 아래 결과는 통합 시간이 1달 단축되었음을 보여 줍니다.

트랜잭션	사용자 92명 NumConsolidationThreads = 4 평균 응답 시간(초)	사용자 92명 NumConsolidationThreads = 8 평균 응답 시간(초)
01_Run_Consolidation_A_1105	10.11	3.22
02_Run_Consolidation_A_000516.15		9.47
03_Run_Consolidation_A_2205	7.75	3.19
04_Run_Consolidation_A_330518.67		9.17
05_Run_Consolidation_B_11058.21		3.14
06_Run_Consolidation_B_000513.26		9.27
07_Run_Consolidation_B_22057.69		6.20
08_Run_Consolidation_B_330518.29		9.41
09_Run_Consolidation_C_000530.59		22.08

 **주:**

테스트 실행 중 Financial Management 애플리케이션 서버의 CPU 활용은 적절했으며 더 심한 워크로드를 처리할 용량을 사용할 수 있었습니다.

 **주:**

이 값을 늘리기 전에 모든 애플리케이션 서버에서 모든 설정이 동일인지 확인합니다. 테스트를 통해 동시 통합을 실행할 때와 대기열에서 대기 중인 통합을 실행할 때 총 통합 시간이 실제로 증가하기 전에 지정된 서버에서 실행되는 현재 통합 수를 확인해야 합니다.

## MaxNumConcurrentConsolidations

애플리케이션 서버당 허용되는 동시 통합 수를 제어합니다. 이 값을 초과하여 실행되는 모든 통합은 스케줄링된 통합으로 대기됩니다.

범위: 1 - 제한 없음

기본값: 8

시나리오: Oracle Hyperion Financial Management 애플리케이션 서버가 3개 있는 경우 각 서버에서 최대 8개의 동시 통합을 실행할 수 있지만 NumConsolidationsAllowed의 기본값이 서버 3개에서 하나의 애플리케이션에 대해 총 8개의 동시 통합만 실행할 수 있도록 제한합니다. 예: 사용자가 ServerA에서 통합 6개를 제출한 다음 ServerB에서 통합 2개를 더

제출하면 통합 8개가 모두 실행됩니다. 동시에 사용자가 ServerC 다음 통합 제출하는 경우 실행되지 않습니다. 실행 중인 태스크 페이지에서 이전에 8 finish(상태인 경우 스케줄이 잡힌 시작).



**주:**

이 값을 늘리기 전에 모든 애플리케이션 서버에서 모든 레지스트리 설정이 동일한지 확인합니다. 테스트를 통해 동시 통합을 실행할 때와 대기열에서 대기 중인 통합을 실행할 때 총 통합 시간이 실제로 증가하기 전에 지정된 서버에서 실행되는 현재 통합 수를 확인해야 합니다.

## NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged

메타데이터 로드 중 엔티티 계층이 변경되면(멤버가 이동, 추가 또는 삭제됨) 사용 중인 멤버가 하위 멤버 또는 상위 멤버와 일치하지 않을 수 있습니다. Oracle Hyperion Financial Management는 각 시나리오 및 연도 조합에 대해 계산 상태를 확인해야 합니다. 이 설정을 사용하면 단일 스레드 대신 여러 스레드가 계산 상태를 확인 및 업데이트할 수 있습니다.

범위: 1 - 제한 없음

기본값: 16

## 대규모 애플리케이션에 대한 Financial Management 메모리 설정

다음 표에는 사용 가능한 물리적 메모리에 따라 매개변수에 대한 제안 값이 포함되어 있습니다. 이 표에서는 Oracle Hyperion Financial Management가 시스템에서 실행되는 메모리를 많이 사용하는 유일한 프로세스이며, 하나의 Financial Management 애플리케이션만 실행 중이라고 가정합니다.



**주:**

여러 애플리케이션이 활성화된 경우 서버에 설치된 총 물리적 메모리를 애플리케이션 수로 나누어 각 애플리케이션에 사용 가능한 물리적 메모리를 구합니다.

사용가능 물리적 메모리	MaxNumDataRecordsInRAM	MaxDataCacheSizeinMB
8GB	10,000,000	1,500
16GB	30,000,000	4,500
32GB	60,000,000	9,000
64GB	100,000,000	15,000
128GB	200,000,000	30,000

## 애플리케이션별 설정

이전에는 환경 레벨 설정이었던 일부 설정이 애플리케이션 레벨로 확장되었습니다. 동일한 Financial Management 애플리케이션 서버에서 여러 애플리케이션을 실행해야 하지만 하나의 애플리케이션만 많이 사용되는 경우 일반적으로 애플리케이션 레벨에서 Oracle Hyperion Financial Management를 조정하면 도움이 됩니다. 이 경우 많이 사용되는 애플리케이션은 서버 메모리를 대량 사용할 수 있고 나머지 애플리케이션은 기본값만 사용하도록 조정되어

애플리케이션이 이용할 수 있는 메모리 크기를 제한합니다. Financial Management 애플리케이션별 또는 서버별 설정을 사용하려면 구성 설정 장에 설명된 재정의 절차를 사용합니다. [값 대체](#)를 참조하십시오.

 **주:**

재정의 우선 순위는 다음과 같습니다.

1. 애플리케이션별 설정은 없고 설치 레지스트리 설정이 있는 경우 설치 레지스트리 설정이 사용됩니다.
2. 애플리케이션별 설정이 없는 경우 서버 키에 정의된 설정이 사용됩니다.
3. 애플리케이션별 또는 서버 설정이 정의되지 않은 경우 기본값이 사용됩니다. 설정 및 기본값은 [구성 설정](#)을 참조하십시오.

애플리케이션별 설정을 사용하여 다음 설정을 재정의할 수 있습니다.

- AllowOverlappingConsolidationOverride
- AutoClearDeadTasks
- AutoClearEAFlatfileTasksAfterSeconds
- EnableRulesLogging
- EnableRunningTasksMaskUserNames
- DSStartupOption
- MaxDataCacheSizeInMB
- MaxNumConcurrentConsolidations
- MaxNumCubesInRAM
- MaxNumDataRecordsInRAM
- MaxNumRetriesOfBaseLevelCalculation
- MinDataCacheSizeInMB
- NumConsolidationThreads
- NumCubesLoadedBeforeCheckingLRU
- NumDataLoadsAllowed
- NumEAExportsAllowed
- NumEAThreads
- NumMinutesBeforeCheckingLRU
- NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged

## Financial Management 애플리케이션 서버 조정

When tuning Oracle Hyperion Financial Management 애플리케이션 서버를 조정하는 경우 대표 사용자 동시성으로 주요 사용자 활동을 측정하는 기준 테스트부터 시작해야 합니다. 대체로 보고 및 Oracle Smart View for Office 사용자 활동과 통합 활동을 구분하기 위해 여러 개의 Financial Management 클러스터를 사용하는 경우 측정되는 사용자 태스크에 따라 각 조정 변경이 서버에 다른 영향을 미칩니다. 예를 들어 주로 보고에



사용되는 Financial Management 애플리케이션 서버는 NumConsolidationThreads를 늘려도 혜택이 없지만 많은 통합을 실행하는 서버는 통합 시간이 향상됩니다. 마찬가지로, 주로 보고에 사용되는 애플리케이션 서버는 MaxNumDataRecordsInRAM이 모든 레코드를 메모리에 유지할 만큼 충분히 높게 설정된 경우 반복 보고서의 응답 시간이 향상되지만 많은 통합을 실행하는 서버는 통합 시간이 향상되지 않습니다. 애플리케이션 서버의 역할을 결정할 때 고려할 다른 사항은 CPU 속도, CPU 코어 수, RAM 크기 및 RAM 속도입니다. 주로 통합 실행에 사용되며 집약 규칙이 있는 Financial Management 애플리케이션 하나를 실행하는 애플리케이션 서버는 일반적으로 CPU/코어를 더 많이 사용하지만 클럭 속도가 더 느린 경우보다 8개 이상 코어로 CPU 클럭 소도가 더 빠른 경우에 최상의 성능을 제공합니다.

## Financial Management 조정 예

이 예에서는 8개 CPU 및 64GB RAM이 있고 HFM 애플리케이션 3개를 호스트하며 3개 중 하나만 많이 사용되는 HFM 애플리케이션 서버를 조정하는 방법을 설명합니다. 이 애플리케이션을 CORP라고 합니다. Oracle Hyperion Financial Management 애플리케이션 서버는 통합 및 보고 활동 둘 다에 사용됩니다.

Financial Management 설치 시 기본적으로 표준 구성 설정이 제공됩니다. [설정] 모듈에서 해당 기본 설정을 보고 수정할 수 있습니다.

이 예에서는 MaxNumDataRecordsinRAM 값 합계를 60,000,000으로 유지하고 MaxDataCacheSizeinMB 값 합계를 9,000으로 유지해야 합니다. 세 애플리케이션 중 두 개는 많이 사용되지 않으므로 두 애플리케이션은 기본값을 사용하도록 허용하고 CORP 애플리케이션은 서버 리소스를 완전히 활용하도록 조정합니다.

MaxNumDataRecordsinRAM 기본값은 30,000,000이고 두 애플리케이션은 기본값을 사용하므로 CORP의 MaxNumDataRecordsinRAM은 40,000,000으로 설정될 수 있습니다.

MaxDataCacheSizeinMB 기본값은 4,500이고 두 애플리케이션은 기본값을 사용하므로 CORP의 MaxDataCacheSizeinMB는 6,000으로 설정될 수 있습니다.

이 예에서 CORP에 대해 특별히 조정을 고려하는 기타 값은 다음과 같습니다.

MinDataCacheSizeInMB – MaxDataCacheSizeinMB 값의 절반(3,000)으로 설정합니다.

CORP에 대한 애플리케이션 레벨을 조정하고 다른 두 애플리케이션은 기본값을 사용하도록 허용하므로 기본값이 아닌 다른 값을 사용하도록 CORP에 대한 대체를 생성합니다. 설정 업데이트 내용은 다음과 같습니다.

MaxNumDataRecordsinRAM (40,000,000)

MaxDataCacheSizeinMB (6,000)

MinDataCacheSizeinMB (4,000)

이러한 구성 설정을 업데이트하려면 **탐색, 관리, 통합 관리, 시스템, 설정** 순으로 선택합니다.

모든 구성 설정에 대한 자세한 내용은 [사용가능 구성 설정](#)을 참조하십시오.

대규모 애플리케이션 설정에 대한 자세한 내용은 [대규모 애플리케이션에 대한 Financial Management 메모리 설정](#)을 참조하십시오.

## 애플리케이션 데이터베이스 유지관리

다음 모범 사례는 Oracle Hyperion Financial Management와 함께 Oracle/SQL Server 데이터베이스를 사용하는 경우 권장됩니다.

Financial Management 테이블 <애플리케이션 이름>\_DATA\_AUDIT, <애플리케이션 이름>\_TASK\_AUDIT 및 HFM\_ERRORLOG의 경우 다음 모범 사례를 구현하는 것이 좋습니다.

분기별 - 감사 로그를 검토하고 아카이브 및 삭제하는 비즈니스입니다.

반기별 - 시스템 메시지를 아카이브하고 테이블을 자릅니다.

### 💡 팁:

이러한 테이블이 권장 레코드 수(500,000 초과)를 초과할 경우 조치를 취할 수 있도록 알림을 설정합니다. 감사 테이블이 크면 Financial Management 성능이 훨씬 저하될 수 있습니다.

## 기본 디자인 고려 사항

- 데이터 감사 기능이 비즈니스 요구사항에 포함되지 않는 경우 데이터 감사를 해제하는 것이 좋습니다. 데이터 감사 테이블이 10GB보다 큰 애플리케이션에서 성능 저하가 관찰되었습니다.

### 💡 팁:

멤버 감사를 해제하려면 메타데이터 파일에서 모든 멤버에 대해 EnableDataAudit 속성을 N으로 변경합니다.

- 잘못 설계된 규칙의 위험(예: 규칙에서 데이터 급증)을 방지하기 위해 프로덕션 환경에서 로드하기 전에 항상 규칙을 테스트해야 합니다. 효율적인 규칙은 허용되는 시스템 성능에 중요합니다.
- Oracle Hyperion Financial Management는 RAM에 저장된 동안 서브큐브를 통해 모든 처리를 수행하므로 서브큐브가 클수록 성능에 미치는 영향이 커집니다. 서브큐브 크기를 가능한 한 최소화합니다. 예를 들어 시스템 성능을 최적화하려면 서브큐브가 200,000개 기본 레코드 제한을 초과하지 않아야 합니다.
- Financial Management 애플리케이션에서 0을 계산하거나 로드하지 않는 것이 좋습니다. 0은 데이터로 저장되므로 데이터베이스 크기가 증가하고 성능이 저하될 수 있습니다. 숫자 정보(예: 1000)만 Financial Management에 저장하는 것이 좋습니다. *Oracle Hyperion Financial Management 사용자 가이드*의 데이터 파일 섹션을 참조하십시오.
- Financial Management가 제대로 종료되지 않은 경우 재부팅 시 임시 파일이 유지될 수 있습니다. 성능을 최적화하려면 Financial Management를 실행하기 전에 Financial Management 서버 작업 폴더에서 \*.db.\* 파일 이름을 모두 삭제하는 것이 좋습니다.
- 데이터 그리드 또는 프로세스 단위에 여러 개의 문서를 첨부하는 경우 한 셀에 첨부한 문서가 세 개를 넘지 않도록 하는 것이 좋습니다. 데이터베이스 성능 저하를 제한하려면 각 문서가 100K보다 작아야 합니다.

 **팁:**

애플리케이션을 생성할 때 문서 첨부 파일의 크기 제한과 문서 첨부 파일의 최대 개수를 사용자별로 설정할 수 있습니다. 애플리케이션 메타데이터의 AppSettings 속성에 제한을 설정할 수 있습니다.

- 적절한 상황에서만 [모두 통합] 옵션을 사용합니다. 이 옵션을 사용하면 시스템에서 데이터 없음 상태인 엔티티를 건너뛰지 않아 통합 성능이 훨씬 저하될 수 있습니다.

 **팁:**

[통합(영향 받는 통합)]은 논리 또는 통합이 필요한 엔티티만 시스템에서 업데이트되므로 가장 효율적인 옵션입니다. [데이터가 있는 항목 모두 통합] 옵션은 메타데이터가 변경된 후 시스템 상태를 OK SC에서 OK로 업데이트하는 데 유용합니다. 자세한 내용은 *Oracle Hyperion Financial Management 사용자 가이드*의 통합 옵션 섹션을 참조하십시오.

## Financial Management용 Oracle 11g 데이터베이스 조정

### 소개

Oracle Hyperion Financial Management에서 애플리케이션 데이터를 저장하려면 관계형 데이터베이스가 필요합니다. 각 Financial Management 애플리케이션에는 테이블 세트, 인덱스, 저장 프로시저 및 기타 객체가 포함되어 있습니다. 이러한 객체의 수와 크기는 사용자 데이터 세트, 애플리케이션 디자인 및 보고 요구사항에 따라 달라지므로 데이터베이스 설정에 대한 구체적인 규칙 세트를 지정하기 어렵습니다. 이 섹션에서는 Oracle 데이터베이스에 배포하는 중 발생하는 가장 일반적인 두 가지 이슈를 다룹니다.

- Oracle DB의 메모리가 부족하여 필요한 개수의 데이터베이스 연결을 지원할 수 없음
- 보고 및 통합 중 성능 저하

이러한 두 이슈는 부적절한 Oracle 매개변수 설정 및 구성으로 추적될 수 있습니다. 너무 큰 SGA(시스템 글로벌 영역)를 생성하면 사용자 연결 및 활동을 지원하는 데 사용 가능한 물리적 메모리량이 제한됩니다. 반대로, 너무 작은 SGA를 생성하면 추가 디스크 액세스가 발생하지만 성능 속도가 느려집니다.

이 섹션에서는 일반 데이터베이스를 모니터링하고 성능을 극대화하는 적절한 초기화 설정을 결정하는 프로세스를 안내합니다. 데이터 세트, 워크로드 및 애플리케이션 디자인의 변경사항을 반영하기 위해 주기적으로 이 프로세스를 반복해야 합니다.

Oracle 11g에서는 이전 버전보다 훨씬 쉽게 데이터베이스를 모니터링 및 조정할 수 있습니다. Oracle Enterprise Manager(데이터 제어 및 그리드 제어)를 사용하여 데이터베이스 성능을 모니터링, 진단 및 조정하는 것이 좋습니다. 정확한 인스턴스 통계를 얻으려면 Oracle 데이터베이스 자동 유지 관리 태스크를 활성화하는 것이 좋습니다.

데이터베이스 시작 후 즉시 조정을 수행하지 않는 것이 매우 중요합니다. 이때는 버퍼 캐시가 비어 있고 수집된 통계가 없습니다. 항상 Financial Management 애플리케이션에서 일정 기간 활동한 후 데이터베이스를 테스트하고 조정합니다.

## 일반적인 성능 이슈

가장 일반적인 성능 저하 원인은 부실 또는 누락된 테이블 통계입니다. Oracle Hyperion Financial Management 애플리케이션은 새 테이블을 생성하거나, 기존 테이블을 지우고 전체 데이터 세트를 즉시 재로드하거나 바꿀 수 있습니다. 기본적으로 Oracle 10g 및 11g는 야간작업을 실행하여 통계를 업데이트해야 하는 테이블을 확인합니다. 이것만으로 충분하지 않은 경우도 있습니다. 다음 활동은 스키마 통계를 업데이트해야 한다고 DBA에 알려야 하는 일반적인 일반 사용자 활동입니다.

- 새 HFM 애플리케이션을 생성 및 로드함
- 기존 시나리오의 데이터를 지우고 재로드함
- 새 시나리오/연도를 열고 데이터를 로드함. 여기에는 시나리오가 Financial Management 규칙에 의해 채워지는 것도 포함됩니다.

시나리오/연도에 첫 번째 달의 데이터가 로드된 후 후속 기간의 데이터 로드는 테이블 통계에 영향을 주지 않습니다.

## Oracle 11g 데이터베이스에 대한 조정 지침

### Oracle 초기화 매개변수

데이터베이스 성능을 향상시키기 위해 많은 초기화 매개변수를 미세 조정할 수 있습니다. 이 섹션에서는 Oracle 데이터베이스를 사용한 Oracle Hyperion Financial Management 성능에 크게 영향을 주는 것으로 알려진 매개변수에 대해 집중적으로 살펴봅니다.

### CURSOR\_SHARING

CURSOR\_SHARING은 동일한 커서를 공유할 수 있는 SQL 문의 유형을 결정합니다. 이 매개변수를 EXACT로 설정하면 동일한 텍스트가 있는 명령문만 동일한 커서를 공유할 수 있습니다. 이 매개변수를 FORCE로 설정하면 리터럴이 명령문의 의미에 영향을 주지 않는 경우 일부 리터럴이 다르지만 그 외에는 동일한 명령문이 SQL 커서를 공유하고 다시 사용할 수 있습니다. 테스트 결과, 이 매개변수를 FORCE 값으로 설정하면 통합 및 보고 성능이 상당히 향상될 수도 있습니다. 이 매개변수를 FORCE로 설정하면 Oracle 데이터베이스가 SQL 구문분석에 더 적은 시간을 사용하고 더 적은 메모리를 요구하기 때문입니다.

11.1.2.2.300 이전의 모든 릴리스에 대한 제안 설정: FORCE

11.1.2.2.300 이후의 릴리스에 대한 제안 설정: EXACT

### MEMORY\_TARGET

MEMORY\_TARGET 및 MEMORY\_MAX\_TARGET은 Oracle 데이터베이스 릴리스 11g의 새 매개변수 두 개입니다. 이러한 두 매개변수는 Oracle 데이터베이스에 대한 자동 메모리 관리 사용을 결정합니다. 자동 메모리 관리를 사용하여 시스템의 메모리를 관리하는 것이 좋습니다. 이러한 두 매개변수를 0이 아닌 값으로 설정하면 Oracle은 자동 메모리 관리를 활성화하고 대상 메모리 크기에 맞게 조정하여 필요에 따라 SGA(시스템 글로벌 영역) 및 인스턴스 PGA(프로그램 글로벌 영역) 간에 메모리를 재배포합니다. 따라서 다음 매개변수의 크기가 자동으로 지정됩니다.

SGA\_TARGET

SGA\_MAX\_SIZE

DB\_CACHE\_SIZE  
SHARED\_POOL\_SIZE  
LARGE\_POOL\_SIZE  
JAVA\_POOL\_SIZE  
STREAMS\_POOL\_SIZE  
PGA\_AGGREGATE\_TARGET

그러나 LOG\_BUFFER는 자동 메모리 관리의 영향을 받지 않으며 수동으로 크기를 지정해야 합니다. LOG\_BUFFER 조정은 이 섹션의 뒷부분에서 설명합니다.

MEMORY\_TARGET은 SGA 및 PGA의 총 메모리 크기를 지정하므로 더 나은 성능을 얻기 위해 비교적 높은 값으로 설정해야 합니다. Oracle Hyperion Financial Management 통합 및 보고는 메모리를 많이 사용하는 태스크이며 중복 메모리가 필요합니다. 32비트 운영 체제는 사용 가능한 주소 공간에 대한 제한(일반적으로 2GB 또는 3GB)이 있음을 고려할 때 이 매개변수를 1.2GB 이상으로 설정하는 것이 좋습니다. 일반적으로 값이 클수록 Oracle 데이터베이스 성능이 향상되므로 가상 주소 공간이 부족하지 않은 한 이 매개변수를 높게 설정하는 것이 좋습니다.

제안 설정: 최소 1.2GB, 일반적으로 1.2GB보다 높음(환경에 따라 달라짐).

Oracle 11g 인스턴스에 할당할 메모리 양을 결정할 때 [Oracle 데이터베이스 릴리스 11g에 대한 프로세스 수를 계산하는 방법](#) 섹션을 검토하십시오.

## MEMORY\_MAX\_TARGET

MEMORY\_TARGET은 DBA가 MEMORY\_MAX\_TARGET 매개변수를 설정할 수 있는 최대값을 지정합니다. 실수로 MEMORY\_TARGET을 너무 높게 설정할 수 없도록 하는 상한 역할을 합니다. 또한 다시 시작하지 않고 런타임에 MEMORY\_TARGET을 늘리려는 경우를 위해 Oracle 데이터베이스 인스턴스에 대한 메모리를 예약합니다. 따라서 MEMORY\_MAX\_TARGET은 MEMORY\_TARGET 이상이어야 합니다.

제안 설정: MEMORY\_TARGET 이상

## SGA\_TARGET

SGA\_TARGET은 모든 SGA 구성요소의 총 크기를 지정합니다. 자동 메모리 관리가 활성화되고 SGA\_TARGET이 0이 아닌 값으로 설정된 경우 이 값은 SGA의 최소 크기 역할을 합니다.

권장 설정: 자동 메모리 관리가 사용으로 설정된 경우 0입니다. 그렇지 않으면 [Oracle 데이터베이스 릴리스 11g에 대한 메모리 설정을 결정하는 방법](#)을 참조하십시오.

## SGA\_MAX\_SIZE

SGA\_MAX\_SIZE는 인스턴스 수명 동안 SGA의 최대 크기를 지정합니다. 이 매개변수는 SGA\_TARGET에 대한 상한을 설정합니다. 자동 메모리 관리가 활성화된 경우 Oracle 데이터베이스는 SGA\_MAX\_SIZE를 초과하여 SGA 구성요소의 총 크기를 늘릴 수 없습니다.

권장 설정: 자동 메모리 관리가 사용으로 설정된 경우 기본 설정을 사용합니다. 그렇지 않으면 [Oracle 데이터베이스 릴리스 11g에 대한 메모리 설정을 결정하는 방법](#)을 참조하십시오.

## PGA\_AGGREGATE\_TARGET

PGA\_AGGREGATE\_TARGET은 인스턴스에 연결된 모든 서버 프로세스가 사용할 수 있는 총 PGA 메모리를 지정합니다. 자동 메모리 관리가 활성화되고 PGA\_AGGREGATE\_TARGET이 0이 아닌 값으로 설정된 경우 이 값은 PGA의 최소 크기 역할을 합니다.

권장 설정: 자동 메모리 관리가 사용으로 설정된 경우 0입니다. 그렇지 않으면 [Oracle 데이터베이스 릴리스 11g에 대한 메모리 설정을 결정하는 방법](#)을 참조하십시오.

## LOG\_BUFFER

LOG\_BUFFER는 Oracle에서 재실행 로그 파일에 재실행 항목을 버퍼링할 때 사용하는 메모리 크기(바이트)를 지정합니다. 재실행 로그 항목에는 데이터베이스 블록 버퍼에 대한 변경사항의 레코드가 포함됩니다. Oracle Hyperion Financial Management는 높은 업데이트 트랜잭션 시스템이며 데이터베이스가 지속적으로 로그 버퍼를 사용합니다. 로그 버퍼의 크기를 제대로 조정하면 데이터베이스 성능을 향상시킬 수 있습니다. 일반적으로 LOG\_BUFFER의 값이 클수록 특히 트랜잭션이 길거나 다수인 경우 재실행 로그 파일 I/O가 감소합니다. 버퍼가 너무 작으면 시스템이 새 업데이트를 추가하기 전에 버퍼가 지워질 때까지 기다리므로 데이터베이스 성능을 향상시키려면 이 버퍼를 정확하게 설정하는 것이 중요합니다.

제안 설정: 8MB로 시작합니다. [Oracle 데이터베이스 릴리스 11g에 대한 메모리 설정을 결정하는 방법](#)을 참조하십시오.

## OPTIMIZER\_MODE

OPTIMIZER\_MODE는 인스턴스에 대한 최적화 방법을 선택하는 기본 동작을 설정합니다. 최적 온라인 애플리케이션 응답을 위해 최적화 프로그램 모드를 FIRST\_ROWS로 설정하거나 बै치 작업의 총 실행 시간을 최소화하기 위해 ALL\_ROWS로 설정할 수 있습니다. Oracle Hyperion Financial Management는 반환된 총 데이터 세트만 처리하므로 총 실행 시간을 최소화하는 것이 더 적절합니다.

제안 설정: ALL\_ROWS

## OPTIMIZER\_INDEX\_COST\_ADJ

OPTIMIZER\_INDEX\_COST\_ADJ를 사용하면 액세스 경로 선택에 대한 최적화 프로그램 동작을 다소 인덱스 친화적으로 조정할 수 있습니다. 즉, 최적화 프로그램이 전체 테이블 스캔보다 인덱스 액세스 경로를 선택하도록 할 수 있습니다. 이 매개변수의 기본값은 100%로, 이때 최적화 프로그램은 일반 비용으로 인덱스 액세스 경로를 평가합니다. 그러나 Oracle Hyperion Financial Management 트랜잭션은 일반적으로 전체 테이블 스캔 경로보다 인덱스 액세스 경로를 더 선호하므로 이 매개변수의 값을 줄이는 것이 좋습니다.

제안 설정: 50

## PROCESSES

PROCESSES는 동시에 Oracle 데이터베이스에 연결할 수 있는 최대 운영 체제 사용자 프로세스 수를 지정합니다. Oracle Hyperion Financial Management는 전용 서버에서만 작동하므로 각 연결에 프로세스가 필요합니다. 각 Financial Management 애플리케이션에는 Financial Management 데이터베이스 연결 풀 설정으로 지정된 개수 외에도 최소 200개(다중 서버 환경의 경우 400개)의 데이터베이스 연결이 있어야 합니다.

제안 설정: [Oracle 데이터베이스 릴리스 11g에 대한 프로세스 수를 계산하는 방법](#)을 참조하십시오.

## SESSIONS

이 매개변수는 데이터베이스 시스템에 생성할 수 있는 최대 세션 수를 지정합니다. 모든 로그인에 세션이 필요하므로 이 매개변수는 실제로 Oracle 데이터베이스의 최대 동시 사용자 수를 결정합니다. 기본값은  $1.1 * PROCESSES + 5$ 입니다. 이 매개변수를 기본값 아래로 설정하지 않는 것이 좋습니다.

## TRANSACTIONS

TRANSACTIONS는 최대 동시 트랜잭션 수를 지정합니다. 일부 트랜잭션은 반복적일 수 있으므로 반복적인 트랜잭션을 허용하기 위해 이 매개변수는 SESSIONS보다 크고, 결과적으로 PROCESSES보다 커야 합니다. 기본값은  $1.1 * SESSIONS$ 입니다. 이 매개변수를 기본값 아래로 설정하지 않는 것이 좋습니다.

## OPEN\_CURSORS

OPEN\_CURSORS는 세션에서 동시에 열어 둘 수 있는 최대 커서(개인 SQL 영역에 대한 핸들) 수를 지정합니다. 애플리케이션에 열린 커서가 부족하지 않도록 OPEN\_CURSORS 값을 충분히 높게 설정하는 것이 중요합니다. 세션에서 OPEN\_CURSORS로 지정된 개수의 커서를 열지 않는다고 가정하면 이 값을 실제 필요한 것보다 높게 설정하기 위한 추가 오버헤드는 없습니다.

제안 설정: 5000

## SESSION\_CACHED\_CURSORS

SESSION\_CACHED\_CURSORS는 캐시할 세션 커서 수를 지정합니다. 동일한 SQL 문의 구문분석 호출을 반복하면 해당 명령문의 세션 커서가 세션 커서 캐시로 이동합니다. 후속 구문분석 호출은 캐시에서 커서를 찾으려 커서를 다시 열 필요가 없습니다. Financial Management 연결도 캐시되므로 이 캐시를 통해 Oracle Hyperion Financial Management 애플리케이션의 성능이 향상됩니다.

제안 설정: 50

## TRACE\_ENABLED

TRACE\_ENABLED는 Oracle 데이터베이스의 실행 내역 또는 코드 경로 추적을 제어합니다. 매개변수를 TRUE로 설정하여 이 옵션을 활성화하면 데이터베이스에 오버헤드가 더 추가되며 일반 Financial Management 애플리케이션 환경에서는 권장되지 않습니다.

제안 설정: FALSE

## STATISTICS\_LEVEL

STATISTICS\_LEVEL은 데이터베이스 및 운영 체제에 대한 통계 수집 레벨을 지정합니다. Oracle 데이터베이스는 자체 관리 결정을 비롯한 다양한 목적으로 이러한 통계를 수집합니다. 기본 설정 TYPICAL은 데이터베이스 자체 관리 기능에 필요한 모든 주요 통계 수집을 보장하며 전반적인 최적 성능을 제공합니다.

제안 설정: TYPICAL

## TIMED\_STATISTICS

TIMED\_STATISTICS는 시간과 관련된 통계를 수집할지 여부를 지정합니다. Oracle 데이터베이스 릴리스 11.1.0.7.0부터 STATISTICS\_LEVEL의 값이 TYPICAL 또는 ALL로 설정된 경우 TIMED\_STATISTICS 매개변수의 값을 FALSE로 설정할 수 없습니다.

제안 설정: TRUE

## TIMED\_OS\_STATISTICS

TIMED\_OS\_STATISTICS는 클라이언트가 서버에 요청할 때 또는 요청이 완료될 때 Oracle에서 운영 체제 통계를 수집하는 간격(초)을 지정합니다. 매개변수를 0보다 큰 숫자로 설정하여 이 옵션을 활성화하면 애플리케이션의 성능이 훨씬 저하됩니다.

제안 설정: 0

## Oracle 데이터베이스 릴리스 11g에 대한 메모리 설정을 결정하는 방법

이 섹션에서는 Oracle 시스템 관련 통계를 모니터 및 보고 Oracle 데이터베이스 메모리 매개변수를 조정하는 방법을 간략하게 설명합니다. 최적 메모리 설정을 결정하는 많은 방법이 있지만 Memory Advisor, SGA Advisor, Shared Pool Advisor, Buffer Cache Advisor, PGA Advisor 등의 메모리 관리자를 사용하는 것이 좋습니다. 이러한 관리자를 사용하고 다음 태스크를 수행하려면 DBA 권한을 가진 Oracle 로그인에 있어야 합니다. 아래 쿼리는 대부분 Oracle Enterprise Manager를 통한 동등한 그래픽 인터페이스가 있습니다.

### 총 메모리 크기 조정(MEMORY\_TARGET)

MEMORY\_TARGET은 SGA와 PGA를 모두 포함하여 Oracle 시스템 전체에서 사용 가능한 메모리를 지정합니다. Oracle 데이터베이스 릴리스 11g 이전에는 SGA와 PGA를 별도로 조정해야 했습니다.

Oracle 10g에서 11g로 데이터베이스를 업그레이드한 경우 Oracle 10g 데이터베이스의 SGA\_TARGET 및 PGA\_AGGREGATE\_TARGET을 추가하면 MEMORY\_TARGET을 간단하게 확인할 수 있습니다.

Oracle 9i에서 11g로 데이터베이스를 업그레이드한 경우 DB\_CACHE\_SIZE, SHARED\_POOL\_SIZE, LARGE\_POOL\_SIZE, JAVA\_POOL\_SIZE 등의 모든 SGA 구성요소와 PGA\_AGGREGATE\_TARGET을 추가하면 MEMORY\_TARGET을 확인할 수 있습니다.

이전 버전에서 데이터베이스를 업그레이드하지 않았으며 내역 참조가 없는 경우 시스템 리소스 및 시스템 제한에 따라 처음에 이 매개변수를 1-3GB로 설정하는 것이 좋습니다. 이



매개변수를 데이터베이스를 사용 중입니다 잠시 동안 후에는 다음과 같이 조정할 수 있습니다. 이 조정은 위의 두 업그레이드 시나리오에도 적용됩니다.

```
SQL> select * from v$memory_target_advice order by memory_size;
```

```
MEMORY_SIZE MEMORY_SIZE_FACTOR ESTD_DB_TIME ESTD_DB_TIME_FACTOR VERSION

180 .5 458 1.344 0

270 .75 367 1.0761 0

360 1 341 1 0

450 1.25 335 .9817 0

540 1.5 335 .9817 0

630 1.75 335 .9817 0

720 2 335 .9817 0
```

MEMORY\_SIZE\_FACTOR가 1인 행은 MEMORY\_TARGET 초기화 매개변수로 설정된 현재 메모리 크기와 현재 워크로드를 완료하는 데 필요한 DB 시간을 표시합니다. 이전 및 이후 행의 결과는 많은 대체 MEMORY\_TARGET 크기를 보여 줍니다. 각 대체 크기에 대해 데이터베이스는 크기 요소(현재 크기의 배수) 및 MEMORY\_TARGET 매개변수가 대체 크기로 변경될 경우 현재 워크로드를 완료하는 예상 DB 시간을 보여 줍니다. 총 메모리 크기가 현재 MEMORY\_TARGET 크기(이 예에서는 360)보다 작은 경우 예상 DB 시간(ESTD\_DB\_TIME)이 증가합니다. 또한 이 예에서 총 메모리 크기를 450MB 이상 늘릴 경우 ESTD\_DB\_TIME 값이 감소하지 않으므로 얻어지는 이득이 없습니다. 따라서 이 예에서 제안된 MEMORY\_TARGET 크기는 450 지정됩니다.

## SGA 크기 조정(SGA\_TARGET)

일반적으로 자동 메모리 관리가 활성화된 경우 SGA는 Oracle 데이터베이스에 의해 자동으로 조정됩니다. 그러나 DBA가 계속 SGA 크기를 모니터하고 최적 크기인지 확인할 수 있습니다.

```
SQL> select * from v$sga_target_advice order by sga_size;
```

```
SGA_SIZE SGA_SIZE_FACTOR ESTD_DB_TIME ESTD_DB_TIME_FACTOR  
ESTD_PHYSICAL_READS
```

```
290 .5 448176 1.6578 1636103
```

```
435 .75 339336 1.2552 1636103
```

```
580 1 201866 1 513881
```

```
725 1.25 201866 1 513881
```

```
870 1.5 201866 1 513881
```

```
1015 1.75 201866 1 513881
```

```
1160 2 201866 1 513881
```

[총 메모리 크기 조정] 섹션의 유사한 분석에 따라 SGA\_TARGET의 현재 설정은 이미 최적 상태입니다.

## PGA 크기 조정(PGA\_AGGREGATE\_TARGET)

SGA와 마찬가지로, 자동 메모리 관리가 활성화된 경우 PGA도 Oracle 데이터베이스에 의해 자동으로 조정됩니다. 다음 쿼리를 사용하여 PGA 크기가 제대로 조정되었는지 모니터할 수 있습니다. 결과는 v\$memory\_target\_size 및 v\$sga\_target\_size의 쿼리 결과와 유사하게 표시됩니다.

```
SQL> select * from v$pga_target_advice order by  
pga_target_for_estimate;
```

## LOG\_BUFFER 크기 조정

시스템 뷰 v\$sysstat에서 재실행 버퍼 할당 재시도 값은 사용자 프로세스가 재실행 로그 버퍼의 공간을 기다리는 횟수를 반영합니다. 크기가 제대로 조정된 데이터베이스의 경우 이 값이 0에 가까워야 합니다. 예:

```
select name, value

from v$sysstat

where name = 'redo buffer allocation retries'

NAME VALUE

redo buffer allocation retries 1021967
```

로그 버퍼에 업데이트에 사용할 공간이 없는 경우 데이터베이스가 대기한 후 재시도해야 합니다. 이 예에서는 데이터베이스가 총 1,021,967회를 재시도했습니다. 성능을 향상시키려면 LOG\_BUFFER 매개변수 값을 늘립니다. 이 값은 바이트 단위로 표시되며 운영 체제 블록 크기인 로그 블록 크기 값의 배수여야 합니다. Oracle Hyperion Financial Management 애플리케이션의 경우 LOG\_BUFFER를 최소 8MB로 설정한 다음 위의 쿼리를 사용하여 성능을 모니터링하고 필요에 따라 늘립니다. LOG\_BUFFER를 늘려야 하는 경우 한 번에 50%씩 증가하는 것이 좋습니다.

## Oracle 데이터베이스 릴리스 11g에 대한 프로세스 수를 계산하는 방법

동시에 Oracle 데이터베이스에 연결할 수 있는 사용자 프로세스 수는 Oracle 데이터베이스 인스턴스의 PROCESSES 초기화 매개변수로 제한됩니다. 기본적으로 각 Oracle Hyperion Financial Management 애플리케이션 프로세스에는 단일 서버 환경에 대해 최소 200개의 데이터베이스 연결이 있어야 합니다. 두 개 이상의 Financial Management 서버가 환경에 대해 구성된 경우 애플리케이션 무결성을 보장하려면 클러스터 컨트롤러 연결에 대한 추가 풀이 있어야 합니다. 클러스터 컨트롤러 풀은 애플리케이션 풀과 같으므로 DBA는 애플리케이션 프로세스당 400개의 연결을 처리해야 합니다.

### 주:

클러스터의 총 서버 수 및 총 애플리케이션 수는 필요한 데이터베이스 연결 수에 영향을 줍니다.

이 예에서는 Oracle 데이터베이스에 연결하는 프로세스 수를 계산하는 방법을 보여 줍니다. 하나의 Financial Management 애플리케이션 서버에 두 개의 애플리케이션이 있다고 가정합니다. 기본 Financial Management 데이터베이스 연결 풀 설정은 200입니다. Financial Management에만 필요한 최소 Oracle 데이터베이스 연결 수는 400입니다. 안전을 위해 이 수치에 1.1 계수를 곱하여 보조 연결 및 일반적인 데이터베이스 사용을 허용합니다. Oracle 데이터베이스에는 일부

백그라운드 프로세스도 있음을 고려하여 해당 수치에 20을 더해 PROCESSES 값을 구합니다. 따라서 이 경우 PROCESSES를 460으로 설정해야 합니다.

단일 HFM 애플리케이션 서버의 경우:

$PROCESSES = (\text{Financial Management 연결 풀 설정}) * (\text{Financial Management 애플리케이션 수}) * 1.1 + 20$

각 서버에서 두 Financial Management 애플리케이션 서버의 클러스터에 두 개의 애플리케이션이 있다고 가정합니다. Financial Management 데이터베이스 연결 풀 설정은 200이고 클러스터 컨트롤러 풀도 200입니다. 이제 Financial Management에만 필요한 최소 Oracle 데이터베이스 연결 수는  $(200+200)*2*2=1600$ 이므로 1,600입니다.

다중 서버 환경의 경우:

$PROCESSES = (\text{Financial Management 연결 풀 설정} + \text{Financial Management 클러스터 컨트롤러 연결 풀 설정}) * (\text{Financial Management 애플리케이션 수}) * (\text{Financial Management 서버 수}) * 1.1 + 20.$

## 기타 고려 사항

### 공유 서버 및 전용 서버

Oracle Hyperion Financial Management는 Oracle 데이터베이스에 대한 모든 연결이 전용 서버 프로세스에서 제공되어야 합니다. 즉, Financial Management는 공유 서버 프로세스에서 작동하지 않습니다. 전용 서버 프로세스는 CPU와 메모리 리소스를 더 많이 사용하지만 성능을 향상시킵니다. 전용 서버를 사용하려면 네트 서비스 이름 값의 연결 기술자에 SERVER=DEDICATED 절을 포함해야 합니다. 이는 전용 서버 프로세스에 대해 구성된 네트 서비스의 예입니다.

```
HFMDB = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST =
hfm.oracle.com) (PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED)
(SERVICE_NAME = HFMDB1) ) )
```

### 온라인 재실행 로그 파일 크기

데이터베이스 작성기 및 아카이버 프로세스의 동작은 재실행 로그 크기에 따라 달라지므로 재실행 로그 파일의 크기는 성능에 영향을 줄 수 있습니다. 일반적으로 재실행 로그 파일이 클수록 성능이 향상됩니다. 로그 파일의 크기를 너무 낮게 지정하면 체크포인트 활동이 증가하고 성능이 감소합니다. 그러나 로그 파일이 작고 체크포인트가 빈번하면 복구 시간이 감소합니다. 따라서 일별 운영 효율성이 복구 시간 최소화보다 중요한 경우 온라인 재실행 로그 파일을 비교적 큰 값으로 설정합니다. Oracle Hyperion Financial Management 데이터베이스의 경우 수백 MB가 일반적인 크기입니다. 그러나 FAST\_START\_MTTT\_TARGET을 활성화하고 잠시 동안 일반적인 데이터베이스 워크로드를 실행하여 재실행 로그 파일의 크기를 결정하는 것이 좋습니다. 그런 후에 다음 쿼리를 실행하여 재실행 로그의 최적 크기를 가져옵니다.

```
SQL> select optimal_logfile_size from v$instance_recovery;
```

MTTR 대상 및 온라인 재실행 파일 크기를 조정하는 방법에 대한 자세한 내용은 *Oracle Database Performance Tuning Guide*를 참조하십시오.

## 테이블 영역 및 세그먼트 조각화

시간 경과에 따라 테이블 영역 내에서 객체를 업데이트하고 삭제할 때 개별적으로 새 데이터에 다시 사용할 수 있을 만큼 크지 않은 빈 공간 포켓이 생성될 수 있습니다. 이 유형의 빈 공간을 조각화된 여유 공간이라고 합니다. 조각화된 여유 공간이 있는 객체로 인해 많은 공간이 낭비되고 데이터베이스 성능이 저하될 수 있습니다. Oracle Hyperion Financial Management 통합은 광범위한 업데이트, 삽입 및 삭제를 수행하므로 테이블 영역의 조각화를 모니터링하고 주기적으로 테이블 영역을 조각 모음하는 것이 중요합니다. 이 공간을 조각 모음하고 회수하려면 온라인 세그먼트 축소를 수행하는 것이 좋습니다. 온라인 세그먼트를 사용하는 방법에 대한 자세한 내용은 Oracle Database Administrator's Guide를 참조하거나 Oracle 데이터베이스 지원 서비스에 문의하십시오.

## 인덱스 조각화

Oracle Hyperion Financial Management 애플리케이션은 일반적으로 수백 또는 수천 개의 인덱스를 생성합니다. 시간 경과에 따라 애플리케이션 데이터가 변경되면서 인덱스가 조각화될 수 있습니다. 이러한 인덱스를 주기적으로 모니터 및 조각 모음하면 성능이 향상될 수 있습니다. 그러나 인덱스 재작성은 시간이 걸리며 리소스를 많이 사용하는 작업입니다. 애플리케이션이 작동하는 동안에는 인덱스를 재작성하지 않는 것이 좋습니다. Oracle Enterprise Manager는 인덱스 통계를 모니터링하는 사용자 친화적 인터페이스를 제공합니다. Enterprise Manager를 통해 인덱스를 모니터링하고 조각 모음하는 방법에 대한 자세한 내용은 Oracle 데이터베이스 설명서를 참조하십시오.

## DEFERRED\_SEGMENT\_CREATION 기능을 비활성화합니다.

DEFERRED\_SEGMENT\_CREATION 기능은 릴리스 11.2에서 도입되었습니다. 모든 설치에서 기본 설정은 켜져 있습니다. 이 기능은 TABLE create 문이 실제로 테이블을 생성하지 않도록 합니다. 데이터 행을 삽입한 후에만 테이블이 생성됩니다.

## DEFERRED\_SEGMENT\_CREATION 기능을 비활성화합니다.

DEFERRED\_SEGMENT\_CREATION 기능은 릴리스 11.2에서 도입되었습니다. 모든 설치에서 기본 설정은 켜져 있습니다. 이 기능을 사용하면 TABLE create 문이 실제로 테이블을 생성하지 않습니다. 데이터 행을 삽입한 후에만 테이블이 생성됩니다. 임포트 중 일부 테이블이 생성되지 않을 수 있으므로 Oracle Hyperion Financial Management 스키마 익스포트 및 임포트 시 이 기능이 이슈를 발생시킬 수 있습니다. 이 기능은 사용 안함으로 설정하여 테이블이 자동으로 생성되도록 하는 것이 좋습니다. 이 기능을 비활성화하려면 SYSTEM 또는 SYS를 사용하여 해당 인스턴스에 로그인하고 다음 명령을 실행합니다.

```
alter system set deferred_segment_creation=false;
```

이 명령문을 실행한 후 생성되는 테이블은 모두 자동으로 생성됩니다. 빈 테이블을 포함하는 인스턴스가 이미 있고 애플리케이션을 내보내려는 경우 개별적으로 각 테이블을 변경하여 강제로 생성하고 EXP(익스포트) 명령에서 테이블을 사용하도록 허용할 수 있습니다.

스키마에 빈 테이블이 있는지 확인하려면 다음 두 명령 중 하나를 실행합니다.

```
select segment_name, segment_type, extents from dba_segments where
extents < 1 and segment_type='TABLE' and owner='<hfm db schema>'
```

```
select table_name from all_tables where owner='<hfm db schema>' and
table_name not in (select segment_name from dba_segments where
owner='<hfm db schema>' and segment_type='TABLE' and extents>0)
```

비어 있는 각각의 테이블에 대해 다음 명령을 실행하십시오:

```
alter table <table_name> allocate extent
```

## 정기 유지 관리 및 조정 계획

이전 섹션에서는 Oracle 메모리 매개변수의 크기를 정확하게 지정하는 일반적인 프로세스에 대해 간략하게 설명합니다. 기본적으로 성능 조정은 반복적입니다. 하나의 성능 블록을 제거해도 하면 다른 블록이 표시될 수 있으므로 즉시 성능이 향상되지 않을 수도 있습니다. 따라서 성능이 허용 가능할 때까지 이 프로세스를 반복해야 합니다. Oracle Hyperion Financial Management 애플리케이션 데이터는 기간마다 지속적으로 변경되기 때문에 정기 데이터베이스 유지관리 및 조정 계획은 사용자가 Oracle 데이터베이스 성능을 사전에 모니터 및 조정하고 이후의 잠재적 성능 이슈를 방지하는 데 도움이 됩니다. 자세한 내용과 추가 조정 옵션은 Oracle 데이터베이스 지원을 참조하십시오.

## 자주 묻는 질문(FAQ)

**지원되는 운영 체제는 무엇입니까?**

Oracle Enterprise Performance Management System Supported Platform Matrix([Oracle EPM Supported Platform Matrix](#))를 참조하십시오.

**관계형 데이터베이스는 64비트여야 합니까?**

지원되는 DBMS 유형 및 버전이지만 하면 데이터베이스는 32비트 또는 64비트일 수 있습니다.

타사 및 확장 소프트웨어를 고려해야 합니다. 기본적으로 Oracle Hyperion Financial Management 설치하는 64비트 운영 체제에 64비트 소프트웨어만 설치합니다. 즉, Financial Management 애플리케이션 서버에는 64비트 클라이언트만 구성요소만 설치됩니다. 32비트 연결이 필요한 경우 64비트 애플리케이션 서버에 32비트 클라이언트 소프트웨어를 설치하지 않으면 구성요소가 작동하지 않을 수도 있습니다. 자세한 내용은 *Oracle Enterprise Performance Management System 설치 및 구성 가이드*를 참조하십시오.

**64비트 Financial Management의 메모리 제한 사항은 무엇입니까?**

실제로 64비트 Financial Management는 가상 메모리가 아니라 물리적 메모리로 제한됩니다. 메모리 매개변수를 제대로 조정해야 사용 가능한 모든 물리적 메모리를 활용할 수 있습니다.

**Financial Management에 대해 조정해야 하는 메모리 설정이 있습니까?**

Financial Management 작은 32비트 환경에서 중간 크기 애플리케이션에 대한 기본 메모리 설정이 적합하지 확인합니다. 사용 가능한 메모리를 완전히 활용하려면 월별

애플리케이션에 대해 다음 설정을 사용하는 것이 좋습니다. 관련 설정은 MaxNumDataRecordsInRAM 및 MaxDataCacheSizeinMB로, 변경해야 합니다. 다음 표에는 사용 가능한 메모리에 따라 이러한 매개변수에 대한 제안 값이 포함되어 있습니다. 이 표에서는 Financial Management가 시스템에서 실행되는 메모리를 많이 사용하는 유일한 프로세스이며, 하나의 Financial Management 애플리케이션만 실행 중이라고 가정합니다. 여러 애플리케이션이 활성화된 경우 서버에 설치된 총 물리적 메모리를 Financial Management 애플리케이션 수로 나누어 각 애플리케이션에 사용 가능한 물리적 메모리를 결정합니다.

사용가능 물리적 메모리	MaxNumDataRecordsinRAM	MaxDataCacheSizeinMB
4	4,000,000	500
8	10,000,000	1500
16	30,000,000	4500
32	60,000,000	9000

예: 24GB RAM이 있고 월별 Financial Management 애플리케이션 2개가 활성화된 서버에서 MaxNumDataRecordsInRAM 값은 22,500,000이고 MaxDataCacheSizeinMB 값은 3375여야 합니다.

평균적으로 월별 애플리케이션에서 하나의 데이터 레코드에 사용되는 메모리 크기는 112바이트이고, 주별 애플리케이션은 데이터 레코드당 472바이트를 사용하고, 일별 애플리케이션은 데이터 레코드당 3,296바이트를 사용합니다.

주별 애플리케이션의 경우 마지막 열의 MaxDataCacheSizeinMB 값을 변경하지 않고 MaxNumDataRecordsInRAM을 4로 나눕니다.

일별 애플리케이션의 경우 마지막 열의 MaxDataCacheSizeinMB 값을 변경하지 않고 MaxNumDataRecordsInRAM을 30으로 나눕니다.