

# Oracle® Hyperion Calculation Manager 디자이너 가이드



릴리스 11.2.x  
F26657-04  
2022년 10월

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Hyperion Calculation Manager 디자이너 가이드NOT\_SUPPORTED 릴리스 11.2.x

F26657-04

Copyright © 2008, 2022, 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 Calculation Manager 개요

---

Calculation Manager 정보	1-1
Calculation Manager를 사용하는 서비스	1-2
역할이 Calculation Manager에서 수행할 수 있는 작업에 영향을 주는 방식	1-2
Calculation Manager 실행	1-3
Calculation Manager에서 객체 보기	1-3
뷰의 열 사용자정의	1-4
객체 필터링	1-4
쿼리를 사용하여 객체 필터링	1-4
규칙, 구성요소 또는 템플리트의 그래픽 그림 보기	1-5

## 2 비즈니스 규칙 디자인

---

비즈니스 규칙 정보	2-2
비즈니스 규칙 생성	2-3
매개변수로 사용할 실행 변수 선택	2-6
매개변수를 비즈니스 규칙으로 사용	2-7
비즈니스 규칙 열기	2-8
비즈니스 규칙 편집	2-8
스크립트 모드로 비즈니스 규칙 편집	2-9
비즈니스 규칙 저장	2-11
비즈니스 규칙 저장	2-12
비즈니스 규칙을 다른 이름으로 저장	2-12
비즈니스 규칙 실행	2-13
비즈니스 규칙 복사	2-13
비즈니스 규칙에서 검색	2-13
비즈니스 규칙 스크립트에서 텍스트 문자열 검색	2-14

그래픽 비즈니스 규칙에서 텍스트 검색 및 바꾸기	2-14
비즈니스 규칙 인쇄	2-14
비즈니스 규칙 삭제	2-15
비즈니스 규칙 구성요소의 공통 차원 정의	2-15
비즈니스 규칙 분석 및 디버깅	2-16
비즈니스 규칙 분석	2-16
추적된 차원의 멤버 숨기기	2-17
통계 정보 캡처	2-17
비즈니스 규칙의 스크립트 분석	2-18
비즈니스 규칙의 스크립트 비교	2-18
변경된 Business Rule을 저장된 Business Rule과 비교	2-19
비즈니스 규칙 디버깅	2-20
비즈니스 규칙에서 구성요소 사용 안 함	2-21
사전 정의된 Workforce Planning, Capital Asset Planning 및 Human Capital Management	
비즈니스 규칙 로드	2-21
비즈니스 규칙에 대한 바로가기 생성	2-22
비즈니스 규칙 또는 비즈니스 규칙 세트 새로고침	2-22
비즈니스 규칙 또는 규칙 세트에 대한 사용 표시	2-23
객체의 소유자 변경	2-23

### 3 비즈니스 규칙 세트 디자인

---

비즈니스 규칙 세트 정보	3-1
비즈니스 규칙 세트 생성	3-2
비즈니스 규칙 세트 열기	3-4
비즈니스 규칙 세트 내의 비즈니스 규칙 열기	3-4
비즈니스 규칙 세트에 비즈니스 규칙 추가	3-4
비즈니스 규칙 세트에서 비즈니스 규칙 제거	3-5
비즈니스 규칙 세트를 다른 애플리케이션에 복사	3-5
비즈니스 규칙 세트 저장	3-6
비즈니스 규칙 세트 삭제	3-6

### 4 시스템 템플릿 작업

---

시스템 템플릿 정보	4-1
시스템 템플릿 표시	4-2
시스템 템플릿 사용	4-2
데이터 지우기 템플릿 사용	4-3
데이터 복사 템플릿 사용	4-4
금액-수량-단가 템플릿 사용	4-5
할당 - 레벨별 템플릿 사용	4-6

할당 템플리트 사용	4-8
집계 템플리트 사용	4-9
SET 명령 템플리트 사용	4-11
통화 변환 템플리트 사용	4-12
통화 변환 템플리트 예	4-14
데이터 익스포트-임포트 템플리트 사용	4-17
템플리트 순서도 표시	4-18
시스템 템플리트를 사용자정의 템플리트로 저장	4-19
비즈니스 규칙에서 시스템 템플리트 제거	4-20

## 5 사용자정의 템플리트 작업

---

사용자정의 템플리트 정보	5-1
사용자정의 템플리트 생성	5-2
그래픽 사용자정의 템플리트 생성	5-2
UpperPOV 디자인 타임 프롬프트를 사용하는 그래픽 템플리트를 생성하는 예	5-4
스크립트 사용자정의 템플리트 생성	5-5
사용자정의 템플리트에 대한 디자인 타임 프롬프트 생성	5-5
속성 DTP	5-6
부울 DTP	5-7
횡차원 DTP	5-7
조건 DTP	5-8
차원 DTP	5-9
차원 DTP	5-10
멤버 DTP	5-11
멤버 DTP	5-12
멤버 범위 DTP	5-13
숫자 DTP	5-15
비밀번호 DTP	5-15
제한된 목록 DTP	5-16
구분자 DTP	5-16
문자열 DTP	5-17
디자인 타임 프롬프트에 대한 단계 생성	5-18
디자인 타임 프롬프트에 대한 종속성 정의	5-19
횡차원, 차원, 차원(둘 이상), 멤버 및 멤버 범위 디자인 타임 프롬프트에 대한 차원 한도 정의	5-20
디자인 타임 프롬프트에서 텍스트 찾기 및 바꾸기	5-21
사용자정의 템플리트 열기	5-21
사용자정의 템플리트 새로고침	5-22
사용자정의 템플리트에 대한 사용 표시	5-22
사용자정의 템플리트 복사 및 붙여넣기	5-22
사용자정의 템플리트 삭제	5-23

## 6 구성요소를 사용하여 비즈니스 규칙 및 템플릿 디자인

---

구성요소 정보	6-2
공식 구성요소	6-2
공식 구성요소 정보	6-3
공식 구성요소 생성	6-3
공식 구성요소 디자인	6-4
조건 빌더를 사용하여 조건문 생성	6-8
공식문에 대한 로깅 정보 입력(Financial Management 사용자만 해당)	6-10
공식문에 대한 설명 입력	6-11
공식 구성요소 열기	6-11
공식 구성요소 편집	6-12
공식 구성요소 삭제	6-12
공식 구성요소 복사 및 붙여넣기	6-13
스크립트 구성요소	6-13
스크립트 구성요소 생성	6-14
스크립트 구성요소 디자인	6-14
스크립트 구성요소 열기	6-18
스크립트 구성요소 편집	6-18
스크립트 구성요소 삭제	6-19
스크립트 구성요소 복사 및 붙여넣기	6-19
조건 구성요소	6-20
조건 구성요소 정보	6-21
조건 구성요소 생성	6-21
조건 구성요소 열기	6-22
조건 구성요소 편집	6-22
조건 구성요소 삭제	6-23
조건 구성요소 복사 및 붙여넣기	6-24
멤버 블록 구성요소	6-24
멤버 블록 구성요소 정보	6-25
멤버 블록 구성요소 생성	6-25
멤버 블록 구성요소 열기	6-26
멤버 블록 구성요소 편집	6-26
멤버 블록 구성요소 삭제	6-26
멤버 블록 구성요소 복사 및 붙여넣기	6-27
멤버 범위 구성요소	6-27
멤버 범위 구성요소 정보	6-28
멤버 범위 구성요소 생성	6-28
멤버 범위 구성요소 열기	6-30

멤버 범위 구성요소 편집	6-31
멤버 범위 구성요소 삭제	6-31
멤버 범위 구성요소 복사 및 붙여넣기	6-32
데이터 범위 구성요소(Financial Management 사용자만 해당)	6-32
데이터 범위 구성요소 정보(Financial Management 사용자만 해당)	6-33
데이터 범위 구성요소 생성(Financial Management 사용자만 해당)	6-33
데이터 범위 구성요소 열기(Financial Management 사용자만 해당)	6-35
데이터 범위 구성요소 편집(Financial Management 사용자만 해당)	6-35
데이터 범위 구성요소 삭제(Financial Management 사용자만 해당)	6-35
데이터 범위 구성요소 복사 및 붙여넣기(Financial Management 사용자만 해당)	6-36
셀 텍스트 범위 구성요소(Financial Management 사용자만 해당)	6-36
셀 텍스트 범위 구성요소 정보(Financial Management 사용자만 해당)	6-37
셀 텍스트 범위 구성요소 생성(Financial Management 사용자만 해당)	6-37
셀 텍스트 범위 구성요소 열기(Financial Management 사용자만 해당)	6-39
셀 텍스트 범위 구성요소 편집(Financial Management 사용자만 해당)	6-39
셀 텍스트 범위 구성요소 삭제(Financial Management 사용자만 해당)	6-39
셀 텍스트 범위 구성요소 복사 및 붙여넣기(Financial Management 사용자만 해당)	6-40
고정 루프 구성요소	6-40
고정 루프 구성요소 정보	6-40
고정 루프 구성요소 생성	6-41
고정 루프 구성요소 열기	6-42
고정 루프 구성요소 편집	6-42
고정 루프 구성요소 삭제	6-43
고정 루프 구성요소 복사 및 붙여넣기	6-44
메타데이터 루프 구성요소	6-44
메타데이터 루프 구성요소 정보	6-45
메타데이터 루프 구성요소 생성	6-45
메타데이터 루프 구성요소 열기	6-46
메타데이터 루프 구성요소 삭제	6-46
메타데이터 루프 구성요소 복사 및 붙여넣기	6-47
DTP 지정 구성요소	6-47
DTP 지정 구성요소 정보	6-48
DTP 지정 구성요소 생성	6-48
DTP 지정 구성요소 열기	6-49
DTP 지정 구성요소 편집	6-49
DTP 지정 구성요소 삭제	6-49
DTP 지정 구성요소 복사 및 붙여넣기	6-50
DTP 지정 구성요소에서 디자인 타임 프롬프트 함수 사용	6-50
디자인 타임 프롬프트 함수 정보	6-51
@Compare	6-51
@Compliment	6-52

@Concat	6-52
@밀집Member	6-53
@DimMember	6-54
@DimName	6-54
@DimType	6-55
@EndsWith	6-55
@Intersect	6-56
@Member	6-57
@Notin	6-58
@Plandim	6-59
@Quote	6-59
@RemoveQuote	6-60
@ReplaceAll	6-60
@ReplaceFirst	6-61
@SmartListFromIndex	6-62
@SmartListFromValue	6-63
@SparseMember	6-64
@StartsWith	6-65
@SubString	6-66
@ToLowerCase	6-66
@ToUpperCase	6-67
@Trim	6-68
@Union	6-68
스크립트 및 공식 구성요소 공유	6-69
스크립트 및 공식 구성요소 공유 정보	6-69
공식 및 스크립트 구성요소를 공유에서 공유 안 됨으로 변경	6-70
공식 및 스크립트 구성요소를 공유 안 됨에서 공유로 변경	6-70
구성요소 복사	6-71
비즈니스 규칙 구성요소의 1차 하위 구성요소 복사 및 붙여넣기	6-71
비즈니스 규칙 공식 또는 스크립트 구성요소에 대한 참조 복사 및 붙여넣기	6-72
구성요소 저장	6-73
구성요소 저장	6-73
공식 및 스크립트 구성요소를 다른 이름으로 저장	6-73
공식 및 스크립트 구성요소 새로고치기	6-74
공식 및 스크립트 구성요소의 사용 표시	6-75
순서도의 구성요소 작업	6-75
순서도의 구성요소 작업 정보	6-75
순서도에서 구성요소 확장 및 축소	6-75
순서도에서 구성요소 제거	6-76
순서도에서 구성요소 복사하여 붙여넣기	6-76
순서도에서 구성요소에 대한 참조 복사하여 붙여넣기	6-77



## 7 집계 저장영역 구성요소를 사용하여 비즈니스 규칙 디자인

집계 저장영역 구성요소를 사용하여 비즈니스 규칙 디자인 정보	7-1
POV 구성요소 작업	7-2
POV 구성요소 생성	7-2
POV 구성요소 편집	7-7
할당 구성요소 작업	7-7
할당 구성요소 생성	7-8
할당 구성요소 편집	7-11
POV 또는 할당 구성요소 열기	7-12
POV 또는 할당 구성요소 삭제	7-12
POV 또는 할당 구성요소 복사 및 붙여넣기	7-12
POV 또는 할당 구성요소 저장	7-13
집계 저장영역 공식 구성요소 작업	7-13
집계 저장영역 공식 구성요소 생성	7-13
집계 저장영역 공식 구성요소 열기	7-15
집계 저장영역 공식 구성요소 편집	7-16
집계 저장영역 공식 구성요소 삭제	7-16
집계 저장영역 공식 구성요소 복사 및 붙여넣기	7-17
다른 애플리케이션이나 데이터베이스에 집계 저장영역 공식 구성요소 복사	7-17
집계 저장영역 공식 구성요소의 사용 표시	7-18

## 8 멤버 선택, 변수, 함수, 스마트 목록 및 Planning 공식 표현식을 사용하여 구성요소 디자인

멤버 선택, 변수, 함수 및 스마트 목록 정보	8-1
구성요소에 멤버 및 함수 추가	8-2
구성요소에 멤버 및 함수 추가 정보	8-2
하나 이상의 차원에 있는 하나의 멤버 또는 함수를 구성요소에 추가	8-3
하나 이상의 차원에 있는 여러 멤버 또는 함수를 구성요소에 추가	8-11
구성요소에서 멤버 및 함수 제거	8-18
멤버 선택기 내에서 멤버 검색	8-18
멤버 선택기에서 이름, 별칭 또는 등록정보로 멤버 검색	8-19
변수 작업	8-20
변수 정보	8-20
변수 생성	8-21
숫자 변수에 대한 변수 값 입력	8-23
문자열 변수에 대한 변수 값 입력	8-24
배열 변수 값 입력	8-25

멤버 범위 변수 값 입력	8-25
횡차원 변수에 대한 변수 값 입력	8-27
차원 변수에 대한 변수 값 입력	8-27
멤버 또는 멤버 변수에 대한 변수 값 입력	8-28
퍼센트 변수에 대한 변수 값 입력	8-29
정수 변수에 대한 값 입력	8-29
문자열(숫자) 변수에 대한 변수 값 입력	8-30
날짜(숫자) 변수에 대한 변수 값 입력	8-30
데이터 범위 변수 값 입력(Financial Management 사용자만 해당)	8-31
부울 변수 값 입력(Financial Management 사용자만 해당)	8-31
로그 정보 변수 값 입력(Financial Management 사용자만 해당)	8-31
런타임 프롬프트 변수 입력	8-32
변수 선택	8-33
변수 편집	8-34
변수 삭제	8-34
변수 새로고침	8-35
변수 복사	8-35
변수 디자이너에서 텍스트 찾기 및 바꾸기	8-36
변수 사용 표시	8-37
사전 정의된 시스템 변수를 Financial Management 애플리케이션으로 로드(Financial Management 사용자만 해당)	8-38
함수 작업	8-38
Calculation Manager에서 지원되는 Essbase 함수	8-40
사용자정의 함수 작업	8-41
사용자정의 함수 정보	8-42
연도 매개변수가 포함된 사용자정의 함수 사용	8-42
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수	8-42
비트 단위 함수	8-42
@CalcMgrBitAnd	8-43
@CalcMgrBitOR	8-43
@CalcMgrBitExOR	8-43
@CalcMgrBitExBoolOR	8-43
@CalcMgrBitCompliment	8-44
@CalcMgrBitShiftLeft	8-44
@CalcMgrBitShiftRight	8-44
@CalcMgrBitUnsignedShiftRight	8-44
카운터 함수	8-45
@CalcMgrCounterAddNumber	8-45
@CalcMgrCounterAddText	8-45
@CalcMgrCounterClear	8-46
@CalcMgrCounterClearAll	8-46

@CalcMgrCounterClearKey	8-46
@CalcMgrCounterDecrement	8-46
@CalcMgrCounterDecrementKey	8-46
@CalcMgrCounterGetKeyNumber	8-47
@CalcMgrCounterGetKeyText	8-47
@CalcMgrCounterGetNumber	8-47
@CalcMgrCounterGetText	8-47
@CalcMgrCounterIncrement	8-48
@CalcMgrCounterIncrementKey	8-48
@CalcMgrCounterUpdate	8-48
@CalcMgrCounterUpdateNumber	8-48
@CalcMgrCounterUpdateNumberText	8-48
@CalcMgrCounterUpdateText	8-49
날짜/시간 함수	8-49
@CalcMgrAddDate	8-51
@CalcMgrAddDatePart	8-51
@CalcMgrAddDays	8-51
@CalcMgrAddMonths	8-51
@CalcMgrAddWeeks	8-52
@CalcMgrAddYears	8-52
@CalcMgrDateDiff	8-52
@CalcMgrDateToExcel	8-52
@CalcMgrDatesToExcel	8-53
@CalcMgrDateTimeToExcel	8-53
@CalcMgrDateTimesToExcel	8-53
@CalcMgrDateToString	8-53
@CalcMgrDaysBetween	8-54
@CalcMgrDaysDiff	8-54
@CalcMgrDiffDate	8-54
@CalcMgrExcelADD	8-55
@CalcMgrExcelDATE	8-55
@CalcMgrExcelDATEDIF	8-55
@CalcMgrExcelDAYOFYEAR	8-56
@CalcMgrExcelDAYS360	8-56
@CalcMgrExcelDAYSINMONTH	8-56
@CalcMgrExcelEOMONTH	8-57
@CalcMgrExcelHOUR	8-57
@CalcMgrExcelMINUTE	8-57
@CalcMgrExcelMONTH	8-58
@CalcMgrExcelNETWORKDAYS	8-58
@CalcMgrExcelSECOND	8-59

@CalcMgrExcelToDate	8-59
@CalcMgrExcelToDateTime	8-59
@CalcMgrExcelWEEKNUM	8-59
@CalcMgrExcelWEEKDAY	8-60
@CalcMgrExcelWORKDAY	8-60
@CalcMgrExcelYEAR	8-61
@CalcMgrExcelYEARFRAC	8-61
@CalcMgrGetCurrentDate	8-62
@CalcMgrGetCurrentDateTZ	8-62
@CalcMgrGetCurrentDateTime	8-62
@CalcMgrGetCurrentDateTimeTZ	8-62
@CalcMgrGetCustomDate	8-63
@CalcMgrGetCustomDateTime	8-63
@CalcMgrGetDatePart	8-63
@CalcMgrGetDateTimePart	8-63
@CalcMgrGetDay	8-64
@CalcMgrGetDayOfYear	8-64
@CalcMgrGetFormattedDate	8-64
@CalcMgrGetMaxDaysInMonth	8-64
@CalcMgrGetMonth	8-64
@CalcMgrGetStringFormattedDateTime	8-65
@CalcMgrGetWeekOfMonth	8-65
@CalcMgrGetWeekOfYear	8-65
@CalcMgrGetYear	8-65
@CalcMgrIsLeapYear	8-66
@CalcMgrMonthsBetween	8-66
@CalcMgrMonthsDiff	8-66
@CalcMgrRollDate	8-67
@CalcMgrRollDay	8-67
@CalcMgrRollMonth	8-67
@CalcMgrRollYear	8-67
@CalcMgrWeeksBetween	8-68
@CalcMgrWeeksDiff	8-68
@CalcMgrYearsBetween	8-68
@CalcMgrYearsDiff	8-69
재무 함수	8-69
@CalcMgrExcelACCRINT	8-70
@CalcMgrExcelACCRINTM	8-71
@CalcMgrExcelAMORDEGRC	8-71
@CalcMgrExcelAMORLINC	8-71
@CalcMgrExcelCOUPDAYBS	8-71

@CalcMgrExcelCOUPDAYS	8-72
@CalcMgrExcelCOUPDAYSNC	8-72
@CalcMgrExcelCOUPNCD	8-72
@CalcMgrExcelCOUPNUM	8-73
@CalcMgrExcelCOUPPCD	8-73
@CalcMgrExcelCUMIPMT	8-74
@CalcMgrExcelCUMPRINC	8-74
@CalcMgrExcelDB	8-74
@CalcMgrExcelDDB	8-74
@CalcMgrExcelDISC	8-74
@CalcMgrExcelDOLLARDE	8-75
@CalcMgrExcelDOLLARFR	8-75
@CalcMgrExcelDURATION	8-75
@CalcMgrExcelEFFECT	8-76
@CalcMgrExcelFV	8-76
@CalcMgrExcelFVSCHEDULE	8-76
@CalcMgrExcelMDURATION	8-76
@CalcMgrExcelINTRATE	8-77
@CalcMgrExcelPMT	8-77
@CalcMgrExcelIRR	8-77
@CalcMgrExcelSPMT	8-78
@CalcMgrExcelMIRR	8-78
@CalcMgrExcelNPER	8-78
@CalcMgrExcelNPV	8-78
@CalcMgrExcelPPMT	8-79
@CalcMgrExcelPRICE	8-79
@CalcMgrExcelPRICEDISC	8-79
@CalcMgrExcelPRICEMAT	8-80
@CalcMgrExcelPV	8-80
@CalcMgrExcelRATE	8-80
@CalcMgrExcelRECEIVED	8-80
@CalcMgrExcelSLN	8-81
@CalcMgrExcelSYD	8-81
@CalcMgrExcelTBILLEQ	8-81
@CalcMgrExcelTBILLPRICE	8-82
@CalcMgrExcelTBILLYIELD	8-82
@CalcMgrExcelXIRR	8-82
@CalcMgrExcelXNPV	8-83
@CalcMgrExcelYIELD	8-83
@CalcMgrExcelYIELDDISC	8-83
@CalcMgrExcelYIELDMAT	8-84

로그 함수	8-84
@CalcMgrLog	8-84
@CalcMgrLogText	8-85
수리 함수	8-86
@CalcMgrExcelCEILING	8-86
@CalcMgrExcelCOMBIN	8-87
@CalcMgrExcelEVEN	8-87
@CalcMgrExcelFACT	8-87
@CalcMgrExcelFLOOR	8-87
@CalcMgrExcelGCD	8-87
@CalcMgrExcelLCM	8-88
@CalcMgrExcelMROUND	8-88
@CalcMgrExcelMULTINOMIAL	8-88
@CalcMgrExcelODD	8-88
@CalcMgrExcelPOWER	8-89
@CalcMgrExcelPRODUCT	8-89
@CalcMgrExcelROUNDDOWN	8-89
@CalcMgrExcelROUNDUP	8-89
@CalcMgrExcelSQRT	8-89
@CalcMgrExcelSQRTPI	8-90
@CalcMgrExcelSUMPRODUCT	8-90
@CalcMgrExcelSUMSQ	8-90
MaxL 함수	8-90
@CalcMgrExecuteEncryptMaxLFile	8-91
@CalcMgrExecuteMaxLScript	8-91
MaxL 암호화 사용	8-91
@CalcMgrExecuteEncryptMaxLFile을 사용하여 MaxL 스크립트 파일 실행	8-91
@CalcMgrExecuteMaxLScript를 사용하여 MaxL 스크립트 실행	8-92
MDX 함수	8-96
@CalcMgrMDXDataCopy	8-96
@CalcMgrMDXDataCopyRTP	8-98
@CalcMgrMDXDataCopyServer	8-99
@CalcMgrMDXExport	8-100
@CalcMgrMDXExportRTP	8-101
통계 함수	8-102
@CalcMgrExcelAVEDEV	8-102
@CalcMgrExcelDEVSQ	8-102
@CalcMgrExcelLARGE	8-102
@CalcMgrExcelMEDIAN	8-103
@CalcMgrExcelSMALL	8-103
@CalcMgrExcelSTDEV	8-103

@CalcMgrExcelVAR	8-103
@CalcMgrExcelVARP	8-104
문자열 함수	8-104
@CalcMgrCompare	8-105
@CalcMgrConcat	8-105
@CalcMgrDecimalFormat	8-105
@CalcMgrDoubleFromString	8-106
@CalcMgrDoubleToString	8-106
@CalcMgrDQuote	8-106
@CalcMgrEndsWith	8-106
@CalcMgrFindFirst	8-106
@CalcMgrFindLast	8-107
@CalcMgrFormatDouble	8-107
@CalcMgrGetListCount	8-107
@CalcMgrGetListItem	8-107
@CalcMgrIndexOf	8-108
@CalcMgrIntegerToString	8-108
@CalcMgrLastIndexOf	8-108
@CalcMgrLowercase	8-108
@CalcMgrMatches	8-109
@CalcMgrMessageFormat	8-109
@CalcMgrPadText	8-109
@CalcMgrUppercase	8-109
@CalcMgrRemoveQuotes	8-110
@CalcMgrRemoveDQuotes	8-110
@CalcMgrRemoveSQuotes	8-110
@CalcMgrReplaceAll	8-110
@CalcMgrReplaceFirst	8-110
@CalcMgrSortAndReturn	8-111
@CalcMgrSortList	8-111
@CalcMgrSortValues	8-111
@CalcMgrSplit	8-111
@CalcMgrSQuote	8-112
@CalcMgrStartsWith	8-112
@CalcMgrStringsToString	8-112
@CalcMgrSubstring	8-112
@CalcMgrTextLength	8-113
@CalcMgrTrim	8-113
구성요소에 함수 삽입	8-113
스마트 목록 작업	8-115
스마트 목록 삽입	8-115

Planning 공식 표현식 작업	8-115
스마트 목록	8-116
차원(둘 이상)	8-116
Planning 사용자 변수	8-117
기간	8-117
Period(periodName)	8-117
NumberOfPeriodsinYear 및 NumberOfYears	8-118
시나리오	8-119
상호 참조	8-123
CrossRef(accountName)	8-123
CrossRef(accountName, prefix)	8-123
CrossRef(accountName, prefix, true)	8-124
Workforce 큐브 연 누계	8-124
CYTD(memberName)	8-124
CYTD(memberName, calTpIndexName, fiscalTpIndexName)	8-125
문자열의 ID 가져오기	8-125

## 9 검증 및 배포

---

시스템 뷰에서 비즈니스 규칙, 비즈니스 규칙 세트, 공식 및 스크립트 구성요소 검증	9-1
규칙 디자이너에서 비즈니스 규칙 검증	9-2
비즈니스 규칙 및 비즈니스 규칙 세트 배포	9-3
비즈니스 규칙 및 비즈니스 규칙 세트 배포 정보	9-3
비즈니스 규칙 및 비즈니스 규칙 세트를 배포 가능 및 배포 불가능하게 만들기	9-4
배포 뷰에서 비즈니스 규칙 및 비즈니스 규칙 세트 배포	9-4
규칙 또는 규칙 세트 디자이너에서 비즈니스 규칙 또는 비즈니스 규칙 세트 배포	9-6
바로가기 있는 비즈니스 규칙 배포	9-6
Planning에 표시되는 배포된 비즈니스 규칙 지정	9-7

## 10 비즈니스 규칙 실행

---

비즈니스 규칙 실행 정보	10-1
시스템 뷰 및 규칙 디자이너에서 Essbase Business Rules 실행	10-1
명령행 실행 프로그램으로 Essbase Business Rules 실행	10-2
명령행 실행 프로그램 예제 및 인수	10-2
Windows에서 Bat 파일을 사용하여 명령행 실행 프로그램 실행	10-3
규칙 디자이너에서 Planning 비즈니스 규칙 실행 및 로그 보기	10-4



## 11 비즈니스 규칙, 비즈니스 규칙 세트, 템플릿, 공식 및 스크립트 구성요소 익스포트 및 임포트

---

익스포트 및 임포트 정보	11-1
비즈니스 규칙, 비즈니스 규칙 세트, 템플릿, 공식 및 스크립트 구성요소 익스포트	11-2
애플리케이션 익스포트	11-2
로그 메시지를 파일로 익스포트	11-3
규칙, 규칙 세트, 템플릿, 공식 및 스크립트 임포트	11-3
Essbase 계산 스크립트를 그래픽 비즈니스 규칙으로 임포트	11-5

## 12 Planning 애플리케이션에 대한 Essbase 서버, 애플리케이션 및 데이터베이스 관리

---

데이터베이스 등록정보 작업	12-2
데이터베이스 등록정보 보기 및 편집	12-2
일반 데이터베이스 등록정보	12-3
차원 등록정보	12-4
통계 등록정보	12-4
집계 저장영역 애플리케이션에 대한 통계	12-4
블록 저장영역 애플리케이션에 대한 통계	12-6
트랜잭션 등록정보	12-7
수정 등록정보	12-8
통화 등록정보	12-8
캐시 등록정보	12-9
MaxL 스크립트 작성 및 실행	12-9
데이터베이스 객체에서 잠금 제거	12-10
애플리케이션 시작 및 중지	12-10
데이터베이스 시작 및 중지	12-11
애플리케이션 로그 파일 삭제	12-12
데이터베이스 재구조화	12-12
아웃라인 확인	12-13
데이터베이스 설정 최적화에 유용한 시뮬레이션 생성	12-14
데이터베이스에서 데이터 지우기	12-16
집계 저장영역 애플리케이션에서 데이터 지우기	12-16
블록 저장영역 애플리케이션에서 데이터 블록 지우기	12-17
블록 저장영역 애플리케이션에 대한 위치 별칭 작업	12-17
블록 저장영역 애플리케이션에 대한 위치 별칭 작업	12-17
위치 별칭 목록 표시	12-18
위치 별칭 익스포트	12-18
위치 별칭 생성	12-18
위치 별칭 삭제	12-18

집계 저장영역 데이터베이스에서 쿼리 추적 사용	12-19
집계 저장영역 데이터베이스 아웃라인 압축	12-19
레벨 0 데이터 임포트 및 익스포트	12-20
ASO 큐브에서 레벨 0 데이터 임포트	12-20
ASO 큐브에서 레벨 0 데이터 익스포트	12-21
BSO 큐브에서 레벨 0 데이터 임포트	12-21
BSO 큐브에서 레벨 0 데이터 익스포트	12-22
증분 데이터 슬라이스 병합	12-22
데이터 집계	12-23
집계 프로세스 실행	12-23
증분 데이터 슬라이스 병합 및 0 값 셀 제거	12-24
쿼리 추적 사용	12-24
쿼리 생성 작업 수행	12-25
쿼리 추적을 사용하여 집계 실행	12-25
요청 관리	12-25
Planning 드릴스루 정의 추가	12-27

## A 관리자 기능 사용

---

Calculation Manager로 전환	A-1
-------------------------	-----

# 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

## Calculation Manager 개요

### 참조:

- [Calculation Manager 정보](#)  
Oracle Hyperion Calculation Manager를 사용하여 비즈니스 문제를 해결할 계산을 생성, 검증, 배포 및 실행합니다.
- [Calculation Manager를 사용하는 서비스](#)  
여러 Enterprise Performance Management Cloud 서비스에서 Oracle Hyperion Calculation Manager를 사용합니다.
- [역할이 Calculation Manager에서 수행 가능한 작업에 영향을 주는 방식](#)  
Oracle Hyperion Shared Services에서 지정된 역할에 따라 작업할 수 있는 규칙, 규칙 세트, 구성요소 및 템플리트와 Oracle Hyperion Calculation Manager에서 수행할 수 있는 작업이 결정됩니다.
- [Calculation Manager 실행](#)  
Oracle Hyperion Calculation Manager를 실행하여 규칙, 규칙 세트, 구성요소 및 템플리트를 생성합니다.
- [Calculation Manager에서 객체 보기](#)  
뷰를 사용하면 Oracle Hyperion Calculation Manager 객체를 다양한 컨텍스트에서 볼 수 있습니다.
- [뷰의 열 사용자정의](#)  
각 뷰에 표시되는 열과 열 표시 순서를 사용자정의할 수 있습니다.
- [객체 필터링](#)  
Oracle Hyperion Calculation Manager의 시스템 뷰 또는 필터 뷰에서 객체를 필터링합니다.
- [쿼리를 사용하여 객체 필터링](#)  
필터 뷰에서 객체를 필터링한 후 쿼리를 사용하여 표시되는 객체를 더욱 구체적으로 지정할 수 있습니다.
- [규칙, 구성요소 또는 템플리트의 그래픽 그림 보기](#)  
규칙, 구성요소 및 템플리트는 규칙 디자이너 및 템플리트 디자이너 내 순서도에 그래픽으로 표시됩니다.

## Calculation Manager 정보

Oracle Hyperion Calculation Manager를 사용하여 비즈니스 문제를 해결할 계산을 생성, 검증, 배포 및 실행합니다.

계산에서 다음 유형의 객체를 생성할 수 있습니다.

- **규칙** - 구성요소, 템플리트 및 기타 규칙이 포함된 객체입니다.
- **규칙 세트** - 동시에 또는 순차적으로 계산할 수 있는 두 개 이상의 비즈니스 규칙이 포함된 객체입니다.
- **구성요소** - 공식, 스크립트, 조건, 멤버 및 데이터 범위, 고정 루프, 디자인 타임 프롬프트가 포함된 객체입니다. 구성요소는 배포할 수 없습니다.
- **템플리트** - 비즈니스 규칙에서 계산 또는 계산 세트를 수행하는 데 사용할 수 있는 객체입니다.

## Calculation Manager를 사용하는 서비스

여러 Enterprise Performance Management Cloud 서비스에서 Oracle Hyperion Calculation Manager를 사용합니다.

- Enterprise Planning and Budgeting Cloud
- Planning and Budgeting Cloud
- Financial Consolidation and Close Cloud

## 역할이 Calculation Manager에서 수행할 수 있는 작업에 영향을 주는 방식

Oracle Hyperion Shared Services에서 지정된 역할에 따라 작업할 수 있는 규칙, 규칙 세트, 구성요소 및 템플릿과 Oracle Hyperion Calculation Manager에서 수행할 수 있는 작업이 결정됩니다.

표 1-1 Calculation Manager의 역할 및 해당 영향

애플리케이션 상태	역할	Calculation Manager에서 수행할 수 있는 작업
배포되지 않음	<ul style="list-style-type: none"> <li>• 글로벌 Calculation Manager 관리자</li> <li>• Oracle Hyperion Financial Management Calculation Manager 관리자</li> <li>• Oracle Hyperion Planning Calculation Manager 관리자</li> <li>• Oracle Essbase 관리자</li> </ul> <p>Essbase 관리자 역할은 Essbase에서 상속됩니다. Essbase Calculation Manager 관리자 역할은 없습니다.</p>	<p>규칙 및 기타 객체를 편집, 삭제, 배포 및 실행합니다.</p> <p>애플리케이션, 규칙 및 기타 객체를 배포한 후 다음을 수행할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• Financial Management에서 Financial Management 비즈니스 규칙 실행</li> <li>• Planning 또는 Calculation Manager에서 Planning 비즈니스 규칙 실행</li> <li>• Oracle Essbase Administration Services 또는 Calculation Manager에서 Essbase 비즈니스 규칙 실행</li> <li>• Oracle Fusion General Ledger에서 General Ledger 비즈니스 규칙 실행</li> </ul>
배포됨	<p>배포된 애플리케이션에 대한 글로벌 역할:</p> <ul style="list-style-type: none"> <li>• Financial Management 관리자</li> <li>• Planning 관리자</li> <li>• Essbase 관리자</li> </ul>	<p>해당 애플리케이션에 속하는 비즈니스 규칙을 생성, 편집 및 삭제합니다. 애플리케이션을 배포할 수도 있습니다.</p>
배포됨	<p>애플리케이션별 역할:</p> <ul style="list-style-type: none"> <li>• Financial Management Calculation Designer</li> <li>• Planning 대화식 사용자</li> <li>• Essbase 계산 쓰기 권한</li> </ul>	<p>자신의 규칙을 작성, 편집 및 삭제하고 다른 사용자의 규칙을 봅니다.</p> <p><b>참고:</b> 다른 사용자가 규칙의 소유권을 사용자에게 지정한 경우 외에는 다른 사용자의 규칙을 편집할 수 없습니다.</p>
배포됨	<p>애플리케이션별 역할:</p> <ul style="list-style-type: none"> <li>• Financial Management 뷰어</li> <li>• Planning 뷰 사용자</li> </ul>	<p>규칙 보기</p>

**주:**

역할 및 보안에 대한 자세한 내용은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

## Calculation Manager 실행

Oracle Hyperion Calculation Manager를 실행하여 규칙, 규칙 세트, 구성요소 및 템플릿을 생성합니다.

달력 관리자를 실행하려면 다음을 수행합니다.

1. Oracle Hyperion Enterprise Performance Management Workspace를 엽니다.
2. **탐색, 관리, Calculation Manager** 순으로 선택합니다.

## Calculation Manager에서 객체 보기

뷰를 사용하면 Oracle Hyperion Calculation Manager 객체를 다양한 컨텍스트에서 볼 수 있습니다.

뷰를 사용 중인 경우 **뷰** 메뉴의 옵션을 사용하여 표시되는 열과 표시 순서를 사용자정의할 수 있습니다.

Calculation Manager를 실행하면 **시스템 뷰**가 자동으로 표시됩니다. 뷰를 전환하려면 **뷰 선택** 옆의 드롭다운에서 뷰를 선택합니다.

Calculation Manager에 포함되어 있는 뷰는 다음과 같습니다.

- **시스템 뷰** - Calculation Manager를 실행할 때 표시되는 기본 뷰입니다. 이 뷰에는 액세스할 수 있는 모든 애플리케이션과 객체가 나열됩니다.

사용자의 액세스 권한은 Oracle Hyperion Shared Services에서 사용자에게 지정한 역할에 따라 결정되며 애플리케이션 단위로 지정됩니다.

- **사용자정의 뷰** - 사용자에게 의미 있는 뷰를 생성하려면 폴더를 생성하고 해당 폴더에 객체를 추가합니다.

**사용자정의 뷰**에서 폴더를 생성하려면 다음을 수행합니다.

1. 오른쪽 창에서 애플리케이션을 마우스 오른쪽 버튼으로 누르고 **새로 작성, 폴더** 순으로 선택합니다.
  2. **새 폴더**에서 폴더 이름을 입력하고 **확인**을 누릅니다.
  3. **기존 객체** 창의 객체를 폴더에 끌어 놓아 폴더에 객체를 추가합니다.
- **배포 뷰** - 애플리케이션 유형 및 애플리케이션별로 배포 가능한 규칙 및 규칙 세트와 해당 배포, 검증 상태가 나열됩니다. 배포 가능으로 설정할 규칙 및 규칙 세트를 선택한 다음 애플리케이션에 객체를 배포할 수 있습니다.  
하나 이상의 규칙 및 규칙 세트를 애플리케이션에 배포하는 것은 부분 배포이며 모든 규칙 및 규칙 세트를 애플리케이션에 배포하는 것은 전체 배포입니다.
  - **필터 뷰** - 표시되는 객체를 필터링합니다.

**필터** 대화상자에서 필터 옵션을 정의한 다음 쿼리를 사용하여 필터를 더 구체적으로 지정할 수 있습니다.

## 뷰의 열 사용자정의

각 뷰에 표시되는 열과 열 표시 순서를 사용자정의할 수 있습니다.

뷰의 열을 사용자정의하려면 다음 중 하나를 수행합니다.

- **뷰, 열** 순으로 선택하고 표시할 열을 선택합니다. **뷰, 열 순서 재지정** 순으로 선택하고 열 순서를 선택하여 열 순서를 조정합니다.
- **뷰, 열, 열 관리** 순으로 선택합니다. **열 관리** 대화상자에서 표시할 열과 표시 순서를 선택합니다.

### 주:


Oracle Financials Cloud에는 열에 대한 새로운 "잠금 해제" 기능이 있습니다. 이 열은 기본적으로 표시되지 않습니다. 잠금 해제를 사용하면 관리자가 다른 관리자가 잠근 객체를 잠금 해제할 수 있습니다.

## 객체 필터링

Oracle Hyperion Calculation Manager의 시스템 뷰 또는 필터 뷰에서 객체를 필터링합니다.

객체를 애플리케이션 유형, 애플리케이션, 계산 유형, 계획 유형, 데이터베이스, 객체 유형 (비즈니스 규칙, 비즈니스 규칙 세트, 공식 및 스크립트 구성요소, 템플릿 등), 배포 또는 검증 상태 등에 따라 필터링할 수 있습니다.



객체를 필터링하려면 다음을 수행합니다.

1. 시스템 뷰 또는 필터 뷰에서  를 누릅니다.
2. 필터 대화상자에서 필요한 정보를 입력하고 **확인**을 누릅니다.

## 쿼리를 사용하여 객체 필터링

필터 뷰에서 객체를 필터링한 후 쿼리를 사용하여 표시되는 객체를 더욱 구체적으로 지정할 수 있습니다.

쿼리를 사용하여 객체를 필터링하려면 다음을 수행합니다.

1. 필터 뷰에서  를 누르고 **필터** 대화상자에 정보를 입력합니다.
2.  를 눌러 열 위에 텍스트 상자를 표시합니다.
3. 열 위의 텍스트 상자에 쿼리를 입력하여 필터를 더욱 구체적으로 지정합니다.

와일드카드나 부분 텍스트 문자열을 사용할 수 없습니다. 계획 유형을 검색하려면 첫 번째 단어의 시작 문자를 입력합니다. 예를 들어 계획 유형의 이름이 "Plan1"이고 쿼리에 "la"를 입력하면 "La"로 시작하는 객체만 표시됩니다. 이 예에서 원하는 결과를 얻으려면 "Pl"을 입력하여 "Pl"로 시작하는 모든 계획 유형을 표시합니다.





주:

변수 디자이너에서 쿼리를 사용하여 객체를 필터링할 수도 있습니다.

## 규칙, 구성요소 또는 템플리트의 그래픽 그림 보기

규칙, 구성요소 및 템플리트는 규칙 디자이너 및 템플리트 디자이너 내 순서도에 그래픽으로 표시됩니다.

규칙 또는 템플리트를 열면 순서도에서 해당 구성요소(예: 공식, 스크립트, 조건, 범위 및 루프)를 선택하여 세부정보를 볼 수 있습니다. 또한 순서도의 크기를 늘리거나 줄여 구성요소 세부정보를 표시하거나 숨길 수 있습니다.

순서도에서 구성요소를 선택하면 해당 등록정보, 사용 및 기타 정보가 순서도 아래의 탭에 표시됩니다. 구성요소 간을 이동하면 순서도 아래의 탭이 변경됩니다.

예를 들어 공식 구성요소 및 스크립트 구성요소가 포함된 비즈니스 규칙을 열고 순서도에서 공식 구성요소를 선택하면 공식 등록정보(이름, 설명, 애플리케이션 및 애플리케이션 유형)가 순서도 아래의 탭에 표시됩니다. 그런 다음 순서도에서 스크립트 구성요소를 선택하면 스크립트 구성요소의 텍스트, 등록정보 및 사용이 순서도 아래의 탭에 표시됩니다.

# 2

## 비즈니스 규칙 디자인

### 참조:

- [비즈니스 규칙 정보](#)  
Oracle Hyperion Calculation Manager를 사용하면 복잡한 다차원 비즈니스 규칙을 생성, 검증, 배포 및 관리할 수 있습니다.
- [비즈니스 규칙 생성](#)  
비즈니스 규칙은 구성요소로 그룹화된 계산으로 이루어진 Oracle Hyperion Calculation Manager 객체입니다.
- [비즈니스 규칙 열기](#)  
Oracle Hyperion Calculation Manager를 열 때 기본적으로 표시되는 시스템 뷰에서 비즈니스 규칙을 열 수 있습니다.
- [비즈니스 규칙 편집](#)  
비즈니스 규칙에서 구성요소(공식, 스크립트, 조건, 범위 및 루프 포함)를 추가, 제거 또는 변경하여 비즈니스 규칙의 구조를 편집할 수 있습니다.
- [스크립트 모드로 비즈니스 규칙 편집](#)  
비즈니스 규칙의 스크립트를 편집할 수 있습니다.
- [비즈니스 규칙 저장](#)  
비즈니스 규칙을 이 규칙이 생성된 애플리케이션 및 애플리케이션 유형에 저장합니다.
- [비즈니스 규칙 실행](#)  
비즈니스 규칙을 실행하기 전에 먼저 열어야 합니다.
- [비즈니스 규칙 복사](#)  
비즈니스 규칙을 다른 애플리케이션 및 계획 유형으로 복사할 수 있습니다.
- [비즈니스 규칙에서 검색](#)  
비즈니스 규칙 스크립트에서 텍스트 문자열을 검색합니다. 그래픽 비즈니스 규칙에서 텍스트를 검색하고 바꿉니다.
- [비즈니스 규칙 인쇄](#)  
비즈니스 규칙의 등록정보, 순서도 및 해당 구성요소의 세부정보를 인쇄할 수 있습니다.
- [비즈니스 규칙 삭제](#)  
다른 규칙이나 규칙 세트에서 사용되지 않는 비즈니스 규칙만 삭제할 수 있습니다.
- [비즈니스 규칙 구성요소의 공통 차원 정의](#)  
비즈니스 규칙을 열고 각 차원에 공통되는 멤버, 변수 및 함수를 선택하여 공통 차원을 정의합니다.
- [비즈니스 규칙 분석 및 디버깅](#)  
규칙을 실행하고 통계 정보를 캡처하여 비즈니스 규칙을 분석합니다. 규칙을 실행하고 스크립트를 검사하여 비즈니스 규칙을 디버그합니다.
- [사전 정의된 Workforce Planning, Capital Asset Planning 및 Human Capital Management 비즈니스 규칙 로드](#)
- [비즈니스 규칙에 대한 바로가기 생성](#)

- [비즈니스 규칙 또는 비즈니스 규칙 세트 새로고침](#)  
시스템 뷰, 사용자정의 뷰 및 배포 뷰에서 애플리케이션 목록의 아무 레벨이나 새로고칠 수 있습니다.
- [비즈니스 규칙 또는 규칙 세트에 대한 사용 표시](#)  
비즈니스 규칙 또는 비즈니스 규칙 세트를 사용하고 있는 규칙, 템플릿 및 규칙 세트를 표시합니다.
- [객체의 소유자 변경](#)

## 비즈니스 규칙 정보

Oracle Hyperion Calculation Manager를 사용하면 복잡한 다차원 비즈니스 규칙을 생성, 검증, 배포 및 관리할 수 있습니다.

일반적으로 다음 작업을 위해 비즈니스 규칙을 생성합니다.

- 엔티티 간에 비용 할당
- 수익 모델링
- 비용 모델링
- 대차대조표 준비
- 현금 플로우 계산
- 외화 환산 조정 계산
- 그룹 및 소액 주주 지분 계산
- 이연 법인세 계산

비즈니스 규칙을 생성하려면 데이터베이스 아웃라인 및 작업 중인 애플리케이션에 대해 잘 알고 있어야 합니다. 이러한 정보가 있으면 비즈니스 규칙을 보다 효율적으로 생성할 수 있습니다. 또한 데이터에 대해 다음 내용을 알고 있어야 합니다.

- 데이터가 저장되고 집계되는 방법
- 데이터가 데이터베이스에 로드된 레벨
- 계산 순서
- 계산 수행에서의 주요 가정

공식, 스크립트, 루프, 데이터 및 멤버 범위, 템플릿, 런타임 프롬프트 변수를 비롯한 여러 변수 등의 구성요소를 사용하여 비즈니스 규칙을 생성할 수 있습니다. [구성요소를 사용하여 비즈니스 규칙 및 템플릿 디자인](#)을 참조하십시오.

비즈니스 규칙을 생성할 때 작업 중인 구성요소, 템플릿 및 변수를 열어 둘 수 있습니다. Calculation Manager에서는 이러한 객체를 탭 인터페이스에 표시하므로 규칙을 생성하는 동안 탭 간에 쉽게 이동할 수 있습니다. 탭은 10개까지 열어 둘 수 있습니다. 최적의 성능을 위해서는 동시에 여는 객체 수가 10개를 초과하지 않는 것이 좋습니다.

Oracle Hyperion Planning에서 비즈니스 규칙을 실행하려면 관리자가 규칙에 실행 권한을 부여해야 합니다.

 주:

또한 동시에 또는 순서대로 실행할 수 있는 두 개 이상의 관련 규칙(또는 규칙 세트)으로 이루어진 비즈니스 규칙 세트를 생성할 수도 있습니다. [비즈니스 규칙 세트 디자인](#)을 참조하십시오.

 주:

Oracle Hyperion Financial Management에서 비즈니스 규칙을 실행할 때는 실행 권한이 필요하지 않습니다.

## 비즈니스 규칙 생성

비즈니스 규칙은 구성요소로 그룹화된 계산으로 이루어진 Oracle Hyperion Calculation Manager 객체입니다.


규칙에는 하나 이상의 구성요소, 템플릿 또는 규칙이 포함될 수 있습니다.

액세스할 수 있는 애플리케이션에 대한 비즈니스 규칙을 생성할 수 있습니다. 규칙 생성 가능 여부는 지정된 역할에 따라 결정됩니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

규칙은 순서도에 그래픽으로 표현되며, 순서도에 구성요소를 끌어 놓아 규칙을 디자인할 수 있습니다.

비즈니스 규칙을 생성하려면 다음을 수행합니다.

1. 다음 중 하나를 수행합니다.

- 시스템 뷰, 사용자정의 뷰, 배포 뷰 또는 필터 뷰에서  을 누르고 새 객체 대화상자에 정보를 입력합니다. 규칙을 객체 유형으로 선택했는지 확인합니다.
- 시스템 뷰에서 규칙을 마우스 오른쪽 버튼으로 누르고 새로 작성을 선택한 다음 새 규칙 대화상자에 정보를 입력합니다.

 주:

규칙 이름에는 'r', 'n', 't', 'f', 'b', '<', '>', '(', ')', '"', '\', '{', '}', '[', ']', '\*', '?' 문자를 사용할 수 없습니다.

2. 규칙 디자이너에서 새 객체 및 기존 객체 아래의 객체를 선택하고 시작 및 종료 사이의 순서도로 끌어 놓습니다.

다음 사항에 유의하십시오.

- 기존 공식 또는 스크립트 구성요소를 순서도에 끌어 놓으면 기본적으로 공식 또는 스크립트가 공유 객체가 됩니다. 구성요소를 공유하지 않으려면 공식 또는 스크립트 구성요소의 등록정보에서 공유 확인란을 선택 취소합니다. [스크립트 및 공식 구성요소 공유](#)를 참조하십시오.

- 스크립트 구성요소를 그래픽 형식으로 보려면 순서도에서 스크립트 구성요소를 마우스 오른쪽 버튼으로 누르고 **그래픽으로 변환**을 선택합니다.  
스크립트가 적합한 경우에만 스크립트 구성요소가 그래픽 형식으로 변환됩니다. 그래픽 형식으로 변환을 실행취소하려면 스크립트 구성요소를 마우스 오른쪽 버튼으로 누르고 **실행취소**를 선택합니다.
  - 규칙과 별도로 공식 및 스크립트와 같은 객체를 생성하고 나중에 규칙에 추가할 수 있습니다.
  - 비그래픽 형식(스크립트 형식)의 비즈니스 규칙으로 작업하려면 **디자이너 옆에 있는 드롭다운을 누르고 스크립트 편집**을 선택합니다([스크립트 모드로 비즈니스 규칙 편집 참조](#)).
3. 규칙의 등록정보를 입력합니다.

규칙에 구성요소를 추가하고 순서도에서 구성요소 간에 이동하게 되면 등록정보가 변경됩니다. 특정 구성요소에 대한 등록정보를 입력하려면 순서도에서 구성요소를 선택합니다.

순서도에서 **시작** 또는 **종료**를 선택하는 경우 표시되는 등록정보는 다음과 같습니다.

- **일반** - 이름, 설명(description) 및 설명(comment)
- **위치** - 애플리케이션 및 계획 유형
- **옵션** - 현재 애플리케이션에 대한 옵션

**표 2-1 Planning 옵션**

등록정보	설명
동적 멤버 생성	런타임 프롬프트를 사용하도록 설정된 해당 멤버 유형의 변수에서 기본 동적 상위 멤버를 지정하는 경우 멤버를 생성합니다. 비즈니스 규칙에서 동적 멤버를 생성하고 기본 동적 상위 멤버를 선택하는 경우 Oracle Hyperion Planning에서 규칙이 실행되기 <i>전에</i> 자동으로 해당 상위 멤버의 1차 하위 멤버가 생성됩니다. <b>참고:</b> [동적 멤버 생성]을 선택할 경우 비즈니스 규칙이 실행되지 못하면 새로 생성된 멤버가 삭제됩니다. <b>참고:</b> 이 등록정보는 Financial Consolidation and Close 애플리케이션에 사용할 수 없습니다.
동적 멤버 삭제	런타임 프롬프트를 사용하도록 설정된 해당 멤버 유형의 변수에서 기본 동적 상위 멤버를 지정하는 경우 멤버를 삭제합니다. 비즈니스 규칙에서 동적 멤버를 삭제하고 기본 동적 상위 멤버를 선택하는 경우 Planning에서 규칙이 실행된 <i>후에</i> 자동으로 해당 상위 멤버의 1차 하위 멤버가 삭제됩니다. <b>참고:</b> 이 등록정보는 <b>Financial Consolidation and Close</b> 애플리케이션에 사용할 수 없습니다.
통지 사용	오류 여부와 관계 없이 규칙이 실행되면 로그인된 사용자에게 전자메일 통지를 보내는 비즈니스 규칙을 사용합니다.

표 2-2 Financial Management 옵션


등록정보	설명
로깅 사용	규칙이 실행되는 경우 로그 파일에 로그 텍스트가 포함됩니다. 규칙, 규칙 세트 및 구성요소에 대해 로깅을 사용할 수 있습니다. 비즈니스 규칙의 공식 명령문에 포함할 로그 텍스트를 지정합니다. 규칙 세트에 대해서는 로깅을 사용하지 않지만 규칙 세트에 속한 규칙 및 구성요소에 대해 로깅을 사용하는 경우 규칙 세트에 대한 로그 설정이 규칙 및 구성요소에 대한 로그 설정을 재정의하므로 로그 파일이 생성되지 않습니다.
타이머 사용	규칙이 실행되는 경우 규칙을 처리하는 데 걸린 시간을 로그 파일에 기록합니다. 규칙, 규칙 세트 및 구성요소에 대해 타이머를 사용할 수 있습니다. 타이머가 활성화된 모든 객체에 대한 처리 시간이 로그 파일에 포함됩니다. 예를 들어 타이머가 사용되는 규칙 세트가 있고 이 규칙 세트에 타이머가 사용되는 규칙 세 개가 있는 경우 규칙 세트 및 규칙 세트의 각 규칙을 처리하는 데 걸린 시간이 기록됩니다.
사용 안함	스크립트가 생성되는 동안 일시적으로 비즈니스 규칙을 사용 안함으로 설정합니다.
함수임	매개변수 탭에서 매개변수로 지정된 실행 변수를 포함할 수 있는 함수로 규칙을 사용합니다. 규칙을 함수로 사용하면 함수처럼 값이 반환됩니다. [함수임]이 선택된 모든 비즈니스 규칙이 [함수 선택기]에 함수로 표시됩니다.


표 2-3 General Ledger 옵션

등록정보	설명
시작 날짜	이 날짜 후에는 비즈니스 규칙을 실행할 수 있습니다.
종료 날짜	이 날짜 전에는 비즈니스 규칙을 실행할 수 있습니다.

- 규칙 디자이너의 다음 탭에서 정보를 입력하거나 검토합니다.
  - 글로벌 범위** - 비즈니스 규칙 구성요소의 공통 차원을 정의합니다.  
비즈니스 규칙 구성요소의 공통 차원 정의를 참조하십시오.
  - 변수** - 비즈니스 규칙에 사용된 변수에 대한 정보를 검토하고 정의합니다.  
비즈니스 규칙에 런타임 프롬프트 값이 포함된 경우에만 변수 탭이 표시됩니다.  
Groovy 이외의 규칙인 경우, 하나 이상의 멤버 유형의 변수는 ASO(집계 저장영역 옵션) 유형의 Planning 큐브에서만 지원되는 변수 유형입니다.
  - 스크립트** - 규칙에 대해 생성된 스크립트를 표시합니다.  
이 탭의 내용은 변경할 수 없습니다. 스크립트를 변경하려면 디자이너 옆 드롭다운에서 스크립트 편집을 선택합니다.  
비즈니스 규칙에 대한 스크립트를 보고 편집하는 기능은 Oracle Hyperion Financial Management, Planning 및 Oracle Essbase 블록 저장영역에서만 사용할 수 있습니다.
  - 사용** - 규칙을 사용하는 규칙과 규칙 세트를 표시합니다.  
이 탭의 정보는 편집할 수 없습니다. 규칙을 생성할 때 규칙은 기본적으로 규칙이나 규칙 세트에서 사용되지 않습니다.
  - 매개변수** - 매개변수로 사용할 실행 변수를 선택합니다.

매개변수로 사용할 실행 변수 선택을 참조하십시오.

- 오류 및 경고 - 비즈니스 규칙에 대해 스크립트 진단을 실행하려면  를 누릅니다. 이 작업을 수행하면 Calculation Manager에서 비즈니스 규칙 스크립트를 분석하고 다음을 표시합니다.
  - 검증 오류, 규칙이 검증되지 않은 경우.  
검증 오류를 해결하려면 오류를 마우스 오른쪽 버튼으로 누르고 **스크립트에 표시** 또는 **디자이너에 표시**를 선택합니다. **디자이너에 표시**를 선택하면 Calculation Manager에서 오류가 있는 구성요소를 표시하며, 여기에서 필요한 내용을 변경하고 규칙을 저장한 다음 스크립트 진단을 다시 실행할 수 있습니다. 구성요소를 편집하려면 디자이너 보기에 있어야 합니다.
  - 요약, 경고 및 블록, 규칙이 검증되는 경우.  
경고 또는 블록 항목을 누르고 **스크립트에 표시** 또는 **디자이너에 표시**를 선택합니다. 디자이너 뷰에서는 구성요소만 편집할 수 있습니다.
    - \* **요약** - 계산의 데이터 셀 수, 데이터 통과 수 및 부정확하게 사용되는 차원 수 등의 통계가 있습니다.
    - \* **경고** - 모든 회소 차원을 셀 참조에 대해 지정할지 여부, 셀이 회소 멤버를 참조할지 여부 및 지정이 다른 데이터 블록의 회소 차원 멤버를 참조할지 여부 등의 정보가 있습니다.
    - \* **블록** - "각 Fix 명령문에 대해 Fix 명령문의 영향을 받을 수 있는 잠재적 예상 블록 수" 등의 정보가 있습니다.

5. 규칙을 저장하려면  를 누릅니다.

## 매개변수로 사용할 실행 변수 선택

### 주:

실행 변수를 매개변수로 사용하도록 선택하는 경우 Oracle Hyperion Financial Management에만 적용됩니다.

실행 변수를 매개변수로 사용하려면 다음을 수행합니다.

1. 비즈니스 규칙 열기.
2. 규칙 디자이너의 **매개변수** 탭에서 실행 변수, **매개변수임** 순으로 선택합니다.  
비즈니스 규칙을 처음 열면 [매개변수] 탭이 비어 있습니다. 비즈니스 규칙에 구성요소를 추가하는 경우 그리고 이러한 구성요소에 변수를 사용되는 구성요소가 있는 경우 이 탭에 변수가 표시됩니다.  
실행 변수에 규칙 세트 범위가 있는 경우(비즈니스 규칙에서 동일한 규칙 세트 내의 다른 비즈니스 규칙으로 전달됨) **매개변수임**이 기본적으로 선택됩니다.  
변수에 규칙 범위가 있는 경우(변수가 생성된 비즈니스 규칙에서만 변수를 사용할 수 있고 다른 비즈니스 규칙으로 전달되지 않음) **매개변수임**이 선택되지 않습니다.
3. 전달 방식에서 **참조 기준** 또는 **값 기준**을 선택합니다.

 주:

비즈니스 규칙에 **매개변수임**이 선택된 실행 변수가 하나 이상 있는 경우 다른 비즈니스 규칙이나 동일한 비즈니스 규칙에 이 비즈니스 규칙을 포함하면 **매개변수** 탭에서 실행 변수의 값을 지정할 수 있습니다.

## 매개변수를 비즈니스 규칙으로 사용

 주:

매개변수를 비즈니스 규칙으로 사용하는 경우 Oracle Hyperion Financial Management에만 적용됩니다.

규칙 디자이너의 **매개변수** 탭을 사용하여 비즈니스 규칙에 사용할 매개변수를 지정합니다. Oracle Hyperion Calculation Manager는 규칙에서 실행 변수가 사용되는지 여부를 결정합니다. 실행 변수의 범위는 규칙 세트나 규칙일 수 있습니다. 비즈니스 규칙에 사용된 모든 실행 변수는 **매개변수** 탭의 그리드에 나열됩니다.

그리드의 [매개변수] 열에서 기본적으로 규칙 세트 범위가 지정된 모든 실행 변수의 확인란이 선택됩니다. 이러한 실행 변수는 매개변수로 전달됩니다. 기본적으로 규칙 범위가 지정된 실행 변수의 확인란은 선택 취소되어 있습니다. 이러한 변수는 변수가 사용되는 비즈니스 규칙에만 적용되기 때문입니다. 해당 실행 변수는 매개변수로 전달되지 않습니다. 그리드에 나열된 모든 변수의 기본 설정을 변경할 수 있습니다.

전달 방식 열의 기본 설정은 **참조 기준**이지만 **값 기준**으로 매개변수를 전달하도록 선택할 수 있습니다.

 주:

**매개변수** 탭에 표시되는 변수는 자동으로 생성됩니다. 변수의 범위를 추가, 삭제 또는 변경하여 변수 디자이너에서 이러한 변수를 변경하는 경우 비즈니스 규칙에 영향을 주는 변경사항은 규칙의 **매개변수** 탭을 다시 선택해야 표시됩니다.


매개변수를 비즈니스 규칙으로 사용하려면 다음을 수행합니다.

1. 규칙 디자이너에서 비즈니스 규칙을 생성하거나 엽니다.
2. **등록정보**에서 **함수임**이 선택되어 매개변수로 지정된 실행 변수를 포함할 수 있는 함수로 규칙이 처리되는지 확인합니다.

**함수임** 선택 여부에 관계없이 모든 비즈니스 규칙에 매개변수 전달을 사용할 수 있습니다. 옵션을 선택하지 않으면 매개변수 전달 기능이 있는 정상적인 비즈니스 규칙입니다. 옵션을 선택하면 비즈니스 규칙이 함수로 처리됩니다.

3. 비즈니스 규칙을 저장합니다.
4. 두 번째 비즈니스 규칙을 엽니다.
5. **기존 객체**에서 첫 번째 비즈니스 규칙의 사본을 두 번째 비즈니스 규칙의 순서도로 끌어서 놓습니다.



6. 매개변수 탭에서 비즈니스 규칙으로 사용할 매개변수의 값을 입력합니다.
7. 를 누릅니다.

## 비즈니스 규칙 열기

Oracle Hyperion Calculation Manager를 열 때 기본적으로 표시되는 시스템 뷰에서 비즈니스 규칙을 열 수 있습니다.

사용자정의 뷰, 필터 뷰 또는 배포 뷰에서 규칙을 열 수도 있습니다.

비즈니스 규칙을 열려면 다음 중 하나를 수행합니다.

- 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
- 규칙을 두 번 누릅니다.

### 주:

Oracle Financials Cloud에서 다른 관리자가 규칙을 잠금 경우 "objectname이 (가) 읽기 전용 모드로 열렸습니다. 현재 FINUSER2이(가) 객체를 편집하고 있습니다."라는 메시지가 표시됩니다. 이 경우 다음 단계를 완료합니다.

1. **확인**을 누릅니다.
2. 규칙 또는 규칙 세트를 마우스 오른쪽 버튼으로 누르고 **잠금 해제**를 선택합니다. 다음 메시지가 표시됩니다.

선택한 항목을 잠금 해제하시겠습니까? 편집 중인 사용자의 변경사항이 손실됩니다.

3. **확인**을 눌러 규칙을 열거나 **취소**를 눌러 규칙을 열지 않고 닫습니다.

규칙이 잠금 해제된 경우 규칙을 편집하고 있던 관리자에게 메시지가 표시되지 않습니다. 원래 관리자가 규칙을 편집하여 저장한 후 규칙을 잠금 해제한 관리자가 변경하여 저장하는 경우, 마지막으로 저장한 내용이 유지되고 원래 관리자가 편집한 내용은 덮어씁니다.

## 비즈니스 규칙 편집

비즈니스 규칙에서 구성요소(공식, 스크립트, 조건, 범위 및 루프 포함)를 추가, 제거 또는 변경하여 비즈니스 규칙의 구조를 편집할 수 있습니다.


또한 비즈니스 규칙 구성요소의 등록정보 및 비즈니스 규칙 자체의 등록정보를 편집할 수 있습니다.

편집할 수 있는 비즈니스 규칙 등록정보는 다음과 같습니다.


- 이름 및 캡션
- 설명(Description) 및 설명(Comment)
- 차원, 멤버 및 변수
- For Oracle Hyperion Financial Management 애플리케이션의 경우 비즈니스 규칙에 로그 파일과 타이머가 있는지 여부, 비즈니스 규칙이 스크립트 생성 중에 사용 안함으로 설정되는지 여부 및 비즈니스 규칙이 매개변수로 작동하는지 여부

비즈니스 규칙을 편집하려면 다음을 수행합니다.

1. 규칙을 엽니다.
2. 규칙 디자이너에서 새 구성요소를 추가하고 규칙 순서도의 기존 구성요소를 복사하고 삭제합니다.
  - 구성요소를 추가하려면 **새 객체** 또는 **기존 객체**에서 순서도로 객체를 끌어 놓습니다. 기존 공식 또는 스크립트 구성요소를 순서도에 추가하면 기본적으로 공식 또는 스크립트가 공유 객체가 됩니다. 구성요소를 공유하지 않으려면 공식 또는 스크립트 구성요소의 **등록정보**에서 **공유 확인란**을 선택 취소합니다. [스크립트 및 공식 구성요소 공유](#)를 참조하십시오.
  - 구성요소를 복사하려면 구성요소를 마우스 오른쪽 버튼으로 누르고 **복사**를 선택한 후 순서도에 붙여넣습니다.
  - 구성요소를 삭제하려면 구성요소를 마우스 오른쪽 버튼으로 누르고 **제거**를 선택합니다.
3. **등록정보**에서 규칙의 등록정보를 편집합니다.
 

규칙에 구성요소를 추가하고 순서도에서 구성요소 간에 이동하게 되면 등록정보가 변경됩니다. 특정 구성요소에 대한 등록정보를 입력하려면 순서도에서 구성요소를 선택합니다. [구성요소를 사용하여 비즈니스 규칙 및 템플릿 디자인](#)을 참조하십시오.
4.  을 눌러 변경사항을 저장합니다.

**팁:**

비즈니스 규칙에서 구성요소를 편집할 때 구성요소 아이콘의 크기와 순서도에 표시되는 세부정보의 양을 늘리거나 줄일 수 있습니다. 이 작업을 하려면  을 사용하여 순서도 내에서 확대 및 축소합니다.

순서도가 작은 크기로 표시되는 경우 구성요소 캡션이 표시되지 않지만 마우스 포인터를 아이콘 위로 이동하여 해당 캡션을 읽을 수 있습니다. 순서도에 있는 구성요소의 크기에 관계 없이 등록정보를 표시할 구성요소를 선택할 수 있습니다.

## 스크립트 모드로 비즈니스 규칙 편집

비즈니스 규칙의 스크립트를 편집할 수 있습니다.

기본적으로 비즈니스 규칙은 규칙의 그래픽 플로우를 디자인하는 규칙 디자이너를 사용하여 그래픽 모드로 생성합니다. 비즈니스 규칙을 생성하여 저장한 후 그래픽 모드나 스크립트 모드에서 편집할 수 있습니다. 비즈니스 규칙을 스크립트 모드에서 편집할 경우 나중에 그래픽 모드 편집으로 돌아갈 수 있습니다.

스크립트 모드로 비즈니스 규칙을 편집하려면 다음을 수행합니다.

1. 규칙을 엽니다.
2. 규칙 디자이너에 있는 **디자이너** 옆의 드롭다운에서 **스크립트 편집**을 선택합니다.
 

그래픽 모드로 다시 전환하려면 **스크립트 편집** 옆의 드롭다운에서 **디자이너**를 선택합니다.

그래픽 모드 편집에서 스크립트 모드로 전환할 때 비즈니스 규칙에 공유 구성요소(예: 공식 및 스크립트 구성요소) 또는 템플릿이 포함되어 있으면 비즈니스 규칙 스크립트의 스크립트

편집기에는 공유 구성요소 및 템플릿에 대한 참조만 포함됩니다. 공유 구성요소 및 템플릿의 생성된 스크립트 전체가 맨아래 창의 스크립트 탭에 표시됩니다.

다음 사항에 유의하십시오.

- ASO 규칙에서 **스크립트 편집**을 선택하면 스크립트가 Groovy로 자동 변환됩니다. 그런 후에 규칙을 저장하면 스크립트가 Groovy에서 저장되며 그래픽으로 다시 변환할 수 없습니다. 그래픽 형식을 유지하려면 **다른 이름으로 저장**을 선택하고 Groovy 스크립트가 포함된 규칙을 다른 이름으로 저장합니다.
- BSO 규칙에서 [스크립트 편집]을 선택하면 그래픽 규칙이 계산 스크립트 구문을 기반으로 하는 스크립트 모드로 변환됩니다. 스크립트를 Groovy로 변환하려면 스크립트 유형으로 **Groovy 스크립트**를 선택합니다.
- 스크립트 모드로 편집할 때는 다음 키워드를 사용하지 마십시오.

- /\*STARTCOMPONENT\*/
- /\*STARTCOMPONENT:SCRIPT\*/
- /\*STARTCOMPONENT:FORMULA\*/
- /\*ENDCOMPONENT\*/
- /\*STARTRULE\*/
- /\*ENDRULE\*/
- /\*STARTTEMPLATE\*/
- /\*ENDTEMPLATE\*/

3. 원하는 대로 스크립트를 편집합니다.

표 2-4 스크립트 모드로 편집할 때 사용 가능한 옵션



















아이콘	설명	참조
	스크립트 라인 번호를 표시하거나 숨깁니다. 라인 번호는 기본적으로 표시됩니다.	
	함수와 매개변수 삽입 함수에 대해 자동 제안이 제공됩니다. 함수의 처음 몇 글자를 입력하고 Ctrl+스페이스를 누르면 제안이 표시됩니다. 제안 중 하나를 선택하고 Enter 키를 눌러 함수를 완성합니다.	함수 작업
	횡차원 멤버를 삽입합니다.	구성요소에 멤버 및 함수 추가
	차원에서 선택한 멤버 범위를 삽입합니다.	구성요소에 멤버 및 함수 추가
	변수를 삽입합니다.	변수 작업
	스마트 목록을 삽입합니다.	스마트 목록 작업
	템플릿을 편집합니다.	시스템 템플릿 작업

표 2-4 (계속) 스크립트 모드로 편집할 때 사용 가능한 옵션

아이콘	설명	참조
	설명	
	설명 제거	
	설명을 표시하거나 숨깁니다.	
	코드 형식을 지정합니다. 코드 형식을 지정하면 스크립트 라인이 그룹화 및 들여쓰기 되고 빈 라인이 제거됩니다.	
	구문을 확인합니다.	
	페이지 바깥쪽으로 스크롤되는 스크립트의 긴 라인을 동일한 페이지에 여러 라인으로 표시하도록 스크립트를 줄 바꿈합니다.	
	코드 완료 On/코드 완료 Off	
	코드를 완료하기 위한 제안을 사용하거나 제외합니다.	
	예를 들어 코드 완료를 시작하려면 FIX를 입력한 후 SHIFT + Space를 입력합니다. 그러면 Oracle Hyperion Calculation Manager에서 () ENDFIX를 입력합니다.	
	스크립트에서 텍스트 문자열을 찾아 바꿉니다.	비즈니스 규칙 스크립트에서 텍스트 문자열 검색
	텍스트 문자열 검색 시 대/소문자 구분을 적용할 것인지 전체 단어 일치 적용할 것인지 선택합니다.	비즈니스 규칙 스크립트에서 텍스트 문자열 검색

4. 를 누릅니다.

자습서 비디오:



[스크립트 검토](#)

## 비즈니스 규칙 저장

비즈니스 규칙을 이 규칙이 생성된 애플리케이션 및 애플리케이션 유형에 저장합니다.

참조:


- [비즈니스 규칙 저장](#)
- [비즈니스 규칙을 다른 이름으로 저장](#)

## 비즈니스 규칙 저장

비즈니스 규칙을 저장하면 이 규칙을 생성한 애플리케이션 및 애플리케이션 유형에 저장됩니다. 비즈니스 규칙을 저장한 후에는 배포, 검증 및 실행할 수 있습니다. 비즈니스 규칙을 Oracle Hyperion Calculation Manager에서 배포 및 검증할 수 있으며 Oracle Hyperion Planning(Planning 비즈니스 규칙의 경우), Oracle General Ledger(Oracle General Ledger 비즈니스 규칙의 경우) 및 Calculation Manager 또는 Oracle Essbase Administration Services(Oracle Essbase 비즈니스 규칙의 경우)에서 실행할 수 있습니다.

### 주:

Oracle Hyperion Financial Management에서는 비즈니스 규칙 세트만 실행할 수 있고 비즈니스 규칙은 실행할 수 없습니다.

비즈니스 규칙을 생성하거나 편집한 후 저장하려면  를 누릅니다.

### 주:

비즈니스 규칙을 저장한 후 시스템 뷰에서 보려면 애플리케이션 목록을 새로고쳐야 할 수 있습니다. 이렇게 하려면 시스템 뷰에서 임의의 노드나 객체를 마우스 오른쪽 버튼으로 누르고 **새로고침**을 선택합니다.

## 비즈니스 규칙을 다른 이름으로 저장

[다른 이름으로 저장]을 사용하여 비즈니스 규칙을 다른 이름으로 저장할 수 있습니다. 또한 [다른 이름으로 저장]을 사용하여 규칙을 한 규칙 세트에서 같은 규칙 세트 유형 내의 다른 규칙 세트로 복사할 수도 있습니다. [다른 이름으로 저장]에서는 원래 비즈니스 규칙의 사본을 원본과 구분되도록 다른 이름으로 생성합니다.

비즈니스 규칙을 다른 이름으로 저장하려면 다음을 수행합니다.

1. 규칙을 두 번 누릅니다.
2. 규칙 디자이너에서 **작업**, **다른 이름으로 저장** 순으로 선택합니다.
3. **다른 이름으로 저장**에 규칙의 새 이름을 입력하고 **애플리케이션**을 선택합니다.
4. 다음 태스크 중 하나를 수행합니다.
  - 규칙이 Oracle Hyperion Financial Management 규칙인 경우 **계산 유형**을 선택합니다.
  - 규칙이 Oracle Hyperion Planning 규칙인 경우 **계획 유형**을 선택합니다.
  - 규칙이 Oracle Essbase 또는 Oracle General Ledger 규칙인 경우 **데이터베이스**를 선택합니다.

 주:

다른 이름으로 저장하는 규칙의 애플리케이션 유형은 변경할 수 없습니다. 규칙을 다른 이름으로 저장할 때 규칙이 속하는 애플리케이션과 계산, 계획 유형 또는 데이터베이스를 변경할 수 있습니다.


## 5. 확인을 누릅니다.

새 규칙이 **시스템** 뷰의 애플리케이션 목록에 추가됩니다.

## 비즈니스 규칙 실행

비즈니스 규칙을 실행하기 전에 먼저 열어야 합니다.

비즈니스 규칙을 실행하려면 다음을 수행합니다.

1. 규칙을 두 번 누르거나 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙이 열리면  (실행 버튼)을 누릅니다.

## 비즈니스 규칙 복사

비즈니스 규칙을 다른 애플리케이션 및 계획 유형으로 복사할 수 있습니다.

비즈니스 규칙을 다른 애플리케이션으로 복사하면 다른 이름으로 새 비즈니스 규칙이 생성됩니다. 비즈니스 규칙에서 사용된 모든 변수도 새 애플리케이션으로 복사됩니다.

비즈니스 규칙을 다른 애플리케이션 및 계획 유형으로 복사하려면 다음을 수행합니다.

1. **시스템** 뷰에서 규칙을 마우스 오른쪽 버튼으로 누르고 **복사 위치**를 선택합니다.
2. **복사 위치**에서 비즈니스 규칙의 새 이름을 입력하거나 기본 이름을 적용하고 애플리케이션, 애플리케이션 유형 및 계획 유형을 선택합니다.  
둘 이상의 애플리케이션 및 계획 유형에 비즈니스 규칙을 복사할 수 없습니다.
3. 공유 구성요소가 있는 비즈니스 규칙을 다른 애플리케이션으로 복사하고 공유 구성요소도 해당 애플리케이션으로 복사하려는 경우 **공유 구성요소 복사**를 선택합니다.

이렇게 하면 복사된 애플리케이션에 공유 구성요소가 생성되고 복사된 규칙이 해당 애플리케이션과 계획 유형의 공유 구성요소를 사용합니다. **공유 구성요소 복사**를 선택하지 않으면 공유 구성요소가 규칙에서 복사되지만 복사된 규칙이 원래 규칙에 속한 애플리케이션 및 계획 유형의 공유 구성요소를 사용합니다.

변수가 있는 비즈니스 규칙을 다른 애플리케이션으로 복사하면 복사 위치 애플리케이션에 변수가 생성됩니다. 복사된 애플리케이션에 변수가 이미 있으면 복사 시 변수가 규칙 레벨 변수로 생성됩니다.

4. 공유 애플리케이션을 복사하는 경우 **기존 객체 바꾸기**를 선택하여 이름이 같은 객체를 덮어씁니다.

## 비즈니스 규칙에서 검색


비즈니스 규칙 스크립트에서 텍스트 문자열을 검색합니다. 그래픽 비즈니스 규칙에서 텍스트를 검색하고 바꿉니다.

**참조:**

- [비즈니스 규칙 스크립트에서 텍스트 문자열 검색](#)
- [그래픽 비즈니스 규칙에서 텍스트 검색 및 바꾸기](#)

## 비즈니스 규칙 스크립트에서 텍스트 문자열 검색

비즈니스 규칙 스크립트에서 텍스트 문자열을 검색하려면 다음을 수행합니다.

1. 스크립트가 포함된 비즈니스 규칙을 엽니다.
2. 규칙 디자이너에서 **스크립트** 탭을 선택합니다.
3. 검색할 텍스트를 입력하고  를 누릅니다.

## 그래픽 비즈니스 규칙에서 텍스트 검색 및 바꾸기

문자열을 검색할 때 Oracle Hyperion Calculation Manager는 순서도에서 선택된 구성요소 뒤에 있는 첫 번째 구성요소에서 검색을 시작하여 순서도의 끝까지 검색한 다음 순서도의 처음부터 새로 시작하여 순서도에서 선택된 구성요소에 도달할 때까지 검색합니다.

일치하는 텍스트 문자열을 찾은 후 다른 항목을 검색하려면 다시 검색을 시작하여 다음 항목을 찾아야 합니다.

그래픽 비즈니스 규칙에서 텍스트를 검색하고 바꾸려면 다음을 수행합니다.

1. 비즈니스 규칙을 엽니다.
2. 규칙 디자이너에서 비즈니스 규칙의 **시작** 또는 **종료** 아이콘이나 비즈니스 규칙 순서도 내 구성요소를 마우스 오른쪽 버튼으로 누르고 다음 중 하나를 수행합니다.
  - 텍스트 문자열 인스턴스를 찾으려면 **찾기**를 선택합니다.
  - 텍스트 문자열 인스턴스를 찾아 바꾸려면 **모두 바꾸기**를 선택합니다.

## 비즈니스 규칙 인쇄

비즈니스 규칙의 등록정보, 순서도 및 해당 구성요소의 세부정보를 인쇄할 수 있습니다.

예를 들어 할당 비용에 대한 공식 구성요소가 포함된 비즈니스 규칙을 인쇄하는 경우 출력에서는 공식 구문, 공식을 구성하는 함수 및 변수, 규칙의 순서도에 있는 단계 요약 (그래픽 형식이 아님) 및 규칙의 등록정보를 보여 줍니다.

스크립트에만 있는 비즈니스 규칙을 인쇄하려면 다음을 수행합니다.

- PDF 파일의 규칙 세부정보 섹션에는 비즈니스 규칙 스크립트가 있습니다.
- PDF 파일의 순서도 섹션에는 시작 및 종료 노드만 있습니다.
- PDF 파일의 플로우 요약 섹션이 비어 있습니다.

 **주:**

비즈니스 규칙 세트 또는 구성요소는 인쇄할 수 없습니다. 인쇄하는 비즈니스 규칙에서 사용되는 경우 구성요소를 인쇄할 수 있습니다.

비즈니스 규칙을 인쇄하려면 다음을 수행합니다.

1. **시스템 뷰**에서 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **인쇄**를 선택합니다.
2. **인쇄 미리보기** 대화상자에 정보를 입력하고 **PDF 생성**을 누릅니다.

비즈니스 규칙의 PDF 파일이 Adobe Acrobat에서 열립니다.

## 비즈니스 규칙 삭제

다른 규칙이나 규칙 세트에서 사용되지 않는 비즈니스 규칙만 삭제할 수 있습니다.

규칙이 사용되고 있는 경우에는 삭제하기 전에 사용 중인 규칙 및 규칙 세트에서 비즈니스 규칙을 제거하거나 사용 중인 규칙 및 규칙 세트를 위한 사본을 만들어야 합니다.

비즈니스 규칙을 삭제하려면 다음을 수행합니다.

1. **시스템 뷰**에서 규칙을 마우스 오른쪽 버튼으로 누르고 **사용 표시**를 선택하여 다른 규칙 또는 규칙 세트에서 해당 규칙이 사용되고 있지 않은지 확인합니다.
2. 다른 규칙 및 규칙 세트에서 규칙이 사용되고 있지 않은 경우 규칙을 마우스 오른쪽 버튼으로 누르고 **삭제**를 선택합니다.
3. **확인**을 눌러 규칙의 삭제를 확인합니다.


## 비즈니스 규칙 구성요소의 공통 차원 정의

비즈니스 규칙을 열고 각 차원에 공통되는 멤버, 변수 및 함수를 선택하여 공통 차원을 정의합니다.

### 주:

비즈니스 규칙 구성요소의 공통 차원 정의는 Oracle Hyperion Planning, Oracle General Ledger 및 Oracle Essbase에만 적용됩니다.

비즈니스 규칙 구성요소의 공통 차원을 정의하려면 다음을 수행합니다.

1. 비즈니스 규칙 열기.
2. 규칙 디자이너의 순서도에서 **시작** 또는 **종료**를 누릅니다.
3. **글로벌 범위** 탭에서 차원을 선택하고 를 누른 후 각 차원에 공통되는 멤버, 변수 및 함수를 선택합니다.

차원에 대해 선택하는 값은 규칙이 실행될 때 계산됩니다.

변수를 선택하는 경우 변수를 변경하면 변경사항이 규칙에서 업데이트되도록 변수를 규칙에 동적으로 링크하려면 **동적으로 변수 링크**를 선택합니다.

### 주:

**그리드 값 제외**를 선택하여 비즈니스 규칙에서 "Fix/EndFix" 스크립트 대신 "Exclude/EndExclude" 스크립트를 생성합니다.



# 비즈니스 규칙 분석 및 디버깅

규칙을 실행하고 통계 정보를 캡처하여 비즈니스 규칙을 분석합니다. 규칙을 실행하고 스크립트를 검사하여 비즈니스 규칙을 디버그합니다.

## 참조:

- [비즈니스 규칙 분석](#)
- [비즈니스 규칙의 스크립트 분석](#)
- [비즈니스 규칙의 스크립트 비교](#)
- [변경된 Business Rule을 저장된 Business Rule과 비교](#)
- [비즈니스 규칙 디버깅](#)
- [비즈니스 규칙에서 구성요소 사용 안 함](#)

## 비즈니스 규칙 분석

비즈니스 규칙을 분석하는 경우 비즈니스 규칙을 실행하고 구성요소를 실행하는 데 걸린 시간, 실행된 횟수, 구성요소 실행 전과 후의 멤버 교차 값과 같은 통계 정보를 캡처합니다. 분석에 소요되는 시간은 사용 중인 시스템 메모리에 따라 다릅니다.

### 주:

Oracle Hyperion Planning 또는 Oracle Essbase를 사용하는 경우에만 비즈니스 규칙을 분석하고 디버깅할 수 있습니다.

### 주:

멤버 범위, 멤버 블록, 고정 루프 또는 조건 구성요소에는 분석 기능을 사용할 수 없습니다.

비즈니스 규칙을 분석하려면 다음을 수행합니다.


1. 비즈니스 규칙 열기.
2. 규칙 디자이너에서 **작업**, **분석** 순으로 선택합니다.
3. **분석 기준 입력** 대화상자에서 다음을 수행합니다.
  - 분석에 표시할 항목 수를 입력하거나 **모두 표시**를 선택하여 항목을 표시합니다. 여기에 입력한 숫자가 계산하고 표시할 멤버 교차 수입입니다. 예를 들어 애플리케이션에 다음 차원이 있다고 가정합니다.
    - 연도 = 12개 멤버
    - 측정항목 = 50,525개 멤버
    - 제품 = 450개 멤버
    - 시장 = 2000개 멤버


- 시나리오 = 4개 멤버

이 예에서 계산할 멤버 교차 수는 2,182,680,000(12 x 50,525 x 450 x 2,000 x 4 또는 연도 x 측정항목 x 제품 x 시장 x 시나리오)입니다.

- 각 차원에 대해 분석할 멤버를 선택합니다.

멤버를 선택하는 데 다음 옵션을 사용할 수 있습니다.

- 멤버 선택기 대화상자에 액세스하려면  를 누릅니다.

- 차원을 선택하고  를 누릅니다.

- 멤버를 입력합니다.

차원당 하나의 멤버만 선택할 수 있습니다.

4. 확인을 눌러 분석을 시작합니다.

5. 분석이 완료되면 비즈니스 규칙 순서도에서 구성요소를 누르고 분석 정보 탭을 선택합니다.

#### **팁:**

분석 정보 데이터를 내보내려면 **분석 정보** 탭의 그리드를 마우스 오른쪽 버튼으로 누르고 **익스포트**를 선택합니다.

#### **주:**

Essbase 최적화 때문에 분석에 올바른 시작 값과 끝 값이 표시되지 않을 수도 있습니다. 분석에 하나 이상의 교차에 대한 시작 값과 끝 값이 표시되지 않는 경우 개수 및 경과 시간이 정확하지 않을 수 있습니다.

## 추적된 차원의 멤버 숨기기

추적 중인 차원 멤버를 숨기려면 **분석 정보** 탭에서 **추적 중인 차원 숨기기**를 선택합니다. 이 옵션을 선택하지 않으면 차원에 대해 처리된 모든 멤버가 표시됩니다.

**분석 정보** 탭에 표시된 이전 및 이후 값은 추적 중인 멤버 값을 변경한 멤버를 나타냅니다. 이전 및 이후 값은 추적 중인 멤버 값입니다. 예를 들어 제품 200-30을 추적하는 경우 제품 200에 대한 라인이 표시될 수 있습니다. 제품 200 라인에 표시된 값은 200-30의 값입니다.

## 통계 정보 캡처

비즈니스 규칙 분석 시 통계 정보를 캡처하려면 다음 사항에 유의하십시오.


- Oracle Essbase에 대한 내부 호출을 멤버 블록으로 돌려싸야 합니다.
- Fix 문에서 한 차원만 비워 두고 해당 차원의 추적 멤버를 사용하지 않으면 이 차원이나 하위 멤버 중 하나가 멤버 블록에 사용됩니다.
- Fix 문에서 한 차원만 비워 두고 해당 차원에서 동적 계산 멤버인 추적 멤버를 지정할 경우 멤버 블록은 이 차원이나 동적 계산 또는 레이블 전용 멤버가 아닌 하위 멤버 중 하나입니다.
- **분석 정보** 탭에 추적 멤버 값이 표시되며 멤버 블록에서 사용된 멤버도 표시됩니다.

## 비즈니스 규칙의 스크립트 분석

그래픽 비즈니스 규칙 또는 스크립트 비즈니스 규칙을 실행하고 해당 스크립트를 분석할 수 있습니다.

비즈니스 규칙의 스크립트를 분석하려면 다음을 수행합니다.

1. **시스템 뷰**에서 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙이 표시되면 **작업, 스크립트 분석** 순으로 선택합니다.  
비즈니스 규칙이 실행됩니다.
3. 비즈니스 규칙에 런타임 프롬프트 변수가 포함되어 있으면 **RTP 값 입력** 대화상자에서 런타임 프롬프트 값을 입력한 다음 비즈니스 규칙에 값을 적용할 것인지 여부를 지정하고 **확인**을 누릅니다.
4. **스크립트 분석** 탭에서 스크립트를 분석합니다.

스크립트의 라인에 분석 정보가 있는 경우 해당 라인 옆에 이 표시됩니다. 각 라인을 선택하여 화면 오른쪽의 **분석 정보** 영역에 있는 분석 정보를 검토합니다. 라인에 대한 분석 정보가 없으면 "분석 정보를 사용할 수 없음"이 표시됩니다. 실행 시간이 가장 긴 라인이 빨간색으로 강조 표시됩니다.

**등록정보** 영역에 이름, 애플리케이션, 계획 유형, 스크립트를 분석하는 데 걸린 시간(밀리초) 등 비즈니스 규칙에 대한 정보가 표시됩니다.

5. 스크립트 정보 검토가 완료되면 를 눌러 **스크립트 분석** 탭을 닫습니다.

## 비즈니스 규칙의 스크립트 비교

비즈니스 규칙의 스크립트를 비교하여 차이점, 변경사항 또는 그래픽 비즈니스 규칙의 스크립트를 확인할 수 있습니다.

### 주:

한 번에 비즈니스 규칙을 세 개 이상 비교할 수는 없습니다. 또한 한 비즈니스 규칙의 여러 버전을 비교할 수도 없습니다. 예를 들어 한 비즈니스 규칙의 이전 버전을 방금 저장한 버전과 비교할 수 없습니다.

두 비즈니스 규칙의 스크립트를 비교하려면 다음을 수행합니다.

1. **시스템 뷰**에서 비즈니스 규칙 두 개를 선택합니다.
2. 선택된 두 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **스크립트 비교**를 선택합니다.

**스크립트 비교** 창은 다음과 같이 표시됩니다.

- 스크립트에 추가된 경우 라인이 연한 회색 배경으로 표시됩니다.
- 스크립트에서 삭제된 경우 라인이 진한 회색 배경으로 표시됩니다.
- 스크립트가 변경된 경우 라인이 노란색 배경으로 표시됩니다.
- 첫 번째 열에는 스크립트의 라인 번호가 표시됩니다.

- 두 번째 열에는 다음 세 문자 중 하나가 표시됩니다.
    - \*는 왼쪽 열에 표시된 스크립트 라인이 오른쪽 열에 표시된 스크립트 라인과 다름을 나타냅니다..
    - +는 왼쪽 열에 표시된 스크립트와 비교한 결과 오른쪽 열에 표시된 스크립트에 새 라인이 추가되었음을 나타냅니다.
    - -는 왼쪽 열에 표시된 스크립트와 비교한 결과 오른쪽 열에 표시된 스크립트에서 라인이 삭제되었음을 나타냅니다.
  - 세 번째 열과 네 번째 열에는 두 스크립트가 표시됩니다.
  - 창 아래쪽은 맨위 창에서 선택한 스크립트의 전체 라인을 표시하는 두 개의 창으로 나누어져 있습니다. 첫 번째 창에는 왼쪽 열의 스크립트가 표시되고 두 번째 창에는 오른쪽 열의 스크립트가 표시됩니다.
3. 스크립트 비교가 완료되면 **확인**을 누릅니다.

## 변경된 Business Rule을 저장된 Business Rule과 비교

저장된 버전의 비즈니스 규칙과 변경된 버전의 비즈니스 규칙 스크립트를 보고 두 버전의 차이를 확인하거나, 변경한 내용을 확인하거나, 저장된 그래픽 비즈니스 규칙과 변경된 그래픽 비즈니스 규칙 스크립트를 나란히 볼 수 있습니다.

저장된 버전과 변경된 버전의 **business rule**을 비교하려면 다음을 수행합니다.

1. 그래픽 또는 계산 스크립트 비즈니스 규칙을 엽니다.
2. **편집, 저장된 버전과 비교** 순으로 선택합니다.

**스크립트 비교** 창은 다음과 같이 표시됩니다.

- 비즈니스 규칙 또는 계산 스크립트에 추가된 경우 라인이 연한 회색 배경으로 표시됩니다.
  - 비즈니스 규칙 또는 계산 스크립트에서 삭제된 경우 라인이 진한 회색 배경으로 표시됩니다.
  - 비즈니스 규칙 또는 계산 스크립트가 변경된 경우 라인이 노란색 배경으로 표시됩니다.
  - 첫 번째 열에는 비즈니스 규칙 또는 계산 스크립트의 라인 번호가 표시됩니다.
  - 두 번째 열에는 다음 세 문자 중 하나가 표시됩니다.
    - \*는 왼쪽 열에 표시된 비즈니스 규칙 또는 계산 스크립트 라인이 오른쪽 열에 표시된 비즈니스 규칙 또는 계산 스크립트 라인과 다름을 나타냅니다.
    - +는 왼쪽 열에 표시된 비즈니스 규칙 또는 계산 스크립트와 비교한 결과 오른쪽 열에 표시된 비즈니스 규칙 또는 계산 스크립트에 새 라인이 추가되었음을 나타냅니다.
    - -는 왼쪽 열에 표시된 비즈니스 규칙 또는 계산 스크립트와 비교한 결과 오른쪽 열에 표시된 비즈니스 규칙 또는 계산 스크립트에서 라인이 삭제되었음을 나타냅니다.
  - 세 번째와 네 번째 열에는 저장된 버전과 변경된 버전의 비즈니스 규칙 또는 계산 스크립트가 표시됩니다.
  - 창 아래쪽은 맨위 창에서 선택한 비즈니스 규칙 또는 계산 스크립트의 전체 라인을 표시하는 두 개의 창으로 나누어져 있습니다. 첫 번째 창의 왼쪽 열에는 저장된 버전의 비즈니스 규칙 또는 계산 스크립트가 표시되고, 두 번째 창의 오른쪽 열에는 수정된 버전의 비즈니스 규칙 또는 계산 스크립트가 표시됩니다. 스크립트에서 행을 선택하여 맨아래 창에 표시합니다. 저장된 버전은 맨위에 있고 수정된 버전은 맨아래에 있습니다.
3. **확인**을 누릅니다.

## 비즈니스 규칙 디버깅

비즈니스 규칙 디버깅에는 비즈니스 규칙을 실행하고 스크립트를 라인 단위로 검사하여 스크립트 실행 방식을 확인하는 작업이 포함되어 있습니다.

다음 애플리케이션에 대한 비즈니스 규칙을 디버깅할 수 있습니다.

- Oracle Essbase 블록 저장영역 애플리케이션(쓰기 권한이 있는 경우)
- Oracle Hyperion Planning 애플리케이션(실행 권한이 있는 경우)

### 주:

디버깅 중에는 비즈니스 규칙을 편집할 수 없습니다.

비즈니스 규칙을 디버깅하려면 다음을 수행합니다.


1. 비즈니스 규칙 열기.
2. 규칙 디자이너에서 **작업, 디버그** 순으로 선택합니다.  
비즈니스 규칙의 스크립트가 스크립트 디버거에 표시됩니다. 스크립트의 각 명령문이 개별 라인에 표시됩니다.
3. 비즈니스 규칙 스크립트에서 명령문을 디버깅합니다.

명령문 디버깅 시 다음을 수행할 수 있습니다.

- 스크립트 실행을 중지할 위치에 중단점을 삽입하고 제거하여 명령문의 멤버 교차 값을 검사할 수 있습니다. 중단점에서 실행이 중지되면 명령문에 있는 모든 멤버의 교차 값이 표시됩니다.

중단점을 추가하려면 중단점의 오른쪽을 마우스 오른쪽 버튼으로 누르고 **중단점 추가**를 선택합니다. 지정 및 조건문에 중단점을 추가할 수 있습니다. 중단점이 회색으로 표시되는 스크립트 라인에만 중단점을 추가할 수 있습니다. 중단점을 제거하려면 중단점을 마우스 오른쪽 버튼으로 누르고 **중단점 제거**를 선택합니다.

- 조건이 충족되는 경우에만 명령문 실행을 중지하는 중단점에 조건을 추가할 수 있습니다. 중단점이 있는 명령문에 사용된 멤버만 조건에 사용할 수 있습니다.

중단점에 조건을 추가하려면 중단점의 오른쪽을 마우스 오른쪽 버튼으로 누르고 **조건 추가**를 선택합니다. **조건 추가** 대화상자에서  을 누른 다음 **조건 작성기**에 조건을 입력합니다.



조건을 편집하려면 중단점을 마우스 오른쪽 버튼으로 누르고 **조건 편집**을 선택합니다.

- 중단점이 있는 명령문을 디버깅할 수 있습니다.

중단점이 있는 명령문을 디버깅하려면 명령문을 마우스 오른쪽 버튼으로 누르고 **디버그 시작**을 선택합니다. 디버깅 중인 명령문은 강조 표시됩니다. 명령문의 멤버, 중단점, 실행 전과 후의 멤버 교차 값이 다음 탭에 표시됩니다.

- **멤버(둘 이상)** - 디버그 중단점의 현재 멤버 교차가 표시됩니다. 다음 멤버 교차를 보려면 **디버깅 다시 시작**을 누릅니다.
- **중단점** - 중단점이 포함된 스크립트의 표현식을 보여줍니다. **중단점의 값** 탭에 표현식 멤버와 디버그 실행 전후의 멤버 값이 표시됩니다.

 **팁:**

디버깅을 다시 시작하려면  을 누릅니다. 멤버 교차가 모두 디버깅될 때까지  을 계속 눌러야 합니다. 멤버 교차가 모두 디버깅되면 "스크립트 디버깅이 완료되었습니다." 메시지가 표시됩니다.

## 비즈니스 규칙에서 구성요소 사용 안 함

비즈니스 규칙에서 구성요소를 사용하지 않을 수 있습니다. 예를 들어 검증 오류를 발생시키는 구성요소를 분석하기 위해 검증에서 구성요소를 제외하려 할 수 있습니다.

비즈니스 규칙에서 구성요소를 사용하지 않으려면 다음을 수행합니다.

1. 비즈니스 규칙 순서도에서 구성요소를 선택합니다.
2. 등록정보에서 **사용 안 함**을 선택합니다.

구성요소를 사용하지 않도록 설정하고 해당 구성요소가 속한 비즈니스 규칙을 저장하면 비즈니스 규칙 순서도에서 해당 구성요소 아이콘이 사용하지 않도록 설정됩니다. 비활성화된 구성요소로 이동하거나 이 구성요소 내에서 생성한 모든 구성요소는 비활성화된 구성요소의 [비활성화] 상태를 상속하며 비즈니스 규칙의 일부로 계산되지 않습니다.

 **주:**

조건 구성요소는 비활성화할 수 없습니다. 공유된 구성요소가 사용되지 않도록 설정하면 사용되지 않도록 설정된 비즈니스 규칙에 대해서만 구성요소가 사용되지 않습니다.

## 사전 정의된 Workforce Planning, Capital Asset Planning 및 Human Capital Management 비즈니스 규칙 로드

Oracle Hyperion Planning 애플리케이션 관리에서 생성된 Oracle Hyperion Workforce Planning, Oracle Hyperion Capital Asset Planning 또는 Oracle Fusion Human Capital Management 애플리케이션으로 작업하는 경우 Planning에서 애플리케이션을 생성하고 초기화할 때 사전 정의된 비즈니스 규칙이 애플리케이션에 로드됩니다. 이러한 사전 정의된 비즈니스 규칙을 사용하면 애플리케이션에서 웹 양식으로 작업할 때 사전 정의된 계산을 수행할 수 있습니다.

이러한 사전 정의된 비즈니스 규칙을 변경하거나 삭제하는 경우 Oracle Hyperion Calculation Manager에서 비즈니스 규칙을 재로드하려 할 수 있습니다. 이 절차는 Wrkforce, Capex 및/또는 인적 자본 계획 등의 계획 유형이 활성화된 Planning 애플리케이션에 사전 정의된 비즈니스 규칙을 로드하는 데 사용됩니다.

사전 정의된 비즈니스 규칙을 로드하려면 다음을 수행합니다.

1. **시스템 뷰**에서 Planning 애플리케이션 유형을 확장하고 사전 정의된 비즈니스 규칙을 로드할 애플리케이션을 마우스 오른쪽 버튼으로 누른 다음 **사전 정의된 규칙 로드**를 선택합니다.

비즈니스 규칙이 계획 유형에 로드됩니다.

2. **확인**을 눌러 사전 정의된 규칙이 성공적으로 로드되었다는 메시지를 종료합니다.

규칙을 로드한 애플리케이션 및 계획 유형의 규칙 폴더에서 사전 정의된 규칙을 볼 수 있습니다.

## 비즈니스 규칙에 대한 바로가기 생성

바로가기를 사용하면 각 애플리케이션에서 규칙의 사본을 생성하지 않고도 애플리케이션 간에 비즈니스 규칙을 공유할 수 있습니다. 규칙에 대한 바로가기를 생성하면 바로가기를 사용하여 규칙을 상호 참조할 수 있습니다.

또한 바로가기를 사용하여 규칙이 생성된 애플리케이션이 아닌 애플리케이션에 비즈니스 규칙을 배포할 수도 있습니다. 기본적으로 비즈니스 규칙을 배포할 때 해당 규칙이 생성된 애플리케이션에 배포됩니다.

비즈니스 규칙에 대한 바로가기를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙을 마우스 오른쪽 버튼으로 누르고 **바로가기 생성**을 선택합니다.
2. **바로가기 생성**에서 규칙에 대한 바로가기를 생성할 애플리케이션을 확장합니다.
3. 바로가기를 생성할 계산 유형, 계획 유형 및 데이터베이스 옆의 옵션 버튼을 선택합니다.

동일한 애플리케이션 유형에 속하는 여러 애플리케이션에서 여러 개의 계산 유형, 계획 유형 및 데이터베이스를 선택할 수 있습니다.

비즈니스 규칙 이름은 애플리케이션에서 고유해야 하므로 비즈니스 규칙을 생성한 애플리케이션은 [바로가기 생성] 대화상자에 표시되지 않습니다. 다른 애플리케이션에서만 비즈니스 규칙에 대한 바로가기를 생성할 수 있습니다. 또한 바로가기를 생성할 규칙과 동일한 이름의 비즈니스 규칙이 포함된 애플리케이션은 [바로가기 생성] 대화상자에 표시되지 않습니다.

4. **확인**을 누릅니다.

규칙에 대한 바로가기가 선택한 계산 유형, 계획 유형 또는 데이터베이스로 복사됩니다. 바로가기는 규칙 아이콘의 왼쪽 아래에 위쪽 화살표로 표시됩니다.

## 비즈니스 규칙 또는 비즈니스 규칙 세트 새로고침

시스템 뷰, 사용자정의 뷰 및 배포 뷰에서 애플리케이션 목록의 아무 레벨이나 새로고칠 수 있습니다.

애플리케이션 유형(Oracle Hyperion Financial Management, Oracle Hyperion Planning 또는 Oracle Essbase), 애플리케이션, 계산 유형, 계획 유형, 데이터베이스, 여러 규칙 세트나 규칙 또는 한 규칙 세트나 규칙을 새로고칠 수 있습니다.

기본적으로 애플리케이션 목록의 노드를 새로고치면 이 노드에 속하는 모든 규칙, 규칙 세트, 구성요소 및 템플릿이 새로고쳐집니다. 그러나 애플리케이션 내의 규칙 세트나 규칙을 새로고치더라도 애플리케이션 목록의 상위 레벨이나 다른 애플리케이션에 속한 규칙 세트 또는 규칙은 새로고쳐지지 않습니다.

예를 들어 Planning 애플리케이션 및 계획 유형 내의 규칙을 새로고치면 해당 애플리케이션 및 계획 유형 내의 모든 규칙도 새로고쳐지지만 다른 계획 유형 또는 다른 Planning 애플리케이션 내에 있는 규칙은 새로고쳐지지 않습니다.

비즈니스 규칙 또는 규칙 세트를 새로고치려면 규칙 또는 규칙 세트를 마우스 오른쪽 버튼으로 누르고 **새로고침**을 선택합니다.



**팁:**

새로고칠 비즈니스 규칙이 포함된 Planning 애플리케이션 유형, 애플리케이션 또는 계획 유형을 마우스 오른쪽 버튼으로 누르고 **새로고침**을 선택할 수도 있습니다.

## 비즈니스 규칙 또는 규칙 세트에 대한 사용 표시

비즈니스 규칙 또는 비즈니스 규칙 세트를 사용하고 있는 규칙, 템플릿 및 규칙 세트를 표시합니다.

규칙이나 규칙 세트를 삭제할 때 이를 사용하고 있는 객체를 알아야 하는 경우 규칙 또는 규칙 세트에 대한 사용을 보는 것이 유용합니다.

비즈니스 규칙 또는 규칙 세트에 대한 사용을 표시하려면 다음을 수행합니다.

1. 규칙 또는 규칙 세트를 마우스 오른쪽 버튼으로 누르고 **사용 표시**를 선택합니다.
2. **사용** 대화상자에서 정보를 검토하고 **확인**을 누릅니다.

## 객체의 소유자 변경

객체(규칙, 규칙 세트, 템플릿, 공식 또는 스크립트 구성요소)가 속한 애플리케이션이 배포된 경우 시스템 뷰에서 객체의 소유자를 변경할 수 있습니다. 사용자가 객체의 소유권을 변경하지 않는 한 기본적으로 객체를 생성하는 사용자가 소유자입니다. 모든 객체를 편집할 수 있는 관리자를 제외하고 사용자는 자신이 소유한 객체만 편집할 수 있습니다.

객체의 소유자를 변경하려면 다음을 수행합니다.

1. **시스템 뷰**에서 애플리케이션 유형, 애플리케이션 순으로 확장합니다.
2. 다음 중 한 가지 태스크를 수행합니다.
  - **Oracle Hyperion Financial Management 사용자만 해당:** 규칙 세트의 소유권을 변경하려면 계산 유형 및 **규칙 세트**를 확장합니다.
  - **Oracle Hyperion Planning, Oracle Essbase 및 Oracle General Ledger 사용자만 해당:** 규칙 세트의 소유권을 변경하려면 **규칙 세트**를 확장합니다.



**주:**

Financial Management 애플리케이션의 경우 애플리케이션 내의 각 계산 유형에 대해 [규칙 세트] 노드가 있지만, Planning, Essbase 및 Oracle General Ledger 애플리케이션의 경우 각 애플리케이션에 대해 하나의 [규칙 세트] 노드만 있습니다.

- 규칙의 소유권을 변경하려면 계산 유형, 계획 유형 또는 데이터베이스를 확장한 후 **규칙**을 확장합니다.
- 공식 구성요소, 스크립트 구성요소 또는 사용자정의 템플릿의 소유권을 변경하려면 계산 유형, 계획 유형 또는 데이터베이스를 확장한 다음, 소유권을 변경할 객체에 따라 **공식**, **스크립트** 또는 **템플릿**을 확장합니다.
3. 객체를 마우스 오른쪽 버튼으로 누르고 **소유자 변경**을 선택합니다.



4. **소유자 변경**에서 객체 소유권을 이전할 소유자를 선택합니다.
5. **확인**을 누릅니다.  
소유권이 지정된 사용자는 객체를 편집할 수 있습니다.

# 3

## 비즈니스 규칙 세트 디자인

### 참조:

- [비즈니스 규칙 세트 정보](#)  
동시에 또는 순서대로 실행할 수 있는 비즈니스 규칙 또는 비즈니스 규칙 세트를 결합하여 비즈니스 규칙 세트를 생성합니다.
- [비즈니스 규칙 세트 생성](#)  
시스템, 필터, 사용자정의 및 배포 뷰 그리고 규칙 세트 및 템플릿 디자이너에서도 비즈니스 규칙 세트를 생성할 수 있습니다.
- [비즈니스 규칙 세트 열기](#)  
시스템, 필터, 배포 및 사용자정의 뷰에서 비즈니스 규칙 세트를 열 수 있습니다.
- [비즈니스 규칙 세트 내의 비즈니스 규칙 열기](#)  
비즈니스 규칙 세트 내에서 비즈니스 규칙을 열 수 있습니다.
- [비즈니스 규칙 세트에 비즈니스 규칙 추가](#)  
비즈니스 규칙 세트에 비즈니스 규칙을 추가하는 경우 규칙 세트의 규칙을 순차적으로 또는 동시에 실행할 수 있습니다.
- [비즈니스 규칙 세트에서 비즈니스 규칙 제거](#)  
비즈니스 규칙 세트에서 비즈니스 규칙을 제거해도 규칙이 삭제되지는 않습니다. 규칙은 데이터베이스에서 규칙 세트와 별개로 존재합니다.
- [비즈니스 규칙 세트를 다른 애플리케이션에 복사](#)  
비즈니스 규칙 세트를 다른 애플리케이션으로 복사하는 경우 동일한 애플리케이션 유형으로 복사합니다.
- [비즈니스 규칙 세트 저장](#)  
비즈니스 규칙 세트를 저장하면 이 규칙 세트를 생성한 애플리케이션 및 애플리케이션 유형에 저장됩니다.
- [비즈니스 규칙 세트 삭제](#)  
다른 비즈니스 규칙 세트에서 사용되지 않는 비즈니스 규칙 세트만 삭제할 수 있습니다.

## 비즈니스 규칙 세트 정보

동시에 또는 순서대로 실행할 수 있는 비즈니스 규칙 또는 비즈니스 규칙 세트를 결합하여 비즈니스 규칙 세트를 생성합니다.

규칙 세트 생성 가능 여부는 지정된 역할에 따라 결정됩니다. *Oracle Planning and Budgeting Cloud*를 사용하는 *Oracle Fusion Cloud EPM*을 참조하십시오.

규칙 세트를 생성 및 저장한 후 Oracle Hyperion Calculation Manager에서 검증 및 배포합니다. 그런 다음 애플리케이션에서 규칙 세트를 실행할 수 있습니다.

 **팁:**

비즈니스 규칙 세트를 생성할 때 작업 중인 비즈니스 규칙, 구성요소, 템플릿 및 변수를 열어 두어야 할 수 있습니다. Calculation Manager에서는 이러한 객체를 탭 인터페이스에 표시하므로 비즈니스 규칙 세트를 생성하는 동안 탭 간에 쉽게 이동할 수 있습니다. Calculation Manager 내에서 탭 10개를 열어 둘 수 있지만 성능을 최적화하려면 동시에 여는 객체 수가 10개를 초과하지 않는 것이 좋습니다.

 **주:**

규칙 세트는 Oracle Essbase 집계 저장영역 또는 블록 저장영역 애플리케이션에서 지원되지 않습니다. 규칙 세트는 Oracle General Ledger에 사용되는 Essbase 합산 저장영역 애플리케이션에서 지원됩니다.

## 비즈니스 규칙 세트 생성

시스템, 필터, 사용자정의 및 배포 뷰 그리고 규칙 세트 및 템플릿 디자이너에서도 비즈니스 규칙 세트를 생성할 수 있습니다.

비즈니스 규칙 세트를 생성하려면 다음을 수행합니다.

1. **규칙 세트**를 마우스 오른쪽 버튼으로 누른 다음 **새로 작성**을 선택합니다.
2. **새 규칙 세트** 대화상자에서 이름을 입력하고 애플리케이션 유형 및 애플리케이션을 선택한 후 **확인**을 누릅니다.

시스템 뷰에서 규칙 세트를 추가하는 경우 애플리케이션 유형 및 애플리케이션이 자동으로 채워집니다.

Oracle Hyperion Financial Management 애플리케이션 유형과 애플리케이션을 선택한 경우 **계산 유형**을 선택합니다.

3. **기존 객체**에서 기존 규칙 및 규칙 세트를 순서대로 끌어서 놓습니다.

Oracle Hyperion Planning 및 Oracle Essbase 규칙 세트에는 다른 애플리케이션에서 생성했거나 다른 애플리케이션에 배포된 규칙 및 규칙 세트가 포함될 수 있습니다. Financial Management 규칙 및 규칙 세트는 동일한 계산 유형 또는 일반 계산 유형에 속해야 합니다.

Financial Management 애플리케이션의 규칙은 규칙 세트 내에서 순서대로 실행되므로 규칙의 순서가 중요합니다.

4. 다음 태스크를 수행합니다.
  - 규칙 세트에서 규칙 순서를 변경하려면 위쪽 또는 아래쪽 화살표를 누르거나 규칙을 마우스 오른쪽 버튼으로 누르고 **위로 이동** 또는 **아래로 이동**을 선택합니다.
  - 규칙 세트에서 규칙을 제거하려면 규칙을 마우스 오른쪽 버튼으로 누르고 **제거**를 선택합니다.
  - 편집할 비즈니스 규칙을 열려면 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
5. **등록정보**에서 규칙 세트에 대한 등록정보를 입력합니다.

규칙 세트 디자이너에서는 생성하고 있는 규칙 세트 내의 규칙을 선택하는 경우 해당 규칙의 등록정보가 규칙 세트의 등록정보 대신 **등록정보**에 표시됩니다.

규칙 세트의 규칙 및 규칙 세트를 동시에 실행하려면 **병렬 실행 사용**을 선택합니다. 기본적으로 규칙 세트에 속한 규칙 및 규칙 세트는 순서대로 실행됩니다. 즉, 규칙 세트의 각 규칙 또는 규칙 세트가 오류 없이 실행되어야 다음 규칙 또는 규칙 세트가 실행될 수 있습니다.

규칙 세트에 중첩된 규칙 세트가 포함되어 있고 중첩된 규칙 세트의 **병렬 실행 사용** 설정이 상위 규칙 세트와 다른 경우 중첩된 규칙 세트의 설정이 적용됩니다. 예를 들어 **병렬** 처리로 플래그 지정된 ruleset1이 있고 이 규칙 세트에 rule1, rule2 및 ruleset2(**순차** 처리로 플래그 지정)가 포함되어 있는 경우 ruleset1이 병렬 처리로 플래그 지정되어 있더라도 ruleset2의 규칙 및 규칙 세트는 순차적으로 처리됩니다.

#### Financial Management 사용자만 해당:

- **로깅 사용**을 선택합니다. 그러면 규칙 세트의 규칙에 로그 텍스트가 있는 경우 규칙 세트가 실행될 때 로그 텍스트가 로그 파일에 포함됩니다. 규칙, 규칙 세트 및 구성요소에 대해 로깅을 사용할 수 있습니다. 그러나 비즈니스 규칙의 공식문을 사용하여 포함할 로그 텍스트(있는 경우)를 지정합니다.

#### 주:

규칙 세트에 대해서는 로깅을 사용하지 않지만 규칙 세트에 속한 규칙 및 구성요소에 대해 로깅을 사용하는 경우 규칙 세트에 대한 로그 설정이 규칙 및 구성요소에 대한 로그 설정을 재정의하므로 로그 파일이 생성되지 않습니다.

- **타이머 사용**을 선택합니다. 그러면 규칙 세트가 실행될 때 규칙 세트를 처리하는 데 걸린 시간이 로그 파일에 기록됩니다. 규칙, 규칙 세트 및 구성요소에 대해 타이머를 사용할 수 있습니다. 타이머가 활성화된 모든 객체에 대한 처리 시간이 로그 파일에 포함됩니다. 예를 들어 타이머가 사용되는 규칙 세트가 있고 이 규칙 세트에 타이머가 사용되는 규칙 세 개가 있는 경우 규칙 세트 및 규칙 세트의 각 규칙을 처리하는 데 걸린 시간이 기록됩니다.
- **실행 사용**을 선택합니다. 그러면 Financial Management에서 규칙 세트를 실행할 수 있습니다.

#### 주:

Financial Management에서는 비즈니스 규칙을 실행할 수 없습니다.

**Oracle General Ledger 사용자만 해당:** 시작 날짜 및 종료 날짜 기능을 사용하여 비즈니스 규칙 세트가 실행될 수 있는 날짜 범위를 지정합니다. 이 날짜 범위를 벗어날 경우 비즈니스 규칙 세트를 실행할 수 없습니다.

- **실행 날짜**를 선택하여 해당 날짜 이후에는 비즈니스 규칙 세트가 실행될 수 있는 실행 날짜를 선택할 수 있는 달력에 액세스합니다.
  - **종료 날짜**를 선택하여 해당 날짜 이전에는 비즈니스 규칙 세트가 실행될 수 있는 종료 날짜를 선택할 수 있는 달력에 액세스합니다.
6. **사용** 탭에서 이 규칙 세트를 사용하는 규칙 세트(있는 경우)를 검토합니다.

규칙 세트를 생성하면 기본적으로 다른 규칙 세트에서 사용되지 않으므로 비즈니스 규칙 세트를 처음에 생성할 때는 이 탭이 비어 있습니다.

규칙 세트에서 규칙을 선택하면 비즈니스 규칙 세트의 이름을 볼 수 있습니다.

7. 변수 탭에서 규칙 세트에 사용된 변수에 대한 정보를 검토하고 입력합니다.

변수 병합을 선택하여 이 규칙 세트 내의 규칙에서 사용되는 동일한 변수의 모든 인스턴스를 병합합니다. 그러면 규칙이 실행될 때 각 변수의 첫 번째 인스턴스만 표시됩니다. 변수를 병합하지 않으면 각 변수의 모든 인스턴스가 표시됩니다.

변수 병합을 선택하는 경우 사용자가 런타임 프롬프트에 입력하는 첫 번째 값이 검증 및 실행 동안 해당 런타임 프롬프트의 모든 후속 항목에 사용됩니다.

 주:

Planning, Essbase 또는 Oracle General Ledger 변수에 대해 변수를 병합할 수 있습니다.

8.  를 누릅니다.

## 비즈니스 규칙 세트 열기

시스템, 필터, 배포 및 사용자정의 뷰에서 비즈니스 규칙 세트를 열 수 있습니다.

비즈니스 규칙 세트를 열려면 다음 중 하나를 수행합니다.

- 규칙 세트를 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
- 규칙 세트를 두 번 누릅니다.

## 비즈니스 규칙 세트 내의 비즈니스 규칙 열기

비즈니스 규칙 세트 내에서 비즈니스 규칙을 열 수 있습니다.


- 시스템 뷰에서 규칙 집합, 원하는 규칙 집합 순으로 확장하고 원하는 규칙을 두 번 누릅니다.
- 편집할 수 있도록 비즈니스 규칙 세트가 열리면 규칙 세트 내에서 원하는 규칙을 두 번 누릅니다.

## 비즈니스 규칙 세트에 비즈니스 규칙 추가

비즈니스 규칙 세트에 비즈니스 규칙을 추가하는 경우 규칙 세트의 규칙을 순차적으로 또는 동시에 실행할 수 있습니다.

같은 애플리케이션 유형에 속한 비즈니스 규칙 세트에 비즈니스 규칙을 추가할 수 있습니다. Oracle Hyperion Financial Management에서는 규칙 세트의 규칙이 순차적으로 실행되므로 규칙 세트에서 규칙의 순서가 중요합니다. 다른 애플리케이션 유형에 속하는 규칙 세트에서, 규칙 세트의 규칙을 순차적으로 또는 동시에 실행할 수 있습니다.

비즈니스 규칙 세트에 비즈니스 규칙을 추가하려면 다음을 수행합니다.

1. 규칙 세트를 여십시오.
2. 기존 객체에서 기존 규칙을 규칙 세트로 끕니다.
3.  를 누릅니다.

## 비즈니스 규칙 세트에서 비즈니스 규칙 제거

비즈니스 규칙 세트에서 비즈니스 규칙을 제거해도 규칙이 삭제되지는 않습니다. 규칙은 데이터베이스에서 규칙 세트와 별개로 존재합니다.

비즈니스 규칙 세트에서 비즈니스 규칙을 제거하려면 다음을 수행합니다.

1. 규칙 세트를 엽니다.
2. 원하는 규칙을 마우스 오른쪽 버튼으로 누르고 **제거**를 선택합니다.

## 비즈니스 규칙 세트를 다른 애플리케이션에 복사

비즈니스 규칙 세트를 다른 애플리케이션으로 복사하는 경우 동일한 애플리케이션 유형으로 복사합니다.

애플리케이션 유형 외에도, 계산 유형(Oracle Hyperion Financial Management 규칙 세트의 경우), (Oracle Hyperion Planning 규칙 세트의 경우) 또는 데이터베이스(Oracle General Ledger에서 사용된 Oracle Essbase 규칙 세트의 경우)가 동일한 애플리케이션에 비즈니스 규칙 세트를 복사할 수 있도록 합니다.

### 주:

비즈니스 규칙 세트는 Oracle General Ledger에서 사용되는 합산 저장영역 애플리케이션 이외의 다른 Essbase 합산 저장영역 또는 블록 저장영역 애플리케이션에서 지원되지 않습니다.

비즈니스 규칙 세트를 다른 애플리케이션에 복사하려면 다음을 수행합니다.

1. **시스템 뷰**에서 비즈니스 규칙 세트를 마우스 오른쪽 버튼으로 누르고 **복사 위치**를 선택합니다.
2. **복사 위치**에서 비즈니스 규칙 세트의 새 이름을 입력하거나 기본 이름을 적용하고 애플리케이션, 계산 유형, 계획 유형 또는 데이터베이스를 선택합니다.  
둘 이상의 애플리케이션 및 계산 또는 계획 유형에 비즈니스 규칙 세트를 복사할 수 없습니다.
3. 공유 구성요소를 복사할지 여부 및 같은 이름의 기존 구성요소를 바꿀지 여부를 선택합니다.  
이러한 옵션은 비즈니스 규칙 세트에 새 애플리케이션으로 복사할 공유 구성요소가 있는 경우에만 사용됩니다.
4. **확인**을 누릅니다.


선택한 애플리케이션 및 계산 또는 계획 유형에 새 비즈니스 규칙 세트가 추가됩니다. 시스템 뷰에서 이 규칙 세트를 보려면 애플리케이션 목록을 새로고쳐야 합니다. 애플리케이션 목록을 새로 고치려면 툴바의 **새로고침** 아이콘을 누릅니다. 규칙 세트 노드 또는 애플리케이션 목록에서 이 노드의 상위 레벨을 새로 고쳐 새 규칙 세트를 표시할 수도 있습니다. **비즈니스 규칙 또는 비즈니스 규칙 세트 새로고침**을 참조하십시오.

## 비즈니스 규칙 세트 저장

비즈니스 규칙 세트를 저장하면 이 규칙 세트를 생성한 애플리케이션 및 애플리케이션 유형에 저장됩니다.

규칙 세트를 저장한 후에는 배포, 검증 및 실행할 수 있습니다.

Oracle Hyperion Calculation Manager에서 비즈니스 규칙 세트를 배포 및 검증할 수 있으며 Oracle Hyperion Financial Management(Financial Management 규칙 세트의 경우), Oracle Hyperion Planning(Planning 규칙 세트의 경우) 또는 Oracle General Ledger(Oracle General Ledger에서 사용되는 Oracle Essbase 규칙 세트의 경우에만)에서 비즈니스 규칙 세트를 실행할 수 있습니다.

비즈니스 규칙 세트를 생성하거나 편집한 후 저장하려면  를 누릅니다.

### 주:

비즈니스 규칙 세트를 저장한 후 시스템 뷰에서 보려면 애플리케이션 목록을 새로고쳐야 할 수 있습니다. 이렇게 하려면 시스템 뷰에서 임의의 노드나 객체를 마우스 오른쪽 버튼으로 누르고 **새로고침**을 선택합니다.

## 비즈니스 규칙 세트 삭제

다른 비즈니스 규칙 세트에서 사용되지 않는 비즈니스 규칙 세트만 삭제할 수 있습니다.

규칙 세트가 사용되고 있는 경우 삭제하기 전에 사용하고 있는 비즈니스 규칙 세트에서 제거하거나 사용하고 있는 비즈니스 규칙 세트용 사본을 만들어야 합니다.

비즈니스 규칙 세트를 삭제하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 세트를 마우스 오른쪽 버튼으로 누르고 **사용 표시**를 선택하여 다른 규칙 세트에서 해당 규칙 세트가 사용되고 있지 않은지 확인합니다.
2. 다른 규칙 세트에서 규칙 세트가 사용되고 있지 않은 경우 규칙 세트를 마우스 오른쪽 버튼으로 누르고 **삭제**를 선택합니다.
3. 확인을 눌러 규칙 세트의 삭제를 확인합니다.

# 4

## 시스템 템플릿 작업

### 참조:

- [시스템 템플릿 정보](#)  
시스템 템플릿은 계산을 수행하는 사전 정의된 템플릿입니다.
- [시스템 템플릿 표시](#)  
시스템 템플릿이 표시되는 위치는 *그래픽* 템플릿인지, *규칙* 템플릿인지에 따라 다릅니다.
- [시스템 템플릿 사용](#)  
시스템 템플릿을 사용하려면 템플릿을 선택하고 순서도 또는 스크립트로 끌어 놓습니다.
- [템플릿 순서도 표시](#)  
시스템 템플릿에서 생성된 스크립트를 검토하는 경우 템플릿 순서도를 보는 것이 유용한 경우가 있습니다.
- [시스템 템플릿을 사용자정의 템플릿으로 저장](#)  
시스템 템플릿의 콘텐츠를 사용자정의할 수 있습니다.
- [비즈니스 규칙에서 시스템 템플릿 제거](#)  
비즈니스 규칙에서 시스템 템플릿을 제거할 수 있습니다.

## 시스템 템플릿 정보

시스템 템플릿은 계산을 수행하는 사전 정의된 템플릿입니다.

시스템 템플릿에서 수행하는 계산에는 데이터 지우기, 데이터 복사, 금액, 수량 및 단가 계산, 데이터 배포, 값 할당, 데이터 집계, 성능을 최적화하는 스크립트 명령 입력, 보고 통화로 계정 값 변환 등이 포함됩니다.

시스템 템플릿이 포함될 수 있는 위치는 다음과 같습니다.

- 그래픽 또는 스크립트 비즈니스 규칙
- 그래픽 또는 스크립트 사용자정의 템플릿

비즈니스 규칙 또는 사용자정의 템플릿의 구성요소인 시스템 템플릿에는 매개변수를 입력하는 일련의 단계가 포함되어 있습니다. 이러한 매개변수는 템플릿 논리에 통합되어 있으며 비즈니스 규칙 또는 템플릿 내에 계산 스크립트를 생성합니다. 이렇게 하면 템플릿을 더욱 쉽게 사용하고 잠재적인 계산 스크립트 구문 오류를 줄일 수 있습니다.

Oracle Hyperion Calculation Manager에서는 모든 시스템 템플릿을 마법사로 사용할 수 있습니다. 마법사에서 시스템 템플릿 작업을 수행하면 템플릿을 더욱 쉽게 사용하고 잠재적인 계산 스크립트 구문 오류를 줄일 수 있습니다.

시스템 템플릿의 프롬프트는 마법사에서 선택한 내용을 기준으로 필터링됩니다. 예를 들어 집계 시스템 템플릿은 집계할 밀집 차원을 선택하는 단계를 표시하지만, 집계할 밀집 차원이 없는 경우 밀집 차원 전체 및 부분에 대한 단계가 표시되지 않습니다.



 주:

시스템 템플릿은 Oracle Hyperion Financial Management, Oracle Hyperion Planning 블록 저장영역 애플리케이션 및 Oracle Essbase 블록 저장영역 애플리케이션에서 사용할 수 있습니다. Essbase ASO 애플리케이션에서는 시스템 템플릿을 지원하지 않습니다.

## 시스템 템플릿 표시

시스템 템플릿이 표시되는 위치는 *그래픽* 템플릿인지, *규칙* 템플릿인지에 따라 다릅니다.

*그래픽* 규칙 또는 템플릿을 생성하거나 여는 경우 다음 위치 중 하나에 시스템 템플릿이 표시됩니다.

- **새 객체** - 시스템 템플릿이 **새 객체**에 개별 객체로 표시됩니다.
- **기존 객체** - 시스템 템플릿이 **시스템 템플릿** 머리글 아래 **기존 객체**에 표시됩니다.

*스크립트* 규칙 또는 템플릿을 생성하거나 여는 경우 **기존 객체**에만 시스템 템플릿이 표시됩니다.

시스템 템플릿 마법사에 액세스하려면 다음을 수행합니다.

- *그래픽* 규칙 또는 템플릿의 경우 원하는 템플릿을 선택하고 **시작** 및 **종료** 사이의 순서도로 끌어 놓습니다.
- *스크립트* 규칙 또는 템플릿의 경우 템플릿을 선택하고 스크립트로 끌어 놓습니다.

## 시스템 템플릿 사용

시스템 템플릿을 사용하려면 템플릿을 선택하고 순서도 또는 스크립트로 끌어 놓습니다.

## 참조:

- [데이터 지우기 템플릿 사용](#)  
데이터 지우기 템플릿을 사용하여 큐브의 멤버에서 데이터를 지웁니다.
- [데이터 복사 템플릿 사용](#)  
데이터 복사 템플릿을 사용하여 데이터베이스의 한 위치에서 다른 위치로 데이터를 복사합니다.
- [금액-수량-단가 템플릿 사용](#)  
금액-수량-단가 템플릿을 사용하여 금액, 수량, 단가에 대해 입력한 세 멤버 중 하나를 계산합니다.
- [할당 - 레벨별 템플릿 사용](#)  
레벨별 할당 템플릿을 사용하여 데이터베이스 아웃라인의 한 레벨에서 다른 레벨로 할당할 수 있습니다.
- [할당 템플릿 사용](#)  
할당 템플릿을 사용하여 균등하게 또는 지정된 드라이버를 기반으로 소스에서 대상으로 값을 할당합니다.
- [집계 템플릿 사용](#)  
집계 템플릿을 사용하여 지정하는 멤버의 데이터 값을 집계합니다.

- **SET 명령 템플릿 사용**  
SET 명령 템플릿을 사용하여 계산 스크립트의 성능을 최적화하는 스크립트 명령을 입력할 수 있습니다.
- **통화 변환 템플릿 사용**  
통화 변환 템플릿은 시스템 유형 계정 또는 선택한 계정을 사용하여 계정 값을 보고 통화로 변환합니다.
- **데이터 익스포트-임포트 템플릿 사용**

## 데이터 지우기 템플릿 사용

데이터 지우기 템플릿을 사용하여 큐브의 멤버에서 데이터를 지웁니다.

데이터를 지우려면 값을 지울 멤버를 지정합니다. 한 멤버 또는 멤버 블록에 대한 데이터를 지울 수 있습니다.

예를 들어 "실제"에서 "예측"으로 데이터를 복사하고 변경하기 전에 데이터 지우기 템플릿을 사용하여 예측 데이터를 지울 수 있습니다.

데이터 지우기 템플릿을 사용하려면 다음을 수행합니다.

1. 비즈니스 규칙 또는 템플릿을 생성하거나 엽니다.
2. 다음 중 하나를 수행합니다.
  - *그래픽* 규칙 또는 템플릿의 경우 새 객체에서 데이터 지우기를 선택하고 시작 및 종료 사이의 순서도로 끌어 놓습니다.
  - *스크립트* 규칙 또는 템플릿의 경우 시스템 템플릿을 확장하고 데이터 지우기를 선택한 후 스크립트의 원하는 영역으로 끌어 놓습니다.
3. 데이터 지우기 마법사에서 필요한 정보를 입력합니다.

표 4-1 데이터 지우기 마법사 단계




단계	설명
데이터 선택	<p>각 차원에 대해 멤버를 하나 이상 선택하여 지울 데이터를 정의합니다. 다음 옵션을 사용할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• 값으로 모든 차원을 채울 변수를 선택하려면 사전 정의된 선택 사용 옆의  를 누릅니다.</li> <li>• 각 차원의 멤버 및 함수를 선택하려면 멤버 선택기 옆의  를 누릅니다.</li> <li>• 변수, 멤버 또는 함수를 선택하려면 차원을 선택하고  를 누릅니다.</li> </ul> <p><b>Clearblock 옵션 선택</b>에서 데이터 지우기 방법을 정의하는 옵션을 선택합니다.</p> <ul style="list-style-type: none"> <li>• 모두 - 데이터 블록을 모두 지웁니다.</li> <li>• 상위 - 통합된 데이터 블록만 지웁니다.</li> <li>• 동적 - 동적 계산 및 저장 멤버 조합에서 파생된 값이 포함된 블록을 지웁니다.</li> <li>• 비입력 - 파생 값이 포함된 데이터 블록을 지웁니다. 완전히 계산 작업에 의해 생성된 블록에만 적용되고 값이 로드된 블록에는 적용되지 않습니다.</li> <li>• N/A - 데이터 블록을 지우고 멤버를 #Missing으로 설정합니다.</li> </ul>

표 4-1 (계속) 데이터 지우기 마법사 단계

단계	설명
설정	HSP_Rates가 "Sparse"로 설정된 다중 통화 애플리케이션이나 단일 통화 애플리케이션이 있는 경우 <b>데이터 선택</b> 에서 선택된 멤버가 <b>설정</b> 에 표시됩니다. 드롭다운을 누르고 단일 멤버를 선택한 차원을 선택합니다(밀집 차원 권장). <b>참고:</b> 선택한 차원의 멤버가 스크립트에 사용되므로 단일 멤버를 선택한 차원을 선택할 수 없는 경우 이 템플릿을 사용할 수 없습니다.

## 데이터 복사 템플릿 사용

데이터 복사 템플릿을 사용하여 데이터베이스의 한 위치에서 다른 위치로 데이터를 복사합니다.

예를 들어 데이터 복사 템플릿을 사용하면 2014년의 계정 및 엔티티 값을 복사하여 2015년의 예산을 생성할 수 있습니다. 이 경우 2014년의 실제 데이터를 2015년에 대한 예측 데이터로 복사합니다. 데이터 복사 템플릿을 사용하여 "최악"의 예산 시나리오에서 "최상"의 시나리오로 예산 데이터를 복사할 수도 있고, 내용을 변경할 수도 있습니다.

데이터 복사 템플릿을 사용하려면 다음을 수행합니다.

1. 비즈니스 규칙 또는 템플릿을 생성하거나 엽니다.
2. 다음 중 하나를 수행합니다.
  - **그래픽** 규칙 또는 템플릿의 경우 **새 객체**에서 **데이터 복사**를 선택하고 **시작** 및 **종료** 사이의 순서도로 끌어 놓습니다.
  - **스크립트** 규칙 또는 템플릿의 경우 **시스템 템플릿**를 확장하고 **데이터 복사**를 선택한 후 스크립트의 원하는 영역으로 끌어 놓습니다.
3. **데이터 복사 마법사**에서 필요한 정보를 입력합니다.

표 4-2 데이터 복사 마법사 단계





단계	설명
정보	템플릿 함수에 대해 설명합니다.
POV	각 차원에 대해 멤버를 하나 이상 선택하여 복사할 데이터를 정의합니다. 복사 출처 및 위치에 따라 달라지는 차원은 비워 두십시오. 이러한 멤버는 다음 단계에서 지정합니다. 다음 옵션을 사용할 수 있습니다. <ul style="list-style-type: none"> <li>• 값으로 모든 차원을 채울 변수를 선택하려면 <b>사전 정의된 선택 사용</b> 옆의  를 누릅니다.</li> <li>• 각 차원의 멤버 및 함수를 선택하려면 <b>멤버 선택기</b> 옆의  를 누릅니다.</li> <li>• 변수, 멤버 또는 함수를 선택하려면 차원을 선택하고  를 누릅니다.</li> </ul>
복사 출처	데이터를 복사해 올 멤버를 선택합니다. <b>참고:</b> 차원마다 하나의 멤버를 선택합니다. 차원마다 하나의 멤버를 선택하지 않으면 경고 메시지가 표시됩니다.
복사 위치	데이터를 복사할 멤버를 선택합니다.

표 4-2 (계속) 데이터 복사 마법사 단계

단계	설명
옵션	<p><b>선택 사항:</b> 대상 데이터 셀 범위에 적용할 증가 또는 감소 백분율을 입력하십시오. 예를 들어 대상 데이터를 10% 늘리려면 <b>10</b>을 입력하고 대상 데이터를 10% 줄이려면 <b>-10</b>을 입력합니다.</p> <p>변수를 입력하려면 를 누릅니다. 예를 들어 런타임 프롬프트 변수를 정의하고 런타임 시 증가 또는 감소에 대한 값을 입력할 수 있습니다.</p>

## 금액-수량-단가 템플릿 사용

금액-수량-단가 템플릿을 사용하여 금액, 수량, 단가에 대해 입력한 세 멤버 중 하나를 계산합니다.

예를 들어, 제품 수익을 계산하여 수량 및 가격을 알아보려면 **금액 계산**을 선택하고 금액에 제품 수익, 수량(unit)에 수량(quantity), 단가에 가격을 입력합니다. 이렇게 하면 수량과 가격을 곱하여 제품 수익을 계산합니다.

금액-수량-단가 템플릿을 사용하려면 다음을 수행합니다.

1. 비즈니스 규칙 또는 템플릿을 생성하거나 엽니다.
2. 다음 중 하나를 수행합니다.
  - **그래픽** 규칙 또는 템플릿의 경우 **새 객체**에서 **금액-수량-단가**를 선택하고 **시작 및 종료** 사이의 순서도로 끌어 놓습니다.
  - **스크립트** 규칙 또는 템플릿의 경우 **시스템 템플릿**을 확장하고 **금액-수량-단가**를 선택한 후 스크립트의 원하는 영역으로 끌어 놓습니다.
3. **금액-수량-단가 마법사**에서 필요한 정보를 입력합니다.

표 4-3 금액-수량-단가 마법사 단계





단계	설명
정보	템플릿 함수에 대해 설명합니다.
계산 방법 선택	<p>드롭다운 목록에서 다음 옵션 중 하나를 선택하여 사용할 계산 방법을 정의합니다.</p> <ul style="list-style-type: none"> <li>• <b>금액 계산</b> - Amounts = Units * Rates</li> <li>• <b>수량 계산</b> - Units = Amounts / Rates</li> <li>• <b>단가 계산</b> - Rates = Amounts / Units</li> <li>• <b>모두 계산</b> - #Missing이 있는 멤버가 달라지거나 알 수 없는 경우 선택합니다. 이 계산 방법은 #Missing 값을 검색하고 적절한 계산 방법을 적용하여 이러한 알 수 없는 값을 해결합니다.</li> </ul>
POV	<p>할당 프로세스에서 변경되지 않는 멤버를 정의합니다. 멤버를 선택하는 데 다음 옵션을 사용할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• 값으로 모든 차원을 채울 변수를 선택하려면 <b>사전 정의된 선택 사용</b> 옆의 를 누릅니다.</li> <li>• 각 차원의 멤버를 선택하려면 <b>멤버 선택기</b> 옆의 를 누릅니다.</li> <li>• 변수, 멤버 또는 함수를 선택하려면 차원을 선택하고 를 누릅니다.</li> </ul>

표 4-3 (계속) 금액-수량-단가 마법사 단계

단계	설명
금액 계산/수량 계산/단가 계산/모두 계산	<p>이 단계는 선택한 계산 방법에 따라 달라집니다.</p> <ul style="list-style-type: none"> <li>• <b>금액 계산</b>을 선택한 경우 금액에 대한 멤버를 입력하거나 를 눌러 멤버를 선택하고 [다음]을 눌러 수량에 대한 멤버를 입력하고 [다음]을 눌러 단가에 대한 멤버를 입력합니다. 수량 및 단가 단계에서는 POV와 다른 멤버를 입력할 수 있습니다.</li> <li>• <b>수량 계산</b>을 선택한 경우 수량에 대한 멤버를 입력하거나 를 눌러 멤버를 선택하고 [다음]을 눌러 금액에 대한 멤버를 입력하고 [다음]을 눌러 단가에 대한 멤버를 입력합니다. 금액 및 단가 단계에서는 POV와 다른 멤버를 입력할 수 있습니다.</li> <li>• <b>단가 계산</b>을 선택한 경우 단가에 대한 멤버를 입력하거나 를 눌러 멤버를 선택하고 [다음]을 눌러 금액에 대한 멤버를 입력하고 [다음]을 눌러 수량에 대한 멤버를 입력합니다. 금액 및 수량 단계에서는 POV와 다른 멤버를 입력할 수 있습니다.</li> <li>• <b>모두 계산</b>을 선택하면 금액, 수량 및 단가에 대한 멤버를 입력하거나 선택합니다.</li> </ul>

## 할당 - 레벨별 템플릿 사용

레벨별 할당 템플릿을 사용하여 데이터베이스 아웃라인의 한 레벨에서 다른 레벨로 할당할 수 있습니다.

예를 들어 할당 - 레벨별 템플릿을 사용하여 한 레벨에서 다른 레벨로 제품을 할당할 수 있습니다. 제품 아웃라인이 다음과 같다고 가정합니다.

```

T_TP
  P_000
    T_TP1
      P_100
        P_110
      etc
    T_TP2
      P_200
      P_210
    etc
  T_TP3
    P_291
      P_292
  
```

이 예에서는 할당 - 레벨별 템플릿에서 시작 시에는 레벨/세대 3을 사용하고 종료 시에는 레벨/세대 4를 사용하여 상위 멤버(T\_TP1, T\_TP2, T\_TP3)에서 해당 1차 하위 구성요소(레벨 0) 멤버로 할당합니다.

할당 - 레벨별 템플릿을 사용하려면 다음을 수행합니다.

1. 비즈니스 규칙 또는 템플릿을 생성하거나 엽니다.
2. 다음 중 하나를 수행합니다.
  - *그래픽* 규칙 또는 템플릿의 경우 새 객체에서 할당 - 레벨별을 선택하고 시작 및 종료 사이의 순서대로 끌어 놓습니다.
  - *스크립트* 규칙 또는 템플릿의 경우 시스템 템플릿을 확장하고 할당 - 레벨별을 선택한 후 스크립트의 원하는 영역으로 끌어 놓습니다.

3. 할당 - 레벨별 마법사에서 필요한 정보를 입력합니다.

표 4-4 할당 - 레벨별 마법사 단계











단계	설명
POV	<p>나열된 각 차원에 대해 할당 중 변경되지 않도록 하려는 멤버를 하나 이상 선택합니다. 다음 옵션을 사용할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• 값으로 모든 차원을 채울 변수를 선택하려면 <b>사전 정의된 선택 사용</b> 옆의  를 누릅니다.</li> <li>• 각 차원의 멤버 및 함수를 선택하려면 <b>멤버 선택기</b> 옆의  를 누릅니다.</li> <li>• 변수, 멤버 또는 함수를 선택하려면 차원을 선택하고  를 누릅니다.</li> </ul>
소스	<p>나열된 각 차원에 대해 데이터를 할당할 멤버를 선택합니다. 멤버를 선택한 후 할당할 소스 백분율을 입력합니다. 참고: 레벨별 할당에 사용할 차원을 비워 두십시오. 다음 단계에서 이 차원을 선택합니다.</p>
할당 범위	<p>필요한 정보를 입력하여 할당 범위를 정의합니다. 규칙은 시작 레벨로 지정된 레벨에서 끝 레벨로 지정된 레벨까지 선택한 멤버 아래의 모든 데이터를 할당합니다.</p>
대상	<ul style="list-style-type: none"> <li>• 차원이 소스 단계에서 정의되었으나 할당에 사용되지 않은 경우 각 차원에 대해 대상 멤버를 정의합니다. 소스에서 선택한 멤버가 기본적으로 여기에 입력됩니다. 다음 옵션 중 하나를 사용하여 대상 멤버를 정의합니다. <ul style="list-style-type: none"> <li>- 값으로 모든 차원을 채울 변수를 선택하려면 <b>사전 정의된 선택 사용</b> 옆의  를 누릅니다.</li> <li>- 각 차원의 멤버 및 함수를 선택하려면 <b>멤버 선택기</b> 옆의  를 누릅니다.</li> <li>- 변수, 멤버 또는 함수를 선택하려면 차원을 선택하고  를 누릅니다. 최적화를 위해 멤버가 하나만 있는 차원을 선택합니다.</li> </ul> </li> <li>• 위의 단일 멤버를 선택한 소스 차원 선택.</li> </ul>
오프셋	<p>오프셋을 정의합니다. 오프셋 계산을 생성하지 않으려면 이 단계의 테이블을 비워 둡니다. 오프셋을 정의하는 경우 다음을 수행합니다.</p> <ul style="list-style-type: none"> <li>• 오프셋 차원 정의 오프셋 멤버에 할당된 값의 총 금액을 기록하려면 오프셋을 정의할 멤버를 선택합니다.</li> <li>• 할당 차원의 오프셋 멤버 정의 멤버를 입력하거나  를 눌러 멤버를 선택합니다.</li> </ul>

표 4-4 (계속) 할당 - 레벨별 마법사 단계

단계	설명
드라이버	<p>할당 범위의 각 멤버에 적용되는 백분율을 계산하는 데 사용되는 드라이버(기준)를 정의합니다.</p> <ul style="list-style-type: none"> <li>POV와 다른 차원의 멤버만 선택합니다. 다음 옵션 중 하나를 사용하여 멤버를 선택합니다. <ul style="list-style-type: none"> <li>값으로 차원을 채울 변수를 선택하려면 <b>사전 정의된 선택 사용</b> 옆의  를 누릅니다. 멤버를 차원당 하나만 선택합니다.</li> <li>각 차원의 멤버 및 함수를 선택하려면 <b>멤버 선택기</b> 옆의  를 누릅니다.</li> <li>변수, 멤버 또는 함수를 선택하려면 차원을 선택하고  를 누릅니다.</li> </ul> </li> <li>할당 차원에 있는 드라이버의 합계를 업데이트할지 여부를 지정합니다.</li> </ul>
기타 옵션	마법사의 질문에 대답합니다.

## 할당 템플릿 사용

할당 템플릿을 사용하여 균등하게 또는 지정된 드라이버를 기반으로 소스에서 대상으로 값을 할당합니다.

예를 들어 할당 템플릿을 사용하여 총 매출을 기초로 제품 차원의 레벨 0 멤버에 관리 비용을 할당할 수 있습니다.

할당 템플릿을 사용하려면 다음을 수행합니다.

1. 비즈니스 규칙 또는 템플릿을 생성하거나 엽니다.
2. 다음 중 하나를 수행합니다.
  - **그래픽** 규칙 또는 템플릿의 경우 **새 객체**에서 할당을 선택하고 **시작** 및 **종료** 사이의 순서도로 끌어 놓습니다.
  - **스크립트** 규칙 또는 템플릿의 경우 **시스템 템플릿**를 확장하고 할당을 선택한 후 스크립트의 원하는 영역으로 끌어 놓습니다.
3. 할당 마법사에서 필요한 정보를 입력합니다.

표 4-5 할당 마법사 단계




단계	설명
정보	템플릿 함수에 대해 설명합니다.
POV	<p>할당에서 변경되지 않는 멤버를 정의합니다. 차원마다 하나 이상의 멤버를 선택해야 합니다.</p> <p>멤버를 선택하는 데 다음 옵션을 사용할 수 있습니다.</p> <ul style="list-style-type: none"> <li>값으로 모든 차원을 채울 변수를 선택하려면 <b>사전 정의된 선택 사용</b> 옆의  를 누릅니다.</li> <li>각 차원의 멤버를 선택하려면 <b>멤버 선택기</b> 옆의  를 누릅니다.</li> <li>변수, 멤버 또는 함수를 선택하려면 차원을 선택하고  를 누릅니다.</li> </ul>

표 4-5 (계속) 할당 마법사 단계

단계	설명
소스	차원마다 데이터를 할당할 멤버를 선택하고 할당할 소스 백분율을 입력합니다. 예를 들어, 소스 값의 25%를 할당하려면 25를 입력합니다.
할당 범위	데이터를 할당할 차원 및 이 차원의 상위 멤버를 선택합니다. 데이터베이스 아웃라인에서 지정된 상위 멤버 아래의 레벨 0 멤버에 데이터가 할당됩니다. 레벨 0 멤버는 하위 멤버가 없는 아웃라인의 최하위 멤버입니다.
대상 (Destination) - 대상(Target)	소스에 정의된 각 차원의 대상 멤버를 지정하십시오.
대상 - 오프셋	<p><b>선택 사항입니다.</b></p> <ul style="list-style-type: none"> <li>각 차원에 대해 멤버를 입력하여 오프셋 계산을 생성하도록 오프셋을 지정합니다. 오프셋 값은 동일한 소스 값 또는 반대 소스 값으로 계산할 수 있습니다. 예를 들어 할당할 소스에 \$1000가 있는 경우 오프셋을 사용하여 숫자가 중복되지 않도록 할 수 있습니다. 규칙을 실행하는 경우 \$1000가 레벨 0에 할당되고 오프셋에서 지정된 멤버에 -1000을 생성합니다.</li> <li>할당 프로세스를 시작하기 전에 오프셋 데이터를 자율지 여부를 선택합니다.</li> <li>오프셋 값을 할당된 데이터의 반대 부호로 할 것인지 지정합니다.</li> </ul>
제외	<b>선택 사항입니다.</b> 할당 범위에서 제외할 멤버를 선택합니다. 이러한 멤버는 할당된 데이터를 수신하지 않으며 해당 드라이버 값은 드라이버 백분율에서 제외됩니다.
드라이버	"동인을 사용하여 할당"할 것인지 "균등하게 할당"할 것인지 정의합니다. POV와 다른 각 차원에 대해 차원당 멤버를 하나 선택합니다. 비어 있는 차원은 POV 및 규칙에 정의된 상위 멤버 범위에서 상속됩니다.
분산 메소드	동인 단계에서 "균등 할당"을 선택한 경우 데이터 분산 옵션을 선택합니다.
기타 옵션	<p>마법사의 질문에 대답합니다.</p> <p>반올림 방법을 선택하면 선택하는 반올림 방법에 따라 추가 단계가 있습니다. 사용 가능한 반올림 방법은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li><b>반올림 멤버 정의</b> - 이 옵션을 선택하면 다음 단계는 <b>멤버에서 반올림</b>입니다. 이 단계에서는 할당에 사용할 소수 자릿수를 입력하고 할당 반올림 차이를 배치할 멤버를 선택합니다.</li> <li><b>가장 큰 값 사용</b> - 이 옵션을 선택하면 다음 단계는 <b>최대에서 반올림</b>입니다. 이 단계에서는 할당에 사용할 소수 자릿수를 입력합니다.</li> <li><b>가장 작은 값 사용</b> - 이 옵션을 선택하면 다음 단계는 <b>최소에서 반올림</b>입니다. 이 단계에서는 할당에 사용할 소수 자릿수를 입력합니다.</li> <li><b>반올림할 수 없음</b> - 이 옵션을 선택하면 추가 단계가 없습니다.</li> </ul>

## 집계 템플릿 사용

집계 템플릿을 사용하여 지정하는 멤버의 데이터 값을 집계합니다.

집계 템플릿을 사용하여 다음을 수행할 수 있습니다.

- POV에 멤버를 입력하여 집계 제한
- 누락된 값을 집계할지 여부 지정
- 값을 현지 통화로 집계할지 여부 지정(하이브리드 집계에서는 지원되지 않음)

값을 집계할 때 지원되는 계산 명령은 다음과 같습니다.



- SET UPTOLOCAL ON: 데이터를 현지 통화로 집계(하이브리드 집계에서는 지원되지 않음)
- SET CACHE HIGH: 계산기 캐시 값 설정(하이브리드 집계에서는 지원되지 않음)
- SET AGGMISG ON: 데이터베이스에서 누락된 값 집계
- SET FRMLBOTTOMUP ON: 회소 차원의 계산 최적화

예를 들어, 집계 템플릿으로 상위 멤버를 사용하는 밀집 또는 회소 차원을 집계하고 하위 멤버, 1차 하위 멤버 또는 상위 멤버(모두)와 같은 집계 레벨을 선택할 수 있습니다.

집계 템플릿을 사용하려면 다음을 수행합니다.

1. 비즈니스 규칙 또는 템플릿을 생성하거나 엽니다.
2. 다음 중 하나를 수행합니다.
  - *그래픽* 규칙 또는 템플릿의 경우 새 객체에서 집계를 선택하고 시작 및 종료 사이의 순서대로 끌어 놓습니다.
  - *스크립트* 규칙 또는 템플릿의 경우 시스템 템플릿을 확장하고 집계를 선택한 후 스크립트의 원하는 영역으로 끌어 놓습니다.
3. 집계 마법사에서 필요한 정보를 입력합니다.

표 4-6 집계 마법사 단계




단계	설명
정보	템플릿 함수에 대해 설명합니다.
POV	<p>집계를 제한할 차원 멤버를 선택합니다. 멤버를 선택하는 데 다음 옵션을 사용할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• 값으로 모든 차원을 채울 변수를 선택하려면 사전 정의된 선택 사용 옆의  를 누릅니다.</li> <li>• 각 차원의 멤버를 선택하려면 멤버 선택기 옆의  를 누릅니다.</li> <li>• 변수, 멤버 또는 함수를 선택하려면 차원을 선택하고  를 누릅니다.</li> </ul>
전체 밀집 집계	<p>전체적으로 집계하려면 밀집 차원을 최대 두 개까지 선택하고 밀집 차원에 레벨이 0이 아닌 멤버가 저장되어 있는지 여부를 지정합니다. 다음 사항에 유의하십시오.</p> <ul style="list-style-type: none"> <li>• 밀집 차원에 레벨이 0이 아닌 멤버가 저장되어 있는 경우 Calc Dim(밀집 차원)이 스크립트에 생성됩니다.</li> <li>• 밀집 차원에 레벨이 0이 아닌 멤버가 저장되지 않은 경우 해당 밀집 차원에 대해 스크립트가 생성되지 않습니다.</li> <li>• 밀집 차원에서 부분 집계를 수행하려면(함수를 사용하는 상위 멤버(Parent Member) - 하위 멤버, 1차 하위 멤버 또는 상위 멤버(Ancessor)) 이 단계에서 차원을 선택하지 마십시오.</li> </ul>
전체 회소 집계	<p>전체적으로 집계하려면 회소 차원을 최대 세 개까지 선택하고 회소 차원에 계산되어야 하는 멤버 공식이 있는지 여부를 지정합니다. 회소 차원에 멤버 공식이 있는 경우 회소 차원이 계산됩니다. 회소 차원에 멤버 공식이 없는 경우 회소 차원이 집계됩니다.</p>

표 4-6 (계속) 집계 마법사 단계

단계	설명
부분 차원 집계 밀집	<p>부분 집계할 밀집 차원을 최대 두 개까지 선택합니다. 각 밀집 차원에 대해 멤버 및 집계 레벨을 선택합니다. 다음 사항에 유의하십시오.</p> <ul style="list-style-type: none"> <li>전체 밀집 집계 단계에서 사용하지 않는 밀집 차원이 있는 경우 부분 차원 집계 밀집 단계가 표시됩니다. 전체 밀집 집계 탭에서 두 개의 밀집 차원을 사용하고 애플리케이션에 추가 밀집 차원이 없는 경우 부분 차원 집계 밀집 단계가 표시되지 않습니다.</li> <li>회소 차원에서 부분 집계를 수행하려면(함수를 사용하는 상위 멤버(Parent Member) - 하위 멤버, 1차 하위 멤버 또는 상위 멤버(Anccestor)) 이 단계에서 차원을 선택하지 마십시오.</li> </ul>
부분 차원 집계 회소	<p>부분 집계할 회소 차원을 최대 세 개까지 선택합니다. 각 회소 차원에 대해 멤버 및 집계 레벨을 선택합니다. 다음 사항에 유의하십시오.</p> <p>참고:</p> <ul style="list-style-type: none"> <li>부분 집계에 사용 가능한 회소 차원은 회색으로 표시됩니다.</li> <li>전체 회소 집계 단계에서 사용하지 않는 회소 차원이 있는 경우 부분 차원 집계 회소 단계가 표시됩니다. 전체 회소 집계 탭에서 세 개의 회소 차원을 사용하고 애플리케이션에 추가 회소 차원이 없는 경우 부분 차원 집계 회소 단계가 표시되지 않습니다.</li> <li>밀집 차원에서 부분 집계를 수행하려면(함수를 사용하는 상위 멤버(Parent Member) - 하위 멤버, 1차 하위 멤버 또는 상위 멤버(Anccestor)) 이 단계에서 차원을 선택하지 마십시오.</li> </ul>
설정	이 단계에 대한 설정을 지정합니다.

## SET 명령 템플릿 사용

SET 명령 템플릿을 사용하여 계산 스크립트의 성능을 최적화하는 스크립트 명령을 입력할 수 있습니다.

데이터 볼륨, 데이터 처리, 메모리 사용, 스레딩 및 로깅 스크립트 명령을 포함할 수 있습니다.

SET 명령 템플릿을 사용하려면 다음을 수행합니다.

1. 비즈니스 규칙 또는 템플릿을 생성하거나 엽니다.
2. 다음 중 하나를 수행합니다.
  - *그래픽* 규칙 또는 템플릿의 경우 새 객체에서 **SET** 명령을 선택하고 시작 및 종료 사이의 순서대로 끌어 놓습니다.
  - *스크립트* 규칙 또는 템플릿의 경우 시스템 템플릿을 확장하고 **SET** 명령을 선택한 후 스크립트의 원하는 영역으로 끌어 놓습니다.
3. **SET** 명령 마법사에서 필요한 정보를 입력합니다.

마법사에는 다음 단계가 포함되어 있습니다.

- 데이터 볼륨
- 데이터 처리
- 메모리 사용
- 스레딩
- 로깅

각 단계에는 질문이 여러 개 있으며 질문 옆 드롭다운 목록에서 옵션을 선택하여 대답할 수 있습니다. 이 템플릿에서는 옵션을 하나 이상 선택할 수 있습니다.

 주:

Set 명령 마법사의 질문에 모두 대답할 필요는 없습니다. 사용자의 상황에 적용할 수 있는 질문에만 대답하면 됩니다.

 주:

애플리케이션이 Essbase Hybrid에 있는 경우 데이터 볼륨, 데이터 처리, 스테딩 탭만 마법사에 표시됩니다.

## 통화 변환 템플릿 사용

통화 변환 템플릿은 시스템 유형 계정 또는 선택한 계정을 사용하여 계정 값을 보고 통화로 변환합니다.

통화 변환 템플릿은 여러 통화를 선택하여 생성한 Oracle Hyperion Planning 애플리케이션에서 사용됩니다.

통화 변환 템플릿 사용 예는 [통화 변환 템플릿 예](#)를 참조하십시오.

 주:

Planning 애플리케이션이 다중 통화로 생성된 경우에는 통화 변환 템플릿이 시스템 템플릿 목록에 표시되지 않습니다.




통화 변환 템플릿을 사용하려면 다음을 수행합니다.

1. 비즈니스 규칙 또는 템플릿을 생성하거나 엽니다.
2. 다음 중 하나를 수행합니다.
  - *그래픽* 규칙 또는 템플릿의 경우 새 객체에서 통화 변환을 선택하고 시작 및 종료 사이의 순서대로 끌어 놓습니다.
  - *스크립트* 규칙 또는 템플릿의 경우 시스템 템플릿을 확장하고 통화 변환을 선택한 후 스크립트의 원하는 영역으로 끌어 놓습니다.
3. 통화 변환 마법사에서 필요한 정보를 입력합니다.

표 4-7 통화 변환 마법사 단계




단계	설명
정보 - 통화	템플릿의 기능과 템플릿을 사용하는 데 필요한 정보에 대해 설명합니다.

표 4-7 (계속) 통화 변환 마법사 단계

단계	설명
환율 옵션	<p>마법사의 통화에 대한 질문에 대답합니다.</p> <ul style="list-style-type: none"> <li>• <b>통화가 어떤 차원에 있습니까?</b> 통화 차원에 해당하는 차원을 선택합니다.</li> <li>• <b>보고 통화 선택</b> 변환할 보고 통화를 선택합니다.</li> <li>• <b>통화 멤버가 포함된 상위 멤버 선택</b> 애플리케이션에 사용된 모든 통화의 상위 멤버인 통화를 선택합니다(예: "입력 통화").</li> <li>• <b>이 보고 통화에 사용할 환율(통화) 선택</b> 위에서 선택한 보고 통화에 해당하는 통화를 선택합니다. 예를 들어 보고 통화에 <i>EUR 보고</i>를 선택한 경우 <i>EUR</i>를 선택합니다. 여기서 <i>EUR</i>은 데이터를 <i>EUR 보고</i>로 변환하는 데 사용되는 통화입니다.</li> <li>• <b>통화 변환에 계정 유형을 사용하시겠습니까?</b> 통화 변환에 계정 유형을 사용하려면 <b>예</b>를 선택합니다. <b>예</b>를 선택하면 비용 및 수익 계정에서는 평균 환율을 변환에 사용하고 부채 및 자기자본 계정에서는 마감 환율을 변환에 사용합니다. <b>아니요</b>를 선택하면 마법사의 이후 단계에서 평균 및 마감 환율에 사용할 계정을 지정하라는 프롬프트가 표시됩니다.</li> <li>• <b>평균 환율을 계산하시겠습니까?</b> <b>평균 계산 안함</b>을 선택하면 평균 관련 단계가 표시되지 않고 평균 환율에 따라 통화를 계산하기 위한 스크립트가 생성되지 않습니다.</li> <li>• <b>마감 환율을 계산하시겠습니까?</b> 마감 환율을 계산할 것인지 선택합니다. <b>마감 계산 안함</b>을 선택하면 마감 관련 단계가 표시되지 않고 마감 환율에 따라 통화를 계산하기 위한 스크립트가 생성되지 않습니다.</li> </ul>
POV	<p>보고 통화로 변환에 사용할 멤버 선택. 멤버를 선택하는 데 다음 옵션을 사용할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• <b>값으로 모든 차원을 채울 변수를 선택하려면 사전 정의된 선택 사용 옆의</b>  <b>를 누릅니다.</b></li> <li>• <b>각 차원의 멤버를 선택하려면 멤버 선택기 옆의</b>  <b>를 누릅니다.</b></li> <li>• <b>변수, 멤버 또는 함수를 선택하려면 차원을 선택하고</b>  <b>를 누릅니다.</b></li> </ul>
계정 평균	<p>평균 환율을 사용하여 보고 통화로 변환하는 계정을 입력합니다. <b>참고:</b> 이 단계는 <b>환율 옵션</b> 단계의 <b>통화 변환에 계정 유형을 사용하시겠습니까?</b>에서 <b>아니요</b>를 선택한 경우에만 표시됩니다.</p>
계정 마감	<p>마감 환율을 사용하여 보고 통화로 변환하는 계정을 입력합니다. <b>참고:</b> 이 단계는 <b>환율 옵션</b> 단계의 <b>통화 변환에 계정 유형을 사용하시겠습니까?</b>에서 <b>아니요</b>를 선택한 경우에만 표시됩니다.</p>
FX 평균	<p>평균 환율이 포함된 멤버를 선택합니다. <b>참고:</b> POV의 멤버와 다른 멤버만 선택하면 됩니다.</p>
FX 마감	<p>마감 환율이 포함된 멤버를 선택합니다. <b>참고:</b> POV의 멤버와 다른 멤버만 선택하면 됩니다.</p>

## 통화 변환 템플릿 예

1. 기존 차원 외에는 다중 통화를 사용하지 않고 생성된 Oracle Hyperion Planning 애플리케이션에서 "통화"라는 사용자정의 차원을 생성하고 다음 멤버를 추가합니다.
  - 입력 통화
    - USD
    - CAD
    - EUR
    - GBP
    - INR
  - 보고 통화
    - EUR 보고
    - US 보고
    - Can 보고
2. "계정" 차원에서 FX\_Rates에 대해 다음 멤버를 추가합니다.
  - FX\_Average
  - FX\_Ending
3. "엔티티" 차원에서 "회사 가정" 엔티티를 추가합니다.
4. Oracle Smart View for Office 또는 Planning 양식을 사용하여 다음 셀에 환율을 입력합니다.
  - 계정 - FX\_Average 및 FX\_Ending
  - 기간 - 각 월의 환율 입력
  - 연도 - FY12
  - 시나리오 - 현재
  - 버전 - BU Version\_1
  - 엔티티 - 회사 가정
  - 제품 - 000
  - 통화 - 각 통화에 대한 환율 입력
5. 변수에서 다음을 수행합니다.
  - a. "Reporting\_Currency"라는 멤버 변수를 사용하여 통화 차원에 대한 애플리케이션 레벨 런타임 프롬프트를 생성하고 기본값 "EUR 보고"를 입력합니다.
  - b. "Input Currencies"라는 멤버 변수를 사용하여 통화 차원에 대한 애플리케이션 레벨 런타임 프롬프트를 생성하고 기본값 "EUR"을 입력합니다.
  - c. 이 Planning 애플리케이션에서 비즈니스 규칙을 생성하고 통화 변환 시스템 템플릿을 비즈니스 규칙의 순서도로 끌어 놓습니다.  
통화 변환 마법사가 표시됩니다.
6. 통화 변환 마법사에서 다음 단계를 수행합니다.
  - a. 1단계 - 환율 옵션

- 통화가 어떤 차원에 있습니까? 드롭다운을 누르고 위에서 생성한 통화 차원을 선택합니다.
- 보고 통화 선택: 를 누르고 변수, Reporting\_Currency 변수 순으로 선택합니다.  
이 예에서는 각 보고 통화에 대해 규칙이 Planning에서 실행될 수 있도록 보고 통화에 변수를 사용하고 있습니다.
- 통화 멤버가 포함된 상위 멤버 선택: 를 누르고 입력 통화 상위 멤버를 선택합니다.  
통화 멤버에 대해 상위 멤버를 선택해야 하는 이유는 나중에 통화가 추가되는 경우 이 비즈니스 규칙만 Planning에서 재배포하고 다시 실행해야 하기 때문입니다. 템플릿 스크립트에서는 새 통화를 스크립트에 스크립트에 추가하고 보고 통화를 정확하게 계산합니다.
- 이 보고 통화에 사용할 환율(통화) 선택 를 누르고 입력 통화 변수를 선택합니다.  
이 예에서는 각 보고 통화에 대해 규칙이 Planning에서 실행될 수 있도록 보고 통화에 변수를 사용하고 있습니다.
- 통화 변환에 계정 유형을 사용하시겠습니까? 드롭다운을 사용하여 예를 선택합니다.  
[아니요]를 선택하면 이후 단계에서 평균/마감을 사용할 계정을 지정하라는 프롬프트가 표시됩니다.
- 평균 환율을 계산하시겠습니까?  
드롭다운을 사용하여 평균 계산을 선택합니다.
- 마감 환율을 계산하시겠습니까?  
드롭다운을 사용하여 마감 계산을 선택합니다.

#### b. 2단계 - POV

각 차원에 대해 다음을 입력합니다.

- 기간 - 모든 기간에 쓰도록 기간을 비워 둡니다.
- 연도 - "FY12"
- 시나리오 - "현재"
- 버전 - "BU Version\_1"
- 엔티티 - @Relative("South",0)
- 제품 - @Relative("Tennis",0),@Relative("Golf",0)

#### c. 3단계 - FX\_Average

- 평균 환율이 포함된 멤버 선택:
  - 계정 - "FX\_Average"
  - 기간
  - 연도
  - 시나리오
  - 버전
  - 엔티티 - "회사 가정"
  - 제품 - "000"
- 기간 차원을 비워 두면 각 기간에 대해 환율을 사용합니다.

- 연도, 시나리오 및 버전을 비워 두면 이러한 차원에 대해 POV 입력의 멤버를 사용합니다.
- d. 4단계 - FX\_Ending**
- 평균 환율이 포함된 멤버 선택:
    - 계정 - "FX\_Average"
    - 기간
    - 연도
    - 시나리오
    - 버전
    - 엔티티 - "회사 가정"
    - 제품 - "000"
  - 기간 차원을 비워 두면 각 기간에 대해 환율을 사용합니다.
  - 연도, 시나리오 및 버전을 비워 두면 이러한 차원에 대해 POV 입력의 멤버를 사용합니다.
- 7. 비즈니스 규칙을 저장, 검증하고 Planning으로 배포합니다.**
- 이 예의 애플리케이션에는 다음 데이터가 포함됩니다.

**표 4-8 데이터 예 부분 1**

해당 없음	해당 없음	해당 없음	FY12	현재	BU Version_1
해당 없음	해당 없음	Tennessee	Tennessee	Florida	Florida
해당 없음	해당 없음	Jan	Jan	Jan	Jan
해당 없음	해당 없음	G400	G100	G400	G100
총 매출	USD	10750	13450	9500	9699
총 매출	CAD	10450	10000	14450	13000
총 매출	EUR	40000	41450	42450	65000
총 매출	GBP	13250	16750	172000	16300
총 매출	INR	750000	750000	750000	750000
총 매출	US 보고	#Missing	#Missing	#Missing	#Missing
총 매출	CAN 보고	#Missing	#Missing	#Missing	#Missing
총 매출	EUR 보고	#Missing	#Missing	#Missing	#Missing
급여	USD	5000	5000	5000	5000
급여	CAD	4500	4500	4500	4500
급여	EUR	5500	5500	5500	5705
급여	GBP	1200	1200	1200	1200
급여	INR	100000	100,000	100,000	100,000
급여	US 보고	#Missing	#Missing	#Missing	#Missing
급여	CAN 보고	#Missing	#Missing	#Missing	#Missing
급여	EUR 보고	#Missing	#Missing	#Missing	#Missing

이 테이블의 보고 통화에는 값이 없습니다.

- 8. Planning에서 규칙을 실행합니다. 보고 통화 변수에 대해 EUR 보고를 입력하고 입력 통화에 대해 EUR을 입력합니다. 실행을 누릅니다.**

9. Planning의 각 계정에 대해 **EUR 보고**에서 총 계정이 유로로 표시되는지 확인합니다.

표 4-9 데이터 예 부분 2

해당 없음	해당 없음	해당 없음	FY12	현재	BU Version_1
해당 없음	해당 없음	Tennessee	Tennessee	Florida	Florida
해당 없음	해당 없음	Jan	Jan	Jan	Jan
해당 없음	해당 없음	G400	G100	G400	G100
총 매출	USD	10750	13450	9500	9699
총 매출	CAD	10450	10000	14450	13000
총 매출	EUR	40000	41450	42450	65000
총 매출	GBP	13250	16750	172000	16300
총 매출	INR	750000	750000	750000	750000
총 매출	US 보고	#Missing	#Missing	#Missing	#Missing
총 매출	CAN 보고	#Missing	#Missing	#Missing	#Missing
총 매출	EUR 보고	52161391	52168683	52170600	52190796.69
급여	USD	5000	5000	5000	5000
급여	CAD	4500	4500	4500	4500
급여	EUR	5500	5500	5500	5705
급여	GBP	1200	1200	1200	1200
급여	INR	100000	100000	100000	100000
급여	US 보고	#Missing	#Missing	#Missing	#Missing
급여	CAN 보고	#Missing	#Missing	#Missing	#Missing
급여	EUR 보고	6963318	6963318	6963318	6965471.887

10. Planning에서 각 보고 통화에 대해 필요한 횟수만큼 비즈니스 규칙을 실행할 수 있습니다. 매번 변수 입력을 변경하고 변수 및 애플리케이션에서 사용할 수 있는 각 보고 통화에 대해 원하는 보고 통화 및 해당 환율 이름을 입력할 수 있습니다.

이 예에서는 USD로 US 보고에 대해 한 번, CAD로 CAN 보고에 대해 한 번씩 두 번 더 규칙을 실행할 수 있습니다.

## 데이터 익스포트-임포트 템플릿 사용



데이터 익스포트-임포트 템플릿을 사용하여 데이터베이스 아웃라인의 위치에서 데이터를 익스포트하거나 이 위치로 데이터를 임포트할 수 있습니다. 이진 파일을 내보내는 경우 데이터 익스포트 템플릿을 사용하여 가져올 수도 있습니다.

데이터 익스포트-임포트 템플릿을 사용하려면 다음을 수행합니다.

1. 비즈니스 규칙 또는 템플릿을 생성하거나 엽니다.
2. 다음 중 하나를 수행합니다.
  - **그래픽** 규칙 또는 템플릿의 경우 새 객체에서 데이터 익스포트-임포트를 선택하고 시작 및 종료 사이의 순서대로 끌어 놓습니다.
  - **스크립트** 규칙 또는 템플릿의 경우 시스템 템플릿을 확장하고 데이터 익스포트-임포트를 선택한 후 스크립트의 원하는 영역으로 끌어 놓습니다.
3. 데이터 익스포트-임포트 마법사에서 필요한 정보를 입력합니다.



표 4-10 데이터 익스포트-임포트 마법사 단계

단계	설명
모드	데이터를 익스포트할지 임포트할지 지정합니다.
데이터 범위 익스포트	<ul style="list-style-type: none"> <li>익스포트할 데이터의 범위를 지정합니다. 데이터 범위를 지정하는 데 다음 옵션을 사용할 수 있습니다. <ul style="list-style-type: none"> <li>사전 정의된 선택 사용에서 사전 정의된 선택을 선택하여, 나열된 차원을 값으로 채웁니다.</li> <li>각 차원의 멤버를 선택하려면 <b>멤버 선택기</b> 옆의  를 누릅니다.</li> <li>변수, 멤버 또는 함수를 선택하려면 차원을 선택하고  를 누릅니다.</li> </ul> </li> <li>데이터를 익스포트할 파일 유형을 선택합니다. <ul style="list-style-type: none"> <li>플랫 파일</li> <li>데이터베이스 테이블</li> <li>이진 파일</li> </ul> </li> </ul> <p>참고: 파일을 익스포트한 후 임포트하려면 이진 파일이어야 합니다.</p>
출력 - 파일	<ul style="list-style-type: none"> <li>플랫 파일로 내보내는 경우: <ol style="list-style-type: none"> <li>파일 이름과 경로를 따옴표로 묶어서 입력합니다.</li> <li>열 구분자를 선택합니다.</li> <li><b>선택 사항:</b> 텍스트 문자열을 따옴표로 묶어서 입력하여 누락된 데이터 값을 나타내거나 <b>작업</b> 아이콘을 사용하여 문자열을 선택합니다.</li> </ol> </li> <li>데이터베이스 테이블로 내보내는 경우: <ol style="list-style-type: none"> <li>데이터 소스 이름을 따옴표로 묶어서 입력합니다.</li> <li>데이터 소스 테이블 이름을 따옴표로 묶어서 입력합니다.</li> <li>데이터 소스 로그인 이름과 비밀번호를 따옴표로 묶어서 입력합니다.</li> </ol> </li> <li>이진 파일로 내보내는 경우 파일 이름과 경로를 따옴표로 묶어서 입력합니다.</li> </ul>
처리	이 단계에 대한 설정을 지정합니다.

## 템플릿 순서도 표시

시스템 템플릿에서 생성된 스크립트를 검토하는 경우 템플릿 순서도를 보는 것이 유용한 경우가 있습니다.

템플릿 순서도에 표시되는 내용은 다음과 같습니다.

- 템플릿에서 생성되는 스크립트의 플로우
- 디자인 타임 프롬프트가 템플릿 단계에서 선택한 내용으로 바꾸는 텍스트

템플릿 플로우를 표시하려면 다음을 수행합니다.

- 시스템 뷰에서 시스템 템플릿이 포함된 비즈니스 규칙 또는 사용자정의 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
- 아직 템플릿에 정보를 채우지 않은 경우 정보를 모두 채웁니다.
- 규칙 디자이너 순서도에서 시스템 템플릿을 마우스 오른쪽 버튼으로 누르고 **템플릿 순서도 표시**를 선택합니다.

템플리트 순서도가 템플리트 디자이너에 읽기 전용 모드로 표시됩니다. 순서도에서 각 구성요소를 눌러 해당 구성요소와 연관된 스크립트 및 관련 DTP 단계에서 입력된 값을 표시할 수 있습니다.

순서도의 구성요소가 회색으로 표시되는 것은 템플리트에서 데이터를 입력할 때 선택한 내용에 따른 스크립트 생성에 해당 구성요소가 포함되지 않음을 나타냅니다.

특히 조건 구성요소가 있고 조건의 true 및 false 측면 모두에 대한 스크립트가 있는 경우 템플리트 순서도를 보는 것이 유용합니다. 템플리트에서 스크립트를 생성하는 데 사용되는 경로는 굵게 표시되고 다른 경로는 회색으로 표시됩니다.

## 시스템 템플리트를 사용자정의 템플리트로 저장

시스템 템플리트의 콘텐츠를 사용자정의할 수 있습니다.

시스템 템플리트를 편집할 수는 없지만 시스템 템플리트에서 [다른 이름으로 저장]을 사용하여 편집할 수 있는 사용자정의 템플리트를 생성할 수 있습니다. 원래 시스템 템플리트는 변경되지 않습니다.

### 주:

Oracle Essbase 시스템 템플리트에서 [다른 이름으로 저장]을 수행하여 사용자정의 템플리트로 저장하면 새 사용자정의 템플리트에 애플리케이션이라고 하는 디자인 타임 프롬프트가 나타납니다. 그러나 Essbase에 대해 새 사용자정의 템플리트를 생성하는 경우 템플리트에 애플리케이션 디자인 타임 프롬프트가 나타나지 않습니다. 사용자정의 템플리트로 저장하는 시스템 템플리트의 애플리케이션 디자인 타임 프롬프트는 무시하십시오.

시스템 템플리트를 사용자정의 템플리트로 저장하려면 다음을 수행합니다.

1. **시스템 뷰**에서 시스템 템플리트가 포함된 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **새 객체**나 **기존 객체**에서 또는 규칙 디자이너 순서도나 스크립트에서 시스템 템플리트를 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
3. 시스템 템플리트가 **템플리트 디자이너**에서 열리면 읽기 전용으로 열립니다. **확인**을 누르고 **작업**, **다른 이름으로 저장** 순으로 선택합니다.
4. **다른 이름으로 저장** 대화상자에서 템플리트의 새 이름을 입력하고 애플리케이션 유형, 애플리케이션 및 큐브를 선택한 후 **확인**을 누릅니다.

선택한 애플리케이션과 큐브의 **템플리트** 노드에 새 템플리트가 표시됩니다. 이 템플리트를 열어 사용자정의할 수 있습니다.


### 주:

새 템플리트를 표시하려면 **시스템 뷰**에서 애플리케이션 목록을 새로고쳐야 할 수 있습니다. **템플리트** 노드를 마우스 오른쪽 버튼으로 누르고 **새로고침**을 선택합니다.

## 비즈니스 규칙에서 시스템 템플릿 제거

비즈니스 규칙에서 시스템 템플릿을 제거할 수 있습니다.

시스템 템플릿을 제거하려면 다음을 수행합니다.

1. **시스템 뷰**에서 시스템 템플릿을 제거할 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 그래픽 규칙의 경우 규칙 디자이너의 순서도에서 시스템 템플릿을 마우스 오른쪽 버튼으로 누르고 **제거**를 선택합니다.
3. 스크립트 규칙의 경우 시스템 템플릿이 포함된 라인 전체를 강조 표시하고 마우스 오른쪽 버튼을 눌러 **삭제**를 선택합니다.
4. 를 누릅니다.

# 5

## 사용자정의 템플리트 작업

### 참조:

- [사용자정의 템플리트 정보](#)  
사용자정의 템플리트는 관리자가 비즈니스 규칙 및 다른 템플리트에서 사용하기 위해 디자인합니다.
- [사용자정의 템플리트 생성](#)  
그래픽 사용자정의 템플리트 및 스크립트 사용자정의 템플리트를 생성할 수 있습니다.
- [사용자정의 템플리트에 대한 디자인 타임 프롬프트 생성](#)  
사용자가 템플리트를 사용하여 비즈니스 규칙을 디자인할 때 정확한 정보를 입력하라는 프롬프트가 표시되도록 그래픽 사용자정의 템플리트의 디자인 타임 프롬프트를 입력할 수 있습니다.
- [사용자정의 템플리트 열기](#)  
사용자정의 템플리트는 시스템 뷰, 배포 뷰 및 사용자정의 뷰에서 열 수 있습니다.
- [사용자정의 템플리트 새로고침](#)  
사용자정의 템플리트를 생성한 후 템플리트 노드에 새 템플리트를 표시하려면 시스템 뷰에서 애플리케이션 목록을 새로고쳐야 합니다.
- [사용자정의 템플리트에 대한 사용 표시](#)  
사용자정의 템플리트를 사용하는 비즈니스 규칙 목록을 표시할 수 있습니다.
- [사용자정의 템플리트 복사 및 붙여넣기](#)  
사용자정의 템플리트를 복사하여 다른 비즈니스 규칙에 붙여넣을 수 있습니다.
- [사용자정의 템플리트 삭제](#)  
시스템 뷰, 사용자정의 뷰 및 배포 뷰에서 사용자정의 템플리트를 삭제할 수 있습니다.
- [그래픽 사용자정의 템플리트에서 텍스트 찾기 및 바꾸기](#)  
사용자정의 템플리트에서 텍스트 문자열을 검색할 수 있습니다. 텍스트 문자열의 모든 인스턴스를 바꿀 수도 있습니다.

## 사용자정의 템플리트 정보

사용자정의 템플리트는 관리자가 비즈니스 규칙 및 다른 템플리트에서 사용하기 위해 디자인합니다.

### 주:

사용자정의 템플리트는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 애플리케이션에서 지원됩니다.

사용자정의 템플리트에 액세스할 수 있는 위치는 다음과 같습니다.

- 임의 뷰의 계획 유형 **템플리트** 노드
- 기존 객체의 규칙 디자이너

사용자정의 템플릿의 유형은 다음 두 가지입니다.

- **그래픽** 사용자정의 템플릿에는 공식, 스크립트, 조건, 멤버 블록, 멤버 범위, 고정 루프, 메타데이터 루프 및 DTP 지정 구성요소가 포함될 수 있습니다.
- **스크립트** 사용자정의 템플릿에는 스크립트는 포함될 수 있으나 구성요소는 포함될 수 없습니다.

또한 그래픽 및 스크립트 사용자정의 템플릿 모두에 기존 규칙, 공식 및 스크립트(공유 구성요소), 기타 사용자정의 템플릿 및 시스템 템플릿이 포함될 수 있습니다.

규칙 또는 다른 사용자정의 템플릿에 그래픽 및 스크립트 템플릿이 모두 포함될 수 있습니다.

그래픽 및 스크립트 템플릿 간 차이:

- 그래픽 템플릿에는 DTP 지정 구성요소와 메타데이터 루프 구성요소, 구성요소 내에서 DTP 조건을 사용하는 기능이 있습니다.
- 스크립트 템플릿은 Planning에서 배포하고 실행할 수 있습니다.

## 사용자정의 템플릿 생성

그래픽 사용자정의 템플릿 및 스크립트 사용자정의 템플릿을 생성할 수 있습니다.

참조:

- [그래픽 사용자정의 템플릿 생성](#)
- [스크립트 사용자정의 템플릿 생성](#)

## 그래픽 사용자정의 템플릿 생성

그래픽 사용자정의 템플릿을 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 **템플릿**을 마우스 오른쪽 버튼으로 누르고 **새로 작성**을 선택합니다.
2. **새 사용자정의 템플릿**에서 필요한 정보를 입력하고 **확인**을 누릅니다.

새 사용자정의 템플릿 대화상자가 자동으로 시스템 뷰에서 작업 중인 Oracle Hyperion Planning 애플리케이션 유형, 애플리케이션 및 계획 유형으로 채워집니다.

3. **템플릿 디자이너**에서 새 객체 및 기존 객체를 템플릿 순서도에 추가합니다.

객체를 추가하려면 **새 객체** 또는 **기존 객체**에서 **시작** 및 **종료** 사이의 순서도로 객체를 끌어 놓습니다.

표 5-1 그래픽 사용자정의 템플릿에서 삽입할 새 객체

객체	참조
공식	<a href="#">공식 구성요소</a>
스크립트	<a href="#">스크립트 구성요소</a>
조건	<a href="#">조건 구성요소</a>
멤버 블록	<a href="#">멤버 블록 구성요소</a>
멤버 범위	<a href="#">멤버 범위 구성요소</a>
고정 루프	<a href="#">고정 루프 구성요소</a>
메타데이터 루프	<a href="#">메타데이터 루프 구성요소</a>

표 5-1 (계속) 그래픽 사용자정의 템플릿에서 삽입할 새 객체

객체	참조
DTP 지정	DTP 지정 구성요소
데이터 범위 Oracle Hyperion Financial Management 사용자만 해당	데이터 범위 구성요소(Financial Management 사용자만 해당).

표 5-2 그래픽 사용자정의 템플릿에서 삽입할 기존 객체

객체	참조
규칙	비즈니스 규칙 디자인
스크립트	스크립트 및 공식 구성요소 공유
공식	스크립트 및 공식 구성요소 공유
템플릿	사용자정의 템플릿 작업

4. 등록정보에서 템플릿에 대한 등록정보를 입력합니다.

템플릿에 구성요소를 추가하고 순서도에서 구성요소 간에 이동하게 되면 등록정보가 변경됩니다. 특정 구성요소에 대한 등록정보를 입력하려면 순서도에서 구성요소를 선택합니다.

5. 다음 유형의 디자인 타임 프롬프트를 입력하려면 **디자인 타임 프롬프트** 탭을 사용합니다.

- 템플릿을 사용하는 경우 정보를 입력하도록 프롬프트를 표시할 수 있는 디자인 타임 프롬프트
- DTP 지정 구성요소, 사용 조건 및 검증 조건에서 사용할 수 있는, 프롬프트할 수 없는 디자인 타임 프롬프트

사용자정의 템플릿에 대한 디자인 타임 프롬프트 생성을 참조하십시오.

사용자정의 템플릿을 열면 기본적으로 이러한 시스템 디자인 타임 프롬프트가 포함되어 있거나 원하는 경우 시스템 디자인 타임 프롬프트를 템플릿에 추가할 수 있습니다.

- **애플리케이션 유형** - Classic(일반) 애플리케이션인지 Oracle Project Financial Planning(PFP) 애플리케이션인지 결정하는 데 사용됩니다.
- **애플리케이션** - 애플리케이션이 단일, 다중 또는 단순 통화인지 결정하는 데 사용됩니다.
- **UpperPOV** - 글로벌 범위에 멤버가 있는지 또는 템플릿이 멤버 범위에 삽입되었는지 확인하는 데 사용됩니다. **UpperPOV 디자인 타임 프롬프트를 사용하는 그래픽 템플릿을 생성하는 예**를 참조하십시오.

**Oracle Essbase 사용자만 해당:** 사용자정의 템플릿을 열면 기본적으로 비밀번호 디자인 타임 프롬프트가 포함되어 있습니다. 일부 Essbase 기능에는 입력 비밀번호의 매개변수가 필요합니다.

6. 템플릿을 저장하려면  를 누릅니다.

 **팁:**



템플릿에서 구성요소를 편집할 때 구성요소 아이콘의 크기와 순서도에 표시되는 세부정보의 양을 늘리거나 줄일 수 있습니다. 이렇게 하려면



을 사용하여 순서도 내에서 확대 및 축소합니다.

## UpperPOV 디자인 타임 프롬프트를 사용하는 그래픽 템플릿을 생성하는 예

다음 단계에서는 UpperPOV 디자인 타임 프롬프트를 사용하는 그래픽 템플릿을 생성하는 방법을 보여줍니다. 이 예의 단계에서는 샘플 Oracle Hyperion Planning 애플리케이션을 사용합니다.

1. 시스템 뷰에서 **템플릿**을 마우스 오른쪽 버튼으로 누르고 **새로 작성**을 선택합니다.
2. 새 사용자정의 템플릿에서 필요한 정보를 입력하고 **확인**을 누릅니다.
3. 디자인 타임 프롬프트 탭에서  을 누르고 **끝에 행 삽입**을 선택합니다.
4. 프롬프트 생성 대화상자에서 다음을 수행합니다.
  - a. 등록정보 탭에서 다음 정보를 입력합니다.
    - 이름 - "MR1"을 입력합니다.
    - 유형 - "멤버 범위"를 선택합니다.
    - 프롬프트?를 선택합니다.
    - DTP 텍스트 - 멤버 범위를 입력합니다.
  - b. 종속성 탭에서 **UpperPOV** 드롭다운을 누르고 **제외**를 선택합니다.
  - c. **확인**을 누릅니다.
5. 템플릿 디자이너 순서도에서 **멤버 범위** 구성요소로 끕니다.
6. 멤버 범위 탭에서  를 누릅니다.
7. DTP 선택기 대화상자에서 4단계에서 생성된 "멤버 범위" 디자인 타임 프롬프트를 선택하고 **확인**을 누른 다음 **동적으로 변수 링크**가 선택되어 있는지 확인합니다.
8. 템플릿에서 단계를 생성하고 이 단계에 "MR1"이라는 새 DTP를 추가합니다.
9. 템플릿을 저장합니다.
10. 규칙에서 **멤버 범위**를 끌고 "엔티티" 및 "제품"의 멤버를 입력합니다.
11. 새 그래픽 템플릿을 규칙의 멤버 범위로 끕니다.

**멤버 범위** 유형 DTP 단계가 표시되고 "엔티티" 및 "제품" 차원이 표시되지 않는 것을 확인합니다. UpperPOV를 제외하도록 **멤버 범위** 유형의 DTP를 만들었고 UpperPOV(규칙으로 끈 멤버 범위)에서 "엔티티" 및 "제품" 차원을 사용하기 때문입니다.

규칙에서 DTP에 대한 종속성을 **멤버 범위**에서 **포함** 유형으로 변경하는 경우 "엔티티" 차원만 표시되고 템플릿의 단계에 "제품" 차원만 표시됩니다.

DTP 조건의 그래픽 템플릿에서 UpperPOV 시스템 DTP를 사용하여 UpperPOV에 멤버가 있는지(비어 있지 않음), 멤버가 없는지(비어 있음) 그리고 템플릿을 끈 멤버 범위 구성요소가 있는지(사용할 수 있음), 그렇지 않은지(사용할 수 없음) 확인할 수도 있습니다.

## 스크립트 사용자정의 템플릿 생성

스크립트를 사용하여 사용자정의 템플릿을 생성하는 경우 다음을 수행할 수 있습니다.

- Oracle Hyperion Planning에 스크립트 템플릿 배포
- 런타임 시 데이터를 입력할 수 있는 단계가 표시되는 Planning에서 템플릿 실행
- 런타임 함수 및 디자인 타임 프롬프트 사용

스크립트를 사용하여 사용자정의 템플릿을 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 **템플릿**을 마우스 오른쪽 버튼으로 누르고 **새로 작성**을 선택합니다.
2. **새 사용자정의 템플릿** 대화상자에서 필요한 정보를 입력하고 **확인**을 누릅니다.

대화상자가 시스템 뷰에서 작업 중인 Planning 애플리케이션 유형, 애플리케이션 및 큐브로 자동으로 채워집니다.

3. 템플릿이 열리면 **디자이너** 옆의 드롭다운을 누르고 **스크립트 편집**을 선택합니다.

템플릿이 그래픽에서 스크립트로 변환됩니다.


4. **스크립트 유형** 드롭다운에서 **계산 스크립트**를 선택합니다.

**스크립트 유형** 드롭다운이 표시되지 않으면 템플릿 디자이너 오른쪽에 있는 **>>**를 누릅니다.

5. 등록정보에서 템플릿에 대한 등록정보를 입력합니다.

6. **템플릿 디자이너**에서 스크립트를 입력합니다.

7. Planning에서 템플릿을 실행할 때 정보를 입력하도록 프롬프트를 표시하는 디자인 타임 프롬프트를 입력하려면 **디자인 타임 프롬프트** 탭을 사용합니다. [사용자정의 템플릿에 대한 디자인 타임 프롬프트 생성](#)을 참조하십시오.

8. 템플릿을 저장하려면 를 누릅니다.

9. Planning에 템플릿을 배포합니다.

10. Planning을 열고 템플릿을 실행합니다.

## 사용자정의 템플릿에 대한 디자인 타임 프롬프트 생성

사용자가 템플릿을 사용하여 비즈니스 규칙을 디자인할 때 정확한 정보를 입력하라는 프롬프트가 표시되도록 그래픽 사용자정의 템플릿의 디자인 타임 프롬프트를 입력할 수 있습니다.

### 주:

사용자정의 템플릿 및 디자인 타임 프롬프트는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 애플리케이션에서 지원됩니다.

다음 유형의 디자인 타임 프롬프트를 생성할 수 있습니다.

- 속성 DTP
- 부울 DTP
- 횡차원 DTP




- 조건 DTP
- 차원 DTP
- 차원 DTP
- 멤버 DTP
- 멤버 DTP
- 멤버 범위 DTP
- 숫자 DTP
- 비밀번호 DTP
- 제한된 목록 DTP
- 구분자 DTP
- 문자열 DTP


## 속성 DTP

사용자정의 템플릿이 속한 애플리케이션에서 속성을 정의합니다.

예를 들어 제품 크기를 입력하라는 디자인 타임 프롬프트를 생성한다고 가정합니다. 이 경우 "크기" 차원을 선택할 수 있습니다. 필요한 경우 기본값을 입력할 수도 있습니다(예: "대형"). **프롬프트?**를 선택하는 경우 템플릿 사용 시 속성 멤버(이 예에서는 차원 크기)를 입력하라는 메시지가 사용자에게 표시됩니다.

속성 디자인 타임 프롬프트를 생성하려면 다음을 수행합니다.


1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
2. **디자인 타임 프롬프트** 탭에서  를 누릅니다.
3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - 이름을 입력하고 **유형**으로 속성을 선택합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.
    - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.
    - **읽기 전용** - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
  - **DTP 텍스트**에 입력하라는 메시지에서 사용자에게 표시할 텍스트를 입력합니다. **프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.
  - **선택 사항**입니다. **기본값**에서 DTP의 기본값을 지정합니다.

기본값을 지정하려면  을 누르고 **멤버**를 선택한 다음 **멤버 선택기**에서 멤버를 선택합니다.

## 부울 DTP

"True" 또는 "False" 유형 DTP입니다.


부울 디자인 타임 프롬프트를 생성하려면 다음을 수행합니다.






1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
2. **디자인 타임 프롬프트** 탭에서  를 누릅니다.
3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - 이름을 입력하고 **유형**으로 **부울**을 선택합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.
    - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.
    - **읽기 전용** - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
  - **DTP 텍스트**에 입력하라는 메시지에서 사용자에게 표시할 텍스트를 입력합니다. **프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.
  - **선택 사항**입니다. 기본값에서 **True** 또는 **False**를 선택합니다.

## 횡차원 DTP

템플릿이 속한 애플리케이션에서 멤버의 횡차원 구문(->) 범위를 정의합니다.

사용자는 횡차원 DTP를 통해 각 차원에서 멤버를 하나 선택할 수 있습니다. 예를 들어 횡차원 디자인 시간 프롬프트를 공식 소스에 사용할 수 있습니다.

1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
2. **디자인 타임 프롬프트** 탭에서  를 누릅니다.
3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - 이름을 입력하고 **유형**으로 **횡차원**을 선택합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.
    - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.


- 읽기 전용 - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
  - **DTP 텍스트에 입력하라는 메시지**에서 사용자에게 표시할 텍스트를 입력합니다. **프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.
  - **선택 사항입니다. 기본값**에서 DTP의 기본값을 지정합니다. 기본값을 지정하려면 다음 중 하나를 수행합니다.
    - 를 누르고 각 차원 옆의 를 누른 후 **DTP**, **멤버** 또는 **함수**를 선택합니다.
    - 을 눌러 DTP를 기본값으로 선택합니다.
5. **선택사항: 종속성 버튼**()을 누르고 옵션을 선택합니다.
- **제외** - 디자인 타임 프롬프트(예: DTP2)가 다른 디자인 타임 프롬프트(예: DTP1)를 제외하도록 합니다. 사용자에게 DTP2에 대한 프롬프트가 표시될 경우 DTP1에서 사용되지 않는 차원만 표시됩니다.
  - **포함** - 디자인 타임 프롬프트(예: DTP4)가 다른 디자인 타임 프롬프트(예: DTP3)를 포함하도록 합니다. 사용자에게 DTP4에 대한 프롬프트가 표시될 경우 DTP3에서 사용되는 차원만 표시됩니다.
  - **없음** - 디자인 타임 프롬프트에 대해 종속성이 정의되지 않았습니다.
6. **선택사항: 차원 한도 선택 버튼**()을 누르고 다음을 수행합니다.
- **밀도에서** 다음 옵션 중 하나를 선택합니다.
    - **희소 표시** - DTP에 대해 희소 차원만 표시합니다.
    - **밀집 표시** - DTP에 대해 밀집 차원만 표시합니다.
    - **둘 다** - DTP에 대해 밀집 및 희소 차원을 둘 다 표시합니다.
  - **차원 유형에서** 차원 유형을 선택하여 DTP에 선택한 차원 유형만 표시되도록 제한합니다.
 

예를 들어 사용자가 *계정*, *엔티티* 및 *연도*에 대한 멤버만 입력하게 하려는 경우 **차원 유형**에서 해당 차원을 선택합니다. 입력하라는 메시지가 나타날 때 *계정*, *엔티티* 및 *연도* 유형의 차원만 표시됩니다.

## 조건 DTP

조건을 정의합니다.

조건 디자인 타임 프롬프트를 생성하려면 다음을 수행합니다.

1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
2. **디자인 타임 프롬프트** 탭에서 를 누릅니다.
3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - 이름을 입력하고 **유형**으로 **조건**을 선택합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트



템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.


- **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.
- **읽기 전용** - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
- **DTP 텍스트**에 입력하라는 메시지에서 사용자에게 표시할 텍스트를 입력합니다.  
**프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.

## 차원 DTP

템플릿이 속한 애플리케이션에서 차원을 정의합니다.

차원 디자인 타임 프롬프트를 생성하려면 다음을 수행합니다.

1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
2. **디자인 타임 프롬프트** 탭에서  를 누릅니다.
3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - 이름을 입력하고 **유형**으로 **차원**을 선택합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에 DTP를 추가하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 이 DTP가 표시됩니다.
    - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.
    - **읽기 전용** - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
  - **DTP 텍스트**에 입력하라는 메시지에서 사용자에게 표시할 텍스트를 입력합니다.  
**프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.
  - **선택 사항입니다.** 기본값에서 드롭다운을 누르고 템플릿이 속한 애플리케이션에서 차원을 선택합니다.
5. **선택사항: 종속성 버튼**()을 누르고 옵션을 선택합니다.
  - **제외** - 디자인 타임 프롬프트(예: DTP2)가 다른 디자인 타임 프롬프트(예: DTP1)를 제외하도록 합니다. 사용자에게 DTP2에 대한 프롬프트가 표시될 경우 DTP1에서 사용되지 않는 차원만 표시됩니다.
  - **포함** - 디자인 타임 프롬프트(예: DTP4)가 다른 디자인 타임 프롬프트(예: DTP3)를 포함하도록 합니다. 사용자에게 DTP4에 대한 프롬프트가 표시될 경우 DTP3에서 사용되는 차원만 표시됩니다.
  - **없음** - 디자인 타임 프롬프트에 대해 종속성이 정의되지 않았습니다.

차원 DTP는 멤버 범위, 차원, 차원 또는 횡차원 DTP에 종속되어 있을 수 있습니다.
6. **선택사항: 차원 한도 선택 버튼**()을 누르고 다음을 수행합니다.
  - **밀도**에서 다음 옵션 중 하나를 선택합니다.


- **희소 표시** - DTP에 대해 희소 차원만 표시합니다.
- **밀집 표시** - DTP에 대해 밀집 차원만 표시합니다.
- **둘 다** - DTP에 대해 밀집 및 희소 차원을 둘 다 표시합니다.
- **차원 유형**에서 차원 유형을 선택하여 DTP에 선택한 차원 유형만 표시되도록 제한합니다.



예를 들어 사용자가 *계정*, *엔티티* 및 *연도*에 대한 멤버만 입력하게 하려는 경우 **차원 유형**에서 해당 차원을 선택합니다. 입력하라는 메시지가 나타날 때 *계정*, *엔티티* 및 *연도* 유형의 차원만 표시됩니다.

## 차원 DTP


템플릿이 속한 애플리케이션에서 차원을 정의합니다.

차원 디자인 타임 프롬프트를 생성하려면 다음을 수행합니다.

1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
2. **디자인 타임 프롬프트** 탭에서  를 누릅니다.
3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - **이름**을 입력하고 **유형**으로 **차원**을 선택합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.
    - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.
    - **읽기 전용** - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
  - **DTP 텍스트**에 입력하라는 메시지에서 사용자에게 표시할 텍스트를 입력합니다. **프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.
  - **선택 사항**입니다. **기본값**에서 DTP의 기본값을 지정합니다.
 

기본값을 지정하려면  을 누르고 **DTP** 또는 **차원**을 선택한 다음 DTP 또는 차원을 선택합니다.
5. **선택사항: 종속성 버튼**()을 누르고 옵션을 선택합니다.
  - **제외** - 디자인 타임 프롬프트(예: DTP2)가 다른 디자인 타임 프롬프트(예: DTP1)를 제외하도록 합니다. 사용자에게 DTP2에 대한 프롬프트가 표시될 경우 DTP1에서 사용되지 않는 차원만 표시됩니다.
  - **포함** - 디자인 타임 프롬프트(예: DTP4)가 다른 디자인 타임 프롬프트(예: DTP3)를 포함하도록 합니다. 사용자에게 DTP4에 대한 프롬프트가 표시될 경우 DTP3에서 사용되는 차원만 표시됩니다.
  - **없음** - 디자인 타임 프롬프트에 대해 종속성이 정의되지 않았습니다.



차원 DTP는 멤버 범위, 차원, 차원 또는 횡차원 DTP에 종속되어 있을 수 있습니다.

6. **선택사항: 차원 한도 선택 버튼**()을 누르고 다음을 수행합니다.
- 밀도에서 다음 옵션 중 하나를 선택합니다.
    - **희소 표시** - DTP에 대해 희소 차원만 표시합니다.
    - **밀집 표시** - DTP에 대해 밀집 차원만 표시합니다.
    - **둘 다** - DTP에 대해 밀집 및 희소 차원을 둘 다 표시합니다.
  - **차원 유형**에서 차원 유형을 선택하여 DTP에 선택한 차원 유형만 표시되도록 제한합니다.  
 예를 들어 사용자가 *계정*, *엔티티* 및 *연도*에 대한 멤버만 입력하게 하려는 경우 **차원 유형**에서 해당 차원을 선택합니다. 입력하라는 메시지가 나타날 때 *계정*, *엔티티* 및 *연도* 유형의 차원만 표시됩니다.



## 멤버 DTP

애플리케이션의 차원에서 멤버를 정의합니다.

멤버 디자인 시간 프롬프트를 생성하려면 다음을 수행합니다.

1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
  2. **디자인 타임 프롬프트** 탭에서 를 누릅니다.
  3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
  4. 다음 정보를 입력합니다.
    - 이름을 입력하고 **유형**으로 **멤버**를 선택합니다.
    - 원하는 옵션을 선택합니다.
      - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.
      - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.
      - **읽기 전용** - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
    - **DTP 텍스트**에 입력하라는 메시지에서 사용자에게 표시할 텍스트를 입력합니다.  
**프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.
    - **선택 사항입니다. 기본값**에서 DTP의 기본값을 지정합니다.  
 기본값을 지정하려면 을 누르고 **멤버**를 선택한 다음 **멤버 선택기**에서 멤버를 선택합니다. **차원** 드롭다운에서 선택한 속성 차원이 표시됩니다. 이 차원을 확장하여 사용 가능한 멤버를 표시합니다.
    - **스크립트 템플릿만 해당. 제한**에서 DTP, 멤버 또는 사용자가 선택할 수 있는 멤버 세트를 반환하는 함수를 입력합니다.  
 예를 들어 메시지가 표시될 때 상위 "Revenues" 아래의 레벨 0 계층만 표시하도록 DTP를 제한하려면 **차원**으로 "계정"을 선택한 후 **한도**에 "ILvL0Descendants(Revenues)"를 입력합니다.
- 참고:** 한도에 대한 함수를 사용하는 경우 Oracle Essbase 함수 대신 Planning 함수를 사용하는 것이 좋습니다. Essbase 함수 평가에 동적 멤버가 포함된 경우와 같이 Essbase

함수는 필요한 멤버를 반환하지 않는 경우가 있습니다. 멤버 디자인 시간 프롬프트의 한도에 Essbase 함수 "@Relative("Mbr Name", 0)" 대신 "lLvI0Descendants("Mbr Name")"를 사용하십시오.


5. **선택사항: 종속성 버튼** ()을 누르고 옵션을 선택합니다.
  - **제외** - 디자인 타임 프롬프트(예: DTP2)가 다른 디자인 타임 프롬프트(예: DTP1)를 제외하도록 합니다. 사용자에게 DTP2에 대한 프롬프트가 표시될 경우 DTP1에서 사용되지 않는 차원만 표시됩니다.
  - **포함** - 디자인 타임 프롬프트(예: DTP4)가 다른 디자인 타임 프롬프트(예: DTP3)를 포함하도록 합니다. 사용자에게 DTP4에 대한 프롬프트가 표시될 경우 DTP3에서 사용되는 차원만 표시됩니다.
  - **없음** - 디자인 타임 프롬프트에 대해 종속성이 정의되지 않았습니다.
  
6. **선택사항: 차원 한도 선택 버튼** ()을 누르고 다음을 수행합니다.
  - **밀도에서 다음 옵션 중 하나를 선택합니다.**
    - **희소 표시** - DTP에 대해 희소 차원만 표시합니다.
    - **밀집 표시** - DTP에 대해 밀집 차원만 표시합니다.
    - **둘 다** - DTP에 대해 밀집 및 희소 차원을 둘 다 표시합니다.
  - **차원 유형**에서 차원 유형을 선택하여 DTP에 선택한 차원 유형만 표시되도록 제한합니다.
 

예를 들어 사용자가 *계정*, *엔티티* 및 *연도*에 대한 멤버만 입력하게 하려는 경우 **차원 유형**에서 해당 차원을 선택합니다. 입력하라는 메시지가 나타날 때 *계정*, *엔티티* 및 *연도* 유형의 차원만 표시됩니다.

## 멤버 DTP


애플리케이션의 선택된 차원에서 여러 멤버를 정의합니다.

멤버 디자인 타임 프롬프트를 생성하려면 다음을 수행합니다.

1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
2. **디자인 타임 프롬프트** 탭에서 를 누릅니다.
3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - **이름**을 입력하고 **유형**으로 **멤버(둘 이상)**를 선택합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.
    - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.
    - **읽기 전용** - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
  - **DTP 텍스트**에 입력하라는 메시지서 사용자에게 표시할 텍스트를 입력합니다.

프롬프트?를 선택하는 경우 **DTP 텍스트**는 필수입니다.


- **선택 사항입니다. 기본값에서 DTP의 기본값을 지정합니다.**

기본값을 지정하려면 을 누르고 **멤버**를 선택한 다음 **멤버 선택기**에서 멤버를 선택합니다. **차원** 드롭다운에서 선택한 속성 차원이 표시됩니다. 이 차원을 확장하여 사용 가능한 멤버를 표시합니다.


- **스크립트 템플릿만 해당. 제한에서 DTP, 멤버 또는 사용자가 선택할 수 있는 멤버 세트를 반환하는 함수를 입력합니다.**

예를 들어 메시지가 표시될 때 상위 "Revenues" 아래의 레벨 0 계정만 표시하도록 DTP를 제한하려면 **차원**으로 "계정"을 선택한 후 **한도**에 "ILvl0Descendants(Revenues)"를 입력합니다.

**참고:** 한도에 대한 함수를 사용하는 경우 Oracle Essbase 함수 대신 Planning 함수를 사용하는 것이 좋습니다. Essbase 함수 평가에 동적 멤버가 포함된 경우와 같이 Essbase 함수는 필요한 멤버를 반환하지 않는 경우가 있습니다. 멤버(둘 이상) 디자인 시간 프롬프트의 한도에 Essbase 함수 "@Relative("Mbr Name", 0)" 대신 "ILvl0Descendants("Mbr Name")"를 사용하십시오.

5. **선택사항: 종속성 버튼**()을 누르고 옵션을 선택합니다.

- **제외** - 디자인 타임 프롬프트(예: DTP2)가 다른 디자인 타임 프롬프트(예: DTP1)를 제외하도록 합니다. 사용자에게 DTP2에 대한 프롬프트가 표시될 경우 DTP1에서 사용되지 않는 차원만 표시됩니다.
- **포함** - 디자인 타임 프롬프트(예: DTP4)가 다른 디자인 타임 프롬프트(예: DTP3)를 포함하도록 합니다. 사용자에게 DTP4에 대한 프롬프트가 표시될 경우 DTP3에서 사용되는 차원만 표시됩니다.
- **없음** - 디자인 타임 프롬프트에 대해 종속성이 정의되지 않았습니다.

6. **선택사항: 차원 한도 선택 버튼**()을 누르고 다음을 수행합니다.

- **밀도에서 다음 옵션 중 하나를 선택합니다.**
  - **희소 표시** - DTP에 대해 희소 차원만 표시합니다.
  - **밀집 표시** - DTP에 대해 밀집 차원만 표시합니다.
  - **둘 다** - DTP에 대해 밀집 및 희소 차원을 둘 다 표시합니다.
- **차원 유형에서 차원 유형을 선택하여 DTP에 선택한 차원 유형만 표시되도록 제한합니다.**


예를 들어 사용자가 **계정**, **엔티티** 및 **연도**에 대한 멤버만 입력하게 하려는 경우 **차원 유형**에서 해당 차원을 선택합니다. 입력하라는 메시지가 나타날 때 **계정**, **엔티티** 및 **연도** 유형의 차원만 표시됩니다.

## 멤버 범위 DTP





애플리케이션의 선택된 차원에서 멤버 범위를 정의합니다.


멤버 범위 DTP에는 각 차원의 멤버가 하나 또는 여러 개 포함될 수 있습니다. 예를 들어, 멤버 범위 유형 DTP를 사용하여 규칙의 멤버를 입력하도록 사용자에게 요청하는 POV 단계를 생성할 수 있습니다.

멤버 범위 디자인 타임 프롬프트를 생성하려면 다음을 수행합니다.

1. **템플릿 디자이너에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.**
2. **디자인 타임 프롬프트 탭에서**  **를 누릅니다.**



3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - 이름을 입력하고 유형으로 **멤버 범위**를 선택합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.
    - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.
    - **읽기 전용** - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
  - **DTP 텍스트**에 입력하라는 메시지에서 사용자에게 표시할 텍스트를 입력합니다. **프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.
  - **선택 사항입니다. 기본값**에서 DTP의 기본값을 지정합니다. 기본값을 지정하려면 다음 중 하나를 수행합니다.
    - 를 누르고 각 차원 옆의 를 누른 후 **DTP, 멤버 또는 함수**를 선택합니다.
    - 을 눌러 DTP를 기본값으로 사용합니다.
5. **선택사항: 종속성 버튼**()을 누르고 옵션을 선택합니다.
  - **제외** - 디자인 타임 프롬프트(예: DTP2)가 다른 디자인 타임 프롬프트(예: DTP1)를 제외하도록 합니다. 사용자에게 DTP2에 대한 프롬프트가 표시될 경우 DTP1에서 사용되지 않는 차원만 표시됩니다.
  - **포함** - 디자인 타임 프롬프트(예: DTP4)가 다른 디자인 타임 프롬프트(예: DTP3)를 포함하도록 합니다. 사용자에게 DTP4에 대한 프롬프트가 표시될 경우 DTP3에서 사용되는 차원만 표시됩니다.
  - **없음** - 디자인 타임 프롬프트에 대해 종속성이 정의되지 않았습니다.



멤버 범위 DTP는 멤버 범위, 횡차원, 차원, 차원 등의 DTP 유형에 종속되어 있을 수 있습니다.
6. **선택사항: 차원 한도 선택 버튼**()을 누르고 다음을 수행합니다.
  - **밀도**에서 다음 옵션 중 하나를 선택합니다.
    - **희소 표시** - DTP에 대해 희소 차원만 표시합니다.
    - **밀집 표시** - DTP에 대해 밀집 차원만 표시합니다.
    - **둘 다** - DTP에 대해 밀집 및 희소 차원을 둘 다 표시합니다.
  - **차원 유형**에서 차원 유형을 선택하여 DTP에 선택한 차원 유형만 표시되도록 제한합니다.
 

예를 들어 사용자가 **계정**, **엔티티** 및 **연도**에 대한 멤버만 입력하게 하려는 경우 **차원 유형**에서 해당 차원을 선택합니다. 입력하라는 메시지가 나타날 때 **계정**, **엔티티** 및 **연도** 유형의 차원만 표시됩니다.

## 숫자 DTP

숫자를 정의합니다. 예를 들어 Numeric DTP를 공식에 사용할 수 있습니다.


숫자 디자인 타임 프롬프트를 생성하려면 다음을 수행합니다.

1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
2. **디자인 타임 프롬프트** 탭에서  를 누릅니다.
3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - 이름을 입력하고 **유형**으로 **숫자**를 선택합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.
    - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.
    - **읽기 전용** - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
  - **DTP 텍스트**에 입력하라는 메시지에서 사용자에게 표시할 텍스트를 입력합니다. **프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.
  - **선택 사항**입니다. **기본값**에서  를 누르고 **DTP**를 선택한 후 **DTP 값**을 선택합니다.

## 비밀번호 DTP

디자인 타임 프롬프트에 사용할 비밀번호를 정의합니다.

비밀번호 디자인 타임 프롬프트를 생성하려면 다음을 수행합니다.

1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
2. **디자인 타임 프롬프트** 탭에서  를 누릅니다.
3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - 이름을 입력하고 **유형**으로 **비밀번호**를 선택합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.
    - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.

- 읽기 전용 - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
- **DTP 텍스트**에 입력하라는 메시지에서 사용자에게 표시할 텍스트를 입력합니다.  
**프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.
- **선택 사항**입니다. 기본값에서 비밀번호 값을 입력합니다.



## 제한된 목록 DTP

제한된 목록을 정의합니다.

예를 들어 할당 시스템 템플릿에서 반올림 방법을 선택한 다음 제한된 목록 디자인 타임 프롬프트를 사용하여 템플릿에서 사용할 반올림 유형을 선택할 수 있습니다.

템플릿 스크립트에서 각 반올림 유형에 대한 스크립트를 생성할 수 있으며, 사용자가 해당 반올림 옵션을 선택하는 경우에만 스크립트가 사용됩니다.


제한된 목록 프롬프트를 생성하려면 다음을 수행합니다.

1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
2. **디자인 타임 프롬프트** 탭에서  를 누릅니다.
3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - 이름을 입력하고 **유형**으로 **제한된 목록**을 선택합니다.
  - **제한된 목록** 버튼()을 누르고 **규칙 빌더 값** 및 **대체 값**을 입력합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.
    - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.
    - **읽기 전용** - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
  - **DTP 텍스트**에 입력하라는 메시지에서 사용자에게 표시할 텍스트를 입력합니다.  
**프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.
5. **선택사항: 기본값**에서 디자인 타임 프롬프트에 표시할 기본값을 입력합니다.  
기본값을 정의하려면 **기본값** 드롭다운을 누르고 **제한된 목록** 탭에 입력한 **규칙 빌더 값**을 선택합니다.

## 구분자 DTP

구분자는 템플릿 단계의 지칭 또는 공백입니다.

구분자 디자인 타임 프롬프트를 생성하려면 다음을 수행합니다.



1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
2. **디자인 타임 프롬프트** 탭에서  를 누릅니다.

3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - 이름을 입력하고 유형으로 구분자를 선택합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.
    - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.
    - **읽기 전용** - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
  - **DTP 텍스트**에 입력하라는 메시지에서 사용자에게 표시할 텍스트를 입력합니다. **프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.
  - **선택 사항입니다.** 기본값에서 DTP의 기본값을 입력합니다.

## 문자열 DTP

텍스트 문자열을 정의합니다.



문자열 디자인 타임 프롬프트를 생성하려면 다음을 수행합니다.


1. **템플릿 디자이너**에서 기존 템플릿을 열거나 새 템플릿을 생성합니다.
2. **디자인 타임 프롬프트** 탭에서  를 누릅니다.
3. 이미 템플릿에 대해 디자인 타임 프롬프트가 정의되어 있는 경우 새 디자인 타임 프롬프트를 입력할 위치를 선택합니다.
4. 다음 정보를 입력합니다.
  - 이름을 입력하고 유형으로 문자열을 선택합니다.
  - 원하는 옵션을 선택합니다.
    - **프롬프트?** - 단계에서 DTP를 사용할 수 있도록 하려면 이 옵션을 선택합니다. 규칙에서 템플릿을 사용하거나 Oracle Hyperion Planning에서 스크립트 템플릿을 실행하는 경우 입력하라는 메시지가 나타나는 템플릿 단계에 DTP가 표시됩니다.
    - **필수?** - 디자인 타임 프롬프트에 대한 데이터 입력을 필수로 지정하려면 이 옵션을 선택합니다.
    - **읽기 전용** - 사용자가 프롬프트를 읽을 수만 있게 하려면 이 옵션을 선택합니다.
  - **DTP 텍스트**에 입력하라는 메시지에서 사용자에게 표시할 텍스트를 입력합니다. **프롬프트?**를 선택하는 경우 **DTP 텍스트**는 필수입니다.
  - **선택 사항입니다.** 기본값에서  를 누르고 **DTP**, **멤버** 또는 **함수** 중 하나를 선택하고 DTP, 멤버 또는 함수를 기본값으로 입력합니다.

## 디자인 타임 프롬프트에 대한 단계 생성

템플릿 마법사 디자인어를 사용하여 템플릿에 대한 단계를 생성합니다. 마법사를 사용하면 각 단계에서 표시할 디자인 타임 프롬프트, 단계를 표시할 것인지 또는 숨길 것인지 여부(사용 조건), 오류 또는 경고 메시지를 표시할 것인지 여부(검증 조건)를 결정할 수 있습니다.

디자인 타임 프롬프트에 대한 단계를 생성하려면 다음을 수행합니다.


1. 디자인 타임 프롬프트 탭에서 를 누릅니다.
2. 템플릿 마법사 디자인어에서 를 눌러 마법사에서 단계를 생성합니다.
3. 단계 추가 대화상자에서 단계에 대한 정보를 입력하고 **확인**을 누릅니다.  
추가한 단계는 **템플릿 마법사 디자인어**의 **단계** 드롭다운 목록에 표시됩니다.
4. 단계에 표시할 디자인 타임 프롬프트를 **사용 가능한 DTP**에서 **선택한 DTP**로 이동합니다.
5. **사용 조건** 탭에서 단계를 표시할지 숨길지에 대한 정보를 입력합니다.

- **조건 접두어** 드롭다운에서 접두어를 선택합니다.
- **DTP**에서 를 눌러 DTP 또는 함수를 선택합니다.
- **연산자** 드롭다운에서 연산자를 선택합니다.
- **값**에서 조건 값을 입력하거나 선택하십시오.

조건에 있는 모든 문을 정의할 때까지 이 단계를 반복합니다. 행을 추가하려면 마지막 행에서 더하기(+) 아이콘을 누릅니다.

조건에 있는 첫 행은 IF 문입니다. 각 추가 행은 AND 문입니다. 예를 들면 다음과 같습니다.

각 행은 기본적으로 AND 명령문이지만 AND를 누르면 OR로 변경할 수 있습니다.

6. **검증 조건** 탭에서 원하지 않는 데이터(오류) 또는 원하지 않을 수 있는 데이터(경고)를 입력하는 경우 오류 또는 경고 메시지를 표시할지 여부에 대한 정보를 입력합니다.
  - **검증 조건**에서 를 눌러 검증 조건을 정의합니다.
  - **메시지 레벨**에서 **오류** 또는 **경고**를 선택합니다.  
오류 메시지가 표시되면 다음 단계로 진행되지 않습니다. 경고 메시지가 표시되는 경우에는 경고 메시지서 **확인**을 누르면 다음 단계로 진행할 수 있습니다.
  - **검증 메시지**에서 해당 단계의 사용자에게 표시될 메시지를 입력합니다.

단계에 오류나 경고를 더 추가하려면 더하기(+) 아이콘을 누릅니다.

7. **확인**을 누르고 를 누릅니다.

## 디자인 타임 프롬프트에 대한 종속성 정의

### 주:



사용자정의 템플릿 및 디자인 타임 프롬프트는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 애플리케이션에서 지원됩니다.

횡차원, 차원(둘 이상), 차원, 멤버, 멤버(둘 이상) 및 멤버 범위 디자인 타임 프롬프트에 대한 포함 및 제외 종속성을 정의할 수 있습니다. 다른 프롬프트(예: DTP2)를 포함하도록 프롬프트(예: DTP1)를 지정하면 사용자가 DTP1을 확인할 때 DTP2의 차원만 표시됩니다. 다른 프롬프트(예: DTP4)를 제외하도록 프롬프트(예: DTP3)를 지정하면 사용자가 DTP3을 확인할 때 DTP4에서 사용되지 않는 차원만 표시됩니다.

### 주:

디자인 타임 프롬프트는 디자인 타임 프롬프트 그리드에서 앞에 있는 디자인 타임 프롬프트만 포함하거나 제외할 수 있으므로 디자인 타임 프롬프트의 순서는 매우 중요합니다.

종속성을 정의하려면 다음을 수행합니다.

1. 시스템 뷰에서 사용자정의 템플릿을 생성하거나 엽니다.
2. 디자인 타임 프롬프트 탭에서  아이콘을 누르고 다음 유형 중 하나를 선택합니다.
  - 횡차원
  - 차원
  - 차원(둘 이상)
  - 멤버
  - 멤버(둘 이상)
  - 멤버 범위
  - UDA
3. 종속성 탭에서 디자인 타임 프롬프트를 선택하고 종속성을 옵션을 선택합니다.
  - 제외 - 디자인 타임 프롬프트(예: DTP2)가 다른 디자인 타임 프롬프트(예: DTP1)를 제외하도록 합니다. 사용자에게 DTP2에 대한 프롬프트가 표시될 경우 DTP1에서 사용되지 않는 차원만 표시됩니다.
  - 포함 - 디자인 타임 프롬프트(예: DTP4)가 다른 디자인 타임 프롬프트(예: DTP3)를 포함하도록 합니다. 사용자에게 DTP4에 대한 프롬프트가 표시될 경우 DTP3에서 사용되는 차원만 표시됩니다.
  - 없음 - 디자인 타임 프롬프트에 대해 종속성이 정의되지 않았습니다.
4. 확인을 누르고  를 누릅니다.


## 횡차원, 차원, 차원(둘 이상), 멤버 및 멤버 범위 디자인 타임 프롬프트에 대한 차원 한도 정의


### 주:

사용자정의 템플릿 및 디자인 타임 프롬프트는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 애플리케이션에서 지원됩니다.

사용자정의 템플릿에 대한 *횡차원*, *차원*, *차원(둘 이상)*, *멤버* 또는 *멤버 범위* 디자인 타임 프롬프트를 생성할 때 프롬프트를 밀집 및 희소 차원에 표시할지, 밀집 차원에만 표시할지, 희소 차원에만 표시할지를 지정해야 합니다.

*횡차원*, *차원*, *차원(둘 이상)*, *멤버* 또는 *멤버 범위* 디자인 타임 프롬프트를 표시할 차원 유형을 선택하려면 다음을 수행합니다.

1. 시스템 뷰에서 사용자정의 템플릿을 생성하거나 엽니다.
2. 디자인 타임 프롬프트 탭에서  아이콘을 누르고 다음 유형 중 하나를 선택합니다.
  - 횡차원
  - 차원
  - 차원(둘 이상)
  - 멤버
  - 멤버 범위
3. 제한 정의 탭을 선택합니다.
  - 밀도에서 다음 옵션 중 하나를 선택합니다.
    - 희소 표시 - DTP에 대해 희소 차원만 표시합니다.
    - 밀집 표시 - DTP에 대해 밀집 차원만 표시합니다.
    - 둘 다 - DTP에 대해 밀집 및 희소 차원을 둘 다 표시합니다.
  - 차원 유형에서 차원 유형을 선택하여 디자인 타임 프롬프트에 선택한 차원 유형만 표시되도록 제한합니다.

예를 들어 사용자가 *계정*, *엔티티* 및 *연도*에 대한 멤버만 입력하게 하려는 경우 *차원 유형*에서 해당 차원을 선택합니다. 입력하라는 메시지가 나타날 때 *계정*, *엔티티* 및 *연도* 유형의 차원만 표시됩니다.
4. 확인을 누르고  를 누릅니다.

## 디자인 타임 프롬프트에서 텍스트 찾기 및 바꾸기

### 주:


사용자정의 템플릿 및 디자인 타임 프롬프트는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 애플리케이션에서 지원됩니다.

사용자정의 템플릿의 디자인 타임 프롬프트에서 텍스트 문자열을 검색하여 바꿀 수 있습니다.

### 주의:

디자인 타임 프롬프트를 바꿀 순서 및 위치가 매우 중요합니다. 디자인 타임 프롬프트 탭 및 템플릿 디자이너 탭에서 디자인 타임 프롬프트 이름을 찾아서 바꾸려면 먼저 디자인 타임 프롬프트 탭에서 템플릿을 검색하여 저장한 다음 템플릿 디자이너 탭에서 찾아서 바꿔야 합니다. 검색을 이 순서대로 수행하지 않으면 템플릿 디자이너 탭에서 디자인 타임 프롬프트 이름이 바뀌지 않습니다.

디자인 타임 프롬프트에서 텍스트를 검색하여 바꾸려면 다음을 수행합니다.

1. 시스템 뷰에서 사용자정의 템플릿을 엽니다.
2. 디자인 타임 프롬프트 탭에서  를 눌러 찾아서 바꿀 정보를 입력하고 **바꾸기** 또는 **모두 바꾸기**를 누릅니다.

## 사용자정의 템플릿 열기

사용자정의 템플릿은 시스템 뷰, 배포 뷰 및 사용자정의 뷰에서 열 수 있습니다.

### 주:

사용자정의 템플릿 및 디자인 타임 프롬프트는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 애플리케이션에서 지원됩니다.

규칙 디자이너의 비즈니스 규칙 순서도에서도 사용자정의 템플릿을 열 수 있습니다.

사용자정의 템플릿을 열려면 템플릿을 두 번 누르거나 템플릿을 마우스 오른쪽 버튼으로 누른 후 열기를 선택합니다.



## 사용자정의 템플리트 새로고침

사용자정의 템플리트를 생성한 후 템플리트 노드에 새 템플리트를 표시하려면 시스템 뷰에서 애플리케이션 목록을 새로고쳐야 합니다.

사용자정의 템플리트가 속한 애플리케이션 유형, 애플리케이션, 계산 또는 계획 유형을 새로고치면 **템플리트** 노드가 기본적으로 새로고쳐집니다. 그러나 **템플리트** 노드를 새로고쳐도 애플리케이션 목록의 상위 레벨(계산 또는 계획 유형, 애플리케이션, 애플리케이션 유형)은 새로고쳐지지 않습니다.

사용자정의 템플리트 목록을 새로고치려면 **템플리트** 노드를 마우스 오른쪽 버튼으로 누른 다음 **새로고침**을 선택합니다.

### 주:

사용자정의 템플리트 및 디자인 타임 프롬프트는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 애플리케이션에서 지원됩니다.

## 사용자정의 템플리트에 대한 사용 표시

사용자정의 템플리트를 사용하는 비즈니스 규칙 목록을 표시할 수 있습니다.

사용자정의 템플리트 사용 보기는 사용자정의 템플리트를 삭제할 때 이 템플리트를 사용하고 있는 객체(있는 경우)를 알아야 하는 경우 유용합니다. 템플리트를 삭제하기 전에 이 템플리트를 사용하고 있는 객체에서 템플리트를 제거해야 합니다.

사용자정의 템플리트 사용을 표시하려면 **시스템 뷰**에서 **템플리트** 노드를 확장하고 템플리트를 마우스 오른쪽 버튼으로 누른 후 **사용 표시**를 선택합니다.

### 주:

사용자정의 템플리트 및 디자인 타임 프롬프트는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 애플리케이션에서 지원됩니다.

## 사용자정의 템플리트 복사 및 붙여넣기

사용자정의 템플리트를 복사하여 다른 비즈니스 규칙에 붙여넣을 수 있습니다.

템플리트를 복사할 규칙은 같은 애플리케이션 유형 및 계산 또는 계획 유형에 속해 있어야 합니다. 예를 들어 Oracle Hyperion Planning 애플리케이션에서 Plan1 계획 유형의 규칙에 있는 사용자정의 템플리트를 Planning 애플리케이션에 있는 Plan1 계획 유형의 다른 규칙에 복사할 수 있지만 Capital Asset 계획 유형의 규칙에는 복사할 수 없습니다.

사용자정의 템플리트를 복사하여 붙여넣으려면 다음 중 하나를 수행합니다.

- **템플릿** 노드를 확장하고, 해당 템플릿을 마우스 오른쪽 버튼으로 누르고, **복사 위치**를 선택하고, **복사 위치** 대화상자에 필요한 정보를 입력한 후 **확인**을 누릅니다.
- 비즈니스 규칙 순서도에서 복사할 템플릿을 마우스 오른쪽 버튼으로 누르고 **복사**를 선택합니다. 템플릿을 붙여넣으려는 비즈니스 규칙을 열고 비즈니스 규칙 순서도에서 마우스 오른쪽 버튼을 누른 다음 **붙여넣기**를 선택합니다.

 **주:**

사용자정의 템플릿 및 디자인 타임 프롬프트는 Oracle Hyperion Financial Management, Planning 및 Oracle Essbase 블록 저장영역 애플리케이션에서 지원됩니다.

## 사용자정의 템플릿 삭제

시스템 뷰, 사용자정의 뷰 및 배포 뷰에서 사용자정의 템플릿을 삭제할 수 있습니다.

사용자정의 템플릿을 삭제하기 전에 템플릿이 비즈니스 규칙 또는 사용자정의 폴더에서 사용되고 있지 않은지 확인해야 합니다. 템플릿 사용을 표시하려면 템플릿을 마우스 오른쪽 버튼으로 누른 후 **사용 표시**를 선택합니다.

사용자정의 템플릿을 삭제하려면 **템플릿** 노드를 확장하고 템플릿을 마우스 오른쪽 버튼으로 누른 후 **삭제**를 선택합니다.

 **주:**

스크립트 템플릿이 Oracle Hyperion Planning에 배포된 경우 **삭제**를 선택하면 다음 메시지가 표시됩니다.

Do you want to delete the selected item(s)? Some of the selected objects were deployed to Planning. Do you want them to be deleted from Planning Server?

상자를 선택하면 Planning 및 Oracle Hyperion Calculation Manager 모두에서 스크립트 템플릿이 삭제됩니다.

 **주:**

사용자정의 템플릿 및 디자인 타임 프롬프트는 Oracle Hyperion Financial Management, Planning 및 Oracle Essbase 블록 저장영역 애플리케이션에서 지원됩니다.

# 그래픽 사용자정의 템플릿에서 텍스트 찾기 및 바꾸기

사용자정의 템플릿에서 텍스트 문자열을 검색할 수 있습니다. 텍스트 문자열의 모든 인스턴스를 바꿀 수도 있습니다.

## 주:

사용자정의 템플릿 및 디자인 타임 프롬프트는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 애플리케이션에서 지원됩니다.

문자열을 검색할 때 Oracle Hyperion Calculation Manager는 템플릿의 순서도에서 선택된 구성요소 뒤에 있는 첫 번째 구성요소에서 검색을 시작하여 순서도의 끝까지 검색한 다음 순서도의 처음부터 시작하여 순서도에서 선택된 구성요소에 도달할 때까지 검색합니다. 일치하는 텍스트 문자열을 찾은 후 다른 항목을 검색하려면 다시 검색을 시작하여 다음 항목을 찾아야 합니다.

사용자정의 템플릿에서 텍스트를 검색하거나 텍스트를 검색하고 바꾸려면 다음을 수행합니다.

1. 시스템 뷰에서 사용자정의 템플릿을 엽니다.
2. 다음 중 하나를 수행합니다.
  - **템플릿 디자이너** 탭에서 **편집, 찾기** 순으로 선택합니다. **찾기** 대화상자에 찾을 텍스트를 입력한 다음 **찾기**를 누릅니다.
  - **템플릿 디자이너** 탭에서 **편집, 모두 바꾸기** 순으로 선택합니다. **바꾸기** 대화상자에서 찾아서 바꿀 정보를 입력하고 **모두 바꾸기**를 선택합니다.

# 6

## 구성요소를 사용하여 비즈니스 규칙 및 템플릿 디자인

### 참조:

- [구성요소 정보](#)  
비즈니스 규칙 및 템플릿에는 여러 유형의 구성요소가 포함될 수 있습니다.
- [공식 구성요소](#)  
공식 구성요소는 공식 계산 문으로 구성됩니다.
- [스크립트 구성요소](#)  
스크립트 구성요소는 비즈니스 규칙과 템플릿에서 사용할 수 있습니다.
- [조건 구성요소](#)  
조건 구성요소는 true 또는 false인 조건문으로 구성됩니다.
- [멤버 블록 구성요소](#)  
멤버 블록 구성요소는 스크립트에서 하나 이상의 명령문을 둘러쌀 멤버를 정의합니다.
- [멤버 범위 구성요소](#)  
멤버 범위 구성요소는 Planning 차원의 멤버 범위로 구성된 루프 유형입니다.
- [데이터 범위 구성요소\(Financial Management 사용자만 해당\)](#)
- [셀 텍스트 범위 구성요소\(Financial Management 사용자만 해당\)](#)
- [고정 루프 구성요소](#)  
고정 루프 구성요소는 메타데이터 멤버 목록을 정해진 횟수만큼 순환하는 객체입니다.
- [메타데이터 루프 구성요소](#)  
메타데이터 루프 구성요소를 사용하면 함수(메타데이터) 또는 시작 및 끝 값(고정)을 사용하여 여러 멤버에 값을 지정할 수 있습니다.
- [DTP 지정 구성요소](#)  
DTP 지정 구성요소를 사용하여 사용자정의 템플릿에 있는 디자인 타임 프롬프트에 DTP, 멤버, 함수 또는 입력된 텍스트를 지정합니다.
- [스크립트 및 공식 구성요소 공유](#)  
공유 공식 또는 스크립트 구성요소는 공식 레벨과 스크립트 레벨에 있습니다.
- [구성요소 복사](#)  
비즈니스 규칙 구성요소의 1차 하위를 복사하여 붙여넣거나 비즈니스 규칙 공식 또는 스크립트 구성요소에 대한 참조를 복사하여 붙여넣습니다.
- [구성요소 저장](#)  
구성요소 디자이너에서 공식 및 스크립트 구성요소를 디자인한 다음 이를 저장합니다.
- [공식 및 스크립트 구성요소 새로고치기](#)  
공식 또는 스크립트 구성요소를 생성한 후에는 시스템 뷰에서 애플리케이션 목록을 새로고쳐야 공식 또는 스크립트 노드에서 확인할 수 있습니다.
- [공식 및 스크립트 구성요소의 사용 표시](#)  
스크립트 또는 공식 구성요소의 사용을 표시할 수 있습니다.
- [순서도의 구성요소 작업](#)  
순서도의 구성요소에 대해 작업을 수행할 수 있습니다.

## 구성요소 정보

비즈니스 규칙 및 템플리트에는 여러 유형의 구성요소가 포함될 수 있습니다.

- **공식 구성요소** - 멤버, 함수 및 조건문을 사용하여 작성하거나 디자인할 수 있는 계산 명령문입니다.
- **스크립트 구성요소** - Visual Basic(Oracle Hyperion Financial Management의 경우) 또는 Oracle Essbase(Oracle Hyperion Planning 및 Essbase의 경우) 계산 스크립트 명령문만 포함됩니다.
- **조건 구성요소** - true 또는 false인 조건문(if...Then 문)입니다.
- **멤버 블록 구성요소** - 사용자가 지정하는 하나의 멤버가 포함됩니다(Planning 및 Essbase 블록 저장영역 애플리케이션 사용자만 해당).
- **멤버 범위 구성요소(또는 메타데이터 루프)** - 메타데이터 멤버 목록(예: 계정 목록)이 포함됩니다.
- **고정 루프 구성요소** - 계정과 같은 멤버 목록 등을 순환하는 메타데이터 루프가 포함됩니다.

Financial Management 고정 루프에는 예를 들어 Financial Management 데이터를 고정된 횟수만큼 순환하는 데이터 루프도 포함될 수 있습니다.

- **메타데이터 루프 구성요소** - (사용자정의 템플리트 사용자만 해당) 상위 차원의 하위 멤버에 적용할 수 있는 매개변수가 포함됩니다.
- **DTP 지정 구성요소** - (사용자정의 템플리트 사용자만 해당) 해당 구성요소에 대해 정의한 디자인 타임 프롬프트 및 조건이 포함됩니다.
- **데이터 범위 구성요소(또는 데이터 루프)** - (Financial Management 사용자만 해당) 데이터 레코드 목록(예: 계정 값 목록)이 포함됩니다.

공식 및 스크립트 구성요소를 해당 요소가 사용되는 규칙 및 템플리트와 별개로 생성할 수 있습니다. 이들 요소는 독립 객체이므로 시스템 뷰에서 열고, 저장하고, 편집하고, 삭제하고, 내보낼 수 있습니다.

공식 및 스크립트 구성요소와 달리 다른 구성요소 유형은 규칙 및 템플리트에서 생성해야 합니다. 이들 요소는 속해 있는 규칙 및 템플리트와 독립적으로 열거나 저장하거나 삭제하거나 내보낼 수 없습니다.

### 주:

구성요소를 생성할 때 작업 중인 비즈니스 규칙, 구성요소, 템플리트 및 변수를 열어 두어야 할 수 있습니다. Oracle Hyperion Calculation Manager에서는 이러한 객체를 탭 인터페이스에 표시하므로 구성요소를 생성하는 동안 탭 간에 쉽게 이동할 수 있습니다. Calculation Manager 내에서 탭 10개를 열어 둘 수 있지만 성능을 최적화하려면 동시에 여는 객체 수가 10개를 초과하지 않는 것이 좋습니다.

## 공식 구성요소

공식 구성요소는 공식 계산 문으로 구성됩니다.

**참조:**

- 공식 구성요소 정보
- 공식 구성요소 생성
- 공식 구성요소 디자인
- 공식 구성요소 열기
- 공식 구성요소 편집
- 공식 구성요소 삭제
- 공식 구성요소 복사 및 붙여넣기

## 공식 구성요소 정보

공식 구성요소는 공식 계산 문으로 구성됩니다. 공식의 계산 문을 생성하려면 멤버, 함수 및 조건을 입력하거나 선택합니다. 공식을 생성하면 공식의 각 계산 문이 구성요소 디자이너에서 그리드 내의 행에 나열됩니다.

**주:**

공식 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

## 공식 구성요소 생성

시스템 뷰에서 공식 구성요소를 생성할 수 있습니다. 공식 구성요소는 비즈니스 규칙 및 템플릿에서 사용할 수 있습니다.

**주:**

공식 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

공식 구성요소를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 새 객체 아이콘을 누릅니다.
2. 애플리케이션 유형을 입력합니다.
3. 애플리케이션을 선택합니다.  
애플리케이션은 Financial Management 또는 Planning 애플리케이션 관리에서 생성된 애플리케이션이거나 Essbase 블록 저장영역 애플리케이션이어야 합니다.
4. 다음 중 한 가지 태스크를 수행합니다.
  - a. Financial Management를 선택한 경우 계산 유형을 선택합니다.
  - b. Planning을 선택한 경우 계획 유형을 선택합니다.

- c. Essbase를 선택한 경우 **데이터베이스**를 선택합니다.

 **주:**

시스템 뷰에서 [공식]을 마우스 오른쪽 버튼으로 누르고 [새로 작성]을 선택하여 새 공식을 생성하면 [새 공식] 대화상자에 사용자가 작업 중인 애플리케이션 유형 이름, 애플리케이션 이름 및 계산 유형, 계획 유형 또는 데이터베이스 이름이 입력됩니다.

5. **객체 유형**으로 공식을 선택합니다.
6. 공식 이름을 입력하고 **확인**을 누릅니다.  
공식이 구성요소 디자이너에 표시됩니다. 공식 구성요소를 디자인하려면 [공식 구성요소 디자인](#)을 참조하십시오.

## 공식 구성요소 디자인

시스템 뷰나 다른 모든 뷰에서 공식 구성요소를 생성할 수 있습니다. 또한 비즈니스 규칙이나 템플릿을 디자인할 때 규칙 디자이너나 템플릿 디자이너에서 공식 구성요소를 생성할 수도 있습니다. 공식 구성요소는 비즈니스 규칙 및 템플릿에서 사용할 수 있습니다.

 **주:**

공식 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

공식 구성요소를 생성하려면 다음을 수행합니다.

1. **시스템 뷰**에서 공식을 마우스 오른쪽 버튼으로 누르고 **새로 작성**을 선택합니다.  
**새 공식** 대화상자에 애플리케이션 유형, 애플리케이션 및 계산 유형, 계획 유형 또는 데이터베이스가 채워집니다.

 **주:**

비즈니스 규칙 또는 템플릿에서 새 공식 구성요소를 비즈니스 규칙 또는 템플릿의 순서대로 끌어와 공식 구성요소를 생성할 수도 있습니다.

2. 공식 구성요소의 이름과 이 구성요소가 속하는 애플리케이션 유형 및 애플리케이션을 입력하거나 기본값을 적용합니다. 다음 중 하나를 수행합니다.
  - **Financial Management 사용자만 해당:** Financial Management 공식 구성요소를 생성하는 경우 계산 유형을 입력합니다.
  - **Planning 사용자만 해당:** Planning 공식 구성요소를 생성하는 경우 계획 유형을 입력합니다.
  - **Essbase 사용자만 해당:** Essbase 공식 구성요소를 생성하는 경우 데이터베이스를 입력합니다.

3. 확인을 누릅니다.
4. **선택 사항: 구성요소 디자이너에서 다음 태스크를 수행할 수 있습니다.**
  - 그리드 추가를 누르고 그리드 삽입(앞), 그리드 삽입(뒤) 또는 끝에 그리드 삽입을 선택하여 다른 그리드를 추가합니다. 기본적으로 [공식] 탭에는 하나의 그리드가 표시됩니다.
  - 그리드 삭제를 눌러 선택한 그리드를 삭제합니다.
  - 찾기 아이콘을 눌러 작업하고 있는 공식 그리드에서 텍스트를 찾습니다. 다음 태스크를 수행합니다.
  - 찾기 및 바꾸기 아이콘을 눌러 스크립트 내의 텍스트를 찾아서 바꿉니다.
5. 공식에 공식의 캡션을 입력합니다.
6. **템플릿의 공식 구성요소를 생성하는 사용자만 해당:** 공식 구성요소에서 디자인 타임 프롬프트를 사용하려는 경우 공식 탭에서 **디자인 프롬프트 사용**을 선택합니다. 그런 다음 조건 그리드에서 다음 태스크를 수행하여 디자인 타임 프롬프트에 대한 조건을 정의합니다.
  - a. **DTP**에서 디자인 타임 프롬프트를 선택합니다.
  - b. 연산자에서 연산자 = = 또는 <>를 선택합니다.
  - c. 값 드롭다운 목록에서 값을 선택합니다.
  - d. 이 단계를 반복하여 조건에 필요한 만큼 명령문을 생성합니다.

 **팁:**

더하기(+) 및 빼기(-) 아이콘을 눌러 그리드에 행을 추가 및 삭제합니다. And 옆에 있는 필드를 누르고 드롭다운 목록에서 Or을 선택하여 And를 Or로 변경합니다.

7. **Planning 및 Essbase 사용자만 해당:** 기본적으로 공식 구성요소 계산 처리는 그리드에 입력된 첫 번째 멤버를 사용하여 시작됩니다. 다른 멤버를 사용하여 공식 구성요소 계산 처리를 시작하려면 **멤버 블록**에 멤버 또는 함수 이름을 입력하거나 **출입표** 아이콘을 눌러 멤버 선택기에서 멤버를 선택합니다. **구성요소에 멤버 및 함수 추가 정보**를 참조하십시오.
8. 설명에, 생성할 조건문 및 공식문에 대한 설명을 입력합니다.
9. **선택 사항:** 공식 구성요소에 대한 조건문(즉, IF 문)을 생성하려면 **조건 추가**를 누릅니다. **조건 빌더를 사용하여 조건문 생성**을 참조하십시오.
10. **선택 사항:** 표시되는 IF 행에 조건문의 텍스트를 입력하거나 행의 오른쪽 옆에 있는 **조건 추가/편집** 아이콘을 눌러 조건 빌더에 액세스합니다. 조건 빌더에서는 조건문을 그래픽 형식으로 디자인할 수 있습니다.

 **주:**

아래쪽 화살표에서 IF, ELSE IF 및 ELSE를 선택할 수 있지만 기본적으로 첫 번째 명령문은 IF 문이어야 합니다.

11. **선택 사항:** 조건 빌더를 사용하여 IF 조건문을 디자인합니다. **조건 빌더를 사용하여 조건문 생성**을 참조하십시오.
12. 공식 행에서 등호 기호 왼쪽의 행을 눌러 공식문을 생성합니다. **작업** 아이콘을 누르고 다음을 선택합니다.



 주:

9단계에서 11단계까지 조건문(즉, IF 문)을 생성한 경우 이 단계에서 생성하는 공식문은 조건의 THEN 문입니다.

- 변수(변수 작업 참조)
  - 멤버(구성요소에 멤버 및 함수 추가 참조)
  - 함수(함수 작업 참조)
  - **Planning 사용자만 해당:** 스마트 목록(스마트 목록 작업 참조)
13. 공식 행에서 등호 기호 오른쪽의 행을 눌러 공식문을 완료합니다. 작업 아이콘을 누르고 다음을 선택합니다.
- 변수(변수 작업 참조)
  - 멤버(구성요소에 멤버 및 함수 추가 참조)
  - 함수(함수 작업 참조)
  - **Planning 사용자만 해당:** 스마트 목록(스마트 목록 작업 참조)
14. **Financial Management 사용자만 해당:** 각 명령문에 대해 로깅 정보 아이콘을 눌러 로그에 대한 선택적 조건문을 입력합니다. 로깅 정보는 조건에 맞는 경우에만 로그 파일에 기록됩니다. 로깅 정보를 저장하면 로그 텍스트가 있음을 나타내는 가로 라인과 함께 [설명] 아이콘이 표시됩니다. 공식문에 대한 로깅 정보 입력(Financial Management 사용자만 해당)을 참조하십시오.

 주:

로그 텍스트를 로그 파일에 포함하려면 [로깅 사용]을 선택해야 합니다.

15. 설명 아이콘을 눌러 계산 문 행에 대한 설명을 입력합니다. 확인을 누릅니다.
16. 선택 사항: 다른 IF 문, ELSE IF 또는 ELSE 문을 생성하려면 다음 단계를 수행합니다.
- a. 조건 추가를 누릅니다.
  - b. 아래쪽 화살표를 누르고 다음 옵션 중 하나를 선택합니다.
    - **IF:** IF 조건문을 생성하려면 선택합니다. IF 문이 TRUE이면 작업이 수행되고 IF 문이 FALSE이면 다른 작업이 수행됩니다.
    - **ELSE IF:** ELSE IF 조건문을 생성하려면 선택합니다. 연결된 IF 문이 있고 IF 문이 FALSE인 경우 이 명령문의 작업이 수행됩니다.
    - **ELSE:** ELSE 조건문을 생성하려면 선택합니다. 연결된 ELSE IF 문이 있고 ELSE IF 문이 FALSE인 경우 이 명령문의 작업이 수행됩니다.

 주:

IF 또는 ELSE IF 블록에서 조건문을 제거하고 다음 블록에 ELSE IF 문이 포함되어 있는 경우, 다음 블록이 IF 블록으로 변경됩니다. 그러나 다음 블록에 ELSE 문이 포함되어 있는 경우에는 이 블록의 조건문이 제거됩니다.

- c. 공식 구성요소에 대한 추가 명령문을 디자인하려면 9단계에서 15단계를 반복합니다. 공식 그리드는 공식문 모음이 포함된 하나 이상의 공식 블록으로 구성될 수 있습니다. 또한 블록에 대한 설명과 조건을 선택적으로 입력할 수 있습니다.

 팁:

필요한 경우 + 아이콘을 눌러 공식 행을 더 추가합니다.

## 17. 등록정보에서 다음 단계를 완료합니다.

- a. **비즈니스 규칙에 대한 공식 구성요소를 생성하는 사용자만 해당:** 새 객체에서 비즈니스 규칙으로 공식 구성요소를 끌어서 놓을 때 **공유**는 선택되지 않습니다. 이 공식을 공유로 설정하려면 공식의 **등록정보** 탭을 선택하고 **공유**를 선택한 다음 공식 이름을 지정합니다.

시스템 뷰에서 공식 구성요소를 생성하는 경우에는 기본적으로 **공유**가 선택되며 변경할 수 없습니다. 대신, 비즈니스 규칙에서 공식을 복사하려는 경우 공유 공식을 비즈니스 규칙으로 끌어온 다음 **등록정보** 탭에서 **공유** 확인란의 선택을 취소합니다. 그러면 비즈니스 규칙에서 공유 공식의 복사본이 생성됩니다. 공유 공식을 변경해도 공유 공식의 복사본은 변경되지 않습니다. 비즈니스 규칙을 저장하면 공식에 더 이상 이름이 없습니다.

- b. **선택 사항:** 새 이름을 최대 50자까지 입력하여 공식 이름을 편집합니다. 이름은 기본적으로 [새 공식] 대화상자에서 가져옵니다.
- c. **선택 사항:** 공식에 대한 설명을 최대 255자까지 입력합니다.
- d. **선택 사항:** 공식의 캡션을 입력합니다. 캡션은 **디자이너 및 템플릿 디자이너** 순서도에서 공식 아래에 표시됩니다.

 주:

구성요소에 캡션이 없고 구성요소가 공유되지 않는 경우 구성요소의 첫 번째 부분이 순서도에 표시됩니다. 예를 들어 공유되지 않고 캡션이 없는 공식 "Salaries" = 40이 있는 경우 순서도에서 공식 구성요소에 대해 "Salaries" = 40이 표시됩니다. 구성요소가 공유되고 캡션이 없는 경우에는 구성요소의 이름이 순서도에 표시됩니다.

- e. 공식에 대한 설명을 입력합니다. 예를 들어 공식의 용도를 알려 줄 수 있습니다.

18. **Financial Management 사용자만 해당:** **로깅 사용**을 선택합니다. 그러면 공식에 로그 텍스트가 있는 경우 규칙이 실행될 때 로그 텍스트가 로그 파일에 포함됩니다. 규칙, 규칙 세트 및 구성요소에 대해 로깅을 사용할 수 있습니다. 그러나 비즈니스 규칙의 공식문에서 포함할 로그 텍스트(있는 경우)를 지정합니다.

 주:

규칙 세트에 대해서는 로깅을 사용하지 않지만 규칙 세트에 속한 규칙 및 구성요소에 대해 로깅을 사용하는 경우 규칙 세트에 대한 로그 설정이 규칙 및 구성요소에 대한 로그 설정을 재정의하므로 로그 파일이 생성되지 않습니다.

19. **Financial Management 사용자: 타이머 사용**을 선택합니다. 그러면 규칙이 실행될 때 공식을 처리하는 데 걸린 시간이 로그 파일에 기록됩니다. 규칙, 규칙 세트 및 구성요소에 대해 타이머를 사용할 수 있습니다. 타이머가 활성화된 모든 객체에 대한 처리 시간이 로그 파일에 포함됩니다. 예를 들어 타이머가 활성화된 규칙이 있고 이 규칙에 타이머가 활성화된 공식 세 개가 포함된 경우 규칙과 규칙의 각 공식을 처리하는 데 걸린 시간이 기록됩니다.
20. **비즈니스 규칙에 대한 공식 구성요소를 생성하는 Financial Management 사용자만 해당: 사용 안 함**을 선택하여 스크립트가 생성되는 동안 공식 구성요소를 사용하는 비즈니스 규칙 내에서 해당 구성요소를 일시적으로 비활성화합니다.
21. **비즈니스 규칙에 대한 공식 구성요소를 생성하는 사용자만 해당: 사용**에서 공식 구성요소를 사용하는 규칙을 볼 수 있습니다.

 주:

[사용] 탭의 정보는 편집할 수 없습니다.

22.  을 누릅니다.

## 조건 빌더를 사용하여 조건문 생성

조건 빌더를 사용하여 공식 및 조건 구성요소의 조건문을 생성할 수 있습니다. 조건문을 If...Then 문이라고도 합니다. 조건문의 첫 번째 부분(If 부분)이 true이면 조건문의 두 번째 부분(Then 부분)도 true입니다. 조건문의 첫 번째 부분이 false이면 두 번째 부분은 true일 수도 있고 아닐 수도 있습니다.

 주:

공식 및 조건 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

조건문을 생성하려면 다음을 수행합니다.

1. 공식을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **구성요소 디자이너**에서 **조건 추가** 아이콘을 누릅니다.

조건을 빌드하는 데 사용하는 드롭다운 목록 및 다양한 필드가 포함된 두 행이 구성요소 디자이너에 추가되었습니다.

3. 첫 번째 행에서 조건의 첫 번째 명령문을 IF, ELSE 또는 ELSE IF로 시작할지 여부를 선택하고 조건문의 텍스트를 입력합니다. 또는 행 오른쪽에 있는 **조건 추가/편집**

아이콘을 눌러 조건 빌더에 액세스합니다. 조건 빌더에서는 조건문을 그래픽 형식으로 디자인할 수 있습니다.

아래쪽 화살표에서 IF, ELSE IF 및 ELSE를 선택할 수 있지만 기본적으로 첫 번째 명령문은 IF 문이어야 합니다.

4. 조건 빌더에서 **메타데이터 조건** 또는 **데이터 조건**을 선택합니다.
5. 다음 태스크 중 하나를 수행합니다.
  - **함수**(메타데이터 조건의 경우) 또는 **공식**(데이터 조건의 경우)의 값을 입력합니다.
  - **함수** 또는 **공식** 행을 눌러 **작업** 아이콘을 표시합니다. **작업** 아이콘을 누르고 다음 옵션을 선택하여 공식 또는 함수를 생성합니다.
    - 변수([변수 작업 참조](#))
    - 멤버([구성요소에 멤버 및 함수 추가 참조](#))
    - 함수([함수 작업 참조](#))
6. 다음 연산자 중 하나를 입력합니다.
  - =(같음)
  - <(보다 큼)
  - >(보다 작음)
  - <>(보다 크거나 보다 작음)
  - >=(작거나 같음)
  - <=(크거나 같음)
7. 다음 태스크 중 하나를 수행합니다.
  - **값**에 값을 입력합니다.
  - **값** 행을 눌러 **작업** 아이콘을 표시합니다. **작업**을 누르고 다음 옵션을 선택하여 공식 또는 함수의 값을 생성합니다.
    - 변수([변수 작업 참조](#))
    - 멤버([구성요소에 멤버 및 함수 추가 참조](#))
    - 함수([함수 작업 참조](#))
8. 조건에 대한 설명을 입력합니다.
9. **+** 아이콘을 눌러 [조건] 그리드에 조건을 추가합니다. 또한 **+** 아이콘을 사용하여 선택한 행의 사본을 생성하여 [조건] 그리드에 추가할 수도 있습니다. **—** 아이콘을 사용하여 [조건] 그리드에서 선택한 행을 메타데이터 또는 데이터 조건으로 바꿉니다.

 **팁:**

위쪽 및 아래쪽 화살표를 사용하여 그리드에서 조건문의 순서를 조정합니다. [그룹] 및 [그룹 해제] 아이콘을 사용하여 조건문의 일부를 그룹화(괄호 추가) 및 그룹 해제(괄호 제거)합니다. And 옆에 있는 필드를 누르고 드롭다운 목록에서 옵션을 선택하여 And를 Or, And Not 또는 Or Not으로 변경합니다.

10. **확인**을 눌러 **조건 빌더** 대화상자를 종료합니다.  
조건이 IF 문에 삽입됩니다.
11. **선택 사항:** 생성할 각 조건문에 대해 2단계에서 10단계까지 반복합니다.

12. **선택 사항:** 각 추가 조건문에 대해 첫 번째 열을 두 번 눌러 각 조건문을 시작할 연산자를 선택할 수 있는 드롭다운 목록을 표시합니다.
- IF: 첫 번째 조건문 *만* IF로 시작할 수 있습니다. 생성하는 첫 번째 조건문에만 표시되는 기본값입니다.
  - NOT IF: 첫 번째 조건문 *만* NOT IF로 시작할 수 있습니다.
  - AND: 두 개 이상의 조건문으로 구성된 복합문을 생성하려는 경우 첫 번째 문을 제외한 모든 조건문을 AND로 시작할 수 있습니다.
  - OR: 두 개 이상의 조건문으로 구성된 복합문을 생성하려는 경우 첫 번째 문을 제외한 모든 조건문을 OR로 시작할 수 있습니다.
  - AND NOT: 뒤에 오는 조건문을 공식 구성요소에 포함하려는 경우 첫 번째 조건문을 제외한 모든 조건문을 AND NOT으로 시작할 수 있습니다.
  - OR NOT: 뒤에 오는 조건문을 공식 구성요소에서 제외하려는 경우 첫 번째 조건문을 제외한 모든 조건문을 OR NOT으로 시작할 수 있습니다.

 **팁:**

[그룹] 및 [그룹 해제]를 사용하여 조건문에서 괄호를 추가하거나 제거할 수 있습니다. 여러 조건문을 그룹화하거나 그룹 해제하려면 Ctrl + 누르기 또는 Shift + 누르기를 사용하여 그룹화하거나 그룹 해제할 조건문을 선택합니다.

13. **확인을 누릅니다.**  
조건문이 [조건] 행에 삽입됩니다.

## 공식문에 대한 로깅 정보 입력(Financial Management 사용자만 해당)

Oracle Hyperion Financial Management 공식 명령문의 로깅 정보를 입력할 수 있습니다.  
로깅 정보를 입력하려면 다음을 수행합니다.

1. **시스템 뷰**에서 Financial Management 애플리케이션 유형, 애플리케이션, 계산 유형 및 공식을 차례로 확장합니다. 로깅 정보를 추가할 공식을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **구성요소 디자이너**에서 각 명령문에 대해 **로깅 정보** 아이콘을 눌러 로그에 대한 선택적 조건문을 입력합니다. 이 조건에 맞는 경우에만 로그 정보가 로그 파일에 기록됩니다. 로깅 정보를 저장하면 로그 텍스트가 있음을 나타내는 가로 라인과 함께 [설명] 아이콘이 표시됩니다.

 **주:**

로그 텍스트를 로그 파일에 포함하려면 [로깅 사용]을 선택합니다.

3. 변수를 사용하려는 경우 **변수 사용**을 선택합니다. 그런 다음 **변수**에서 변수를 선택합니다.
4. 로그 텍스트를 표준 형식으로 표시하려면 **표준 표시 형식 사용**을 선택합니다.
5. **로그 텍스트**에 로그에 대한 조건문을 입력하거나 **조건 추가/편집** 아이콘을 눌러 조건 빌더에서 조건문을 생성합니다. **조건 빌더를 사용하여 조건문 생성**을 참조하십시오.


6. 로그 표현식에 로그 표현식을 입력하고 **확인**을 누릅니다.

## 공식문에 대한 설명 입력

### 주:

공식 명령문은 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

공식문에 대한 설명을 입력하려면 다음을 수행합니다.

1. 공식을 확장한 상태에서 다음 태스크 중 하나를 수행하십시오.
  - 설명을 추가할 공식이 비즈니스 규칙에 있는 경우 **규칙**을 확장하고 공식 구성요소가 포함된 규칙을 마우스 오른쪽 버튼으로 누른 다음 **열기**를 선택합니다.
  - 설명을 추가할 공식이 템플릿에 있는 경우 **템플릿** 노드를 확장하고 공식 구성요소가 포함된 규칙을 마우스 오른쪽 버튼으로 누른 다음 **열기**를 선택합니다.
  - 공식만 열려면 **공식** 노드를 확장하고 공식을 마우스 오른쪽 버튼으로 누른 다음 **열기**를 선택합니다.
2. 다음 태스크 중 하나를 수행합니다.
  - 비즈니스 규칙에서 공식 구성요소에 대한 설명을 추가하는 경우 **규칙 디자이너** 순서도에서 공식 구성요소와 **공식** 탭을 선택합니다. **설명** 텍스트 상자에 설명을 입력합니다.
  - 템플릿에서 공식에 대한 설명을 추가하는 경우 **템플릿 디자이너** 순서도에서 공식 구성요소와 **등록정보** 탭을 선택합니다. **설명** 텍스트 상자에 설명을 입력합니다.
  - 공식 구성요소 자체에 대한 설명을 추가하는 경우 **구성요소 디자이너**의 **설명** 텍스트 상자에 설명을 입력합니다.
3.  를 누릅니다.

## 공식 구성요소 열기

시스템 뷰에서 또는 규칙 디자이너나 템플릿 디자이너 순서도에서 공식 구성요소를 열 수 있습니다.

### 주:

공식 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

공식 구성요소를 열려면 **시스템 뷰**에서 공식을 확장하고 공식을 마우스 오른쪽 버튼으로 누른 후 **열기**를 선택합니다.

 주:

공식 구성요소가 비즈니스 규칙에 사용되는 경우 공식 구성요소를 마우스 오른쪽 버튼으로 누르고 [열기]를 선택하거나 공식 구성요소를 두 번 눌러 규칙 디자이너의 규칙 순서도에서 열 수도 있습니다.


## 공식 구성요소 편집

공식 구성요소를 구성하는 명령문, 공식 구성요소 계산 결과를 로그 파일에 포함할지 여부 (Oracle Hyperion Financial Management 애플리케이션만 해당) 및 공식 구성요소의 설명 (comment), 캡션, 이름, 설명(description)을 편집할 수 있습니다.

 주:

공식 구성요소는 Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

공식 구성요소를 편집하려면 다음을 수행합니다.

1. 시스템 뷰에서 공식을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 구성요소 디자이너에서 공식 등록정보를 편집하고  를 누릅니다.

공식 구성요소 디자인 및 조건 빌더를 사용하여 조건문 생성을 참조하십시오.

## 공식 구성요소 삭제

규칙이나 템플릿에서 사용되고 있지 않은 공식 구성요소만 삭제할 수 있습니다. 규칙이나 템플릿에 공식 구성요소가 사용되고 있는지 확인하려면 사용을 표시하면 됩니다 [공식 및 스크립트 구성요소의 사용 표시](#)를 참조하십시오. 공식 구성요소가 규칙이나 템플릿에 사용되고 있을 때 해당 규칙이나 템플릿에서 더 이상 사용할 필요가 없으면 규칙이나 템플릿에서 공식 구성요소를 제거한 다음 삭제할 수 있습니다. 또한 규칙이나 템플릿을 삭제할 수도 있으며 이 경우 그 안의 공식 구성요소도 삭제됩니다.

 주:

공식 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

공식 구성요소를 삭제하려면 다음을 수행합니다.

1. 시스템 뷰에서 공식을 마우스 오른쪽 버튼으로 누르고 사용 표시를 선택합니다.  
사용 중인 경우에는 해당 규칙이나 템플릿에서 공식 구성요소를 제거해야 합니다.
2. 규칙 또는 템플릿에서 공식 구성요소를 사용하고 있지 않은지 확인하십시오.

순서도에서 구성요소 제거를 참조하십시오.

3. 공식을 다시 마우스 오른쪽 버튼으로 누르고 **삭제**를 선택합니다.

## 공식 구성요소 복사 및 붙여넣기

규칙이나 템플릿에서 공식 구성요소를 복사하여 같거나 다른 규칙 또는 템플릿에 붙여넣을 수 있습니다. 또한 공식 구성요소에서 조건 그리드의 콘텐츠를 복사하여 같거나 다른 공식 구성요소에 붙여넣을 수 있습니다. 공식 구성요소를 복사하여 다른 공식 구성요소나 다른 구성요소 유형에 붙여넣을 수는 없습니다.

### 주:

공식 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

공식 구성요소를 복사하고 붙여넣으려면 다음을 수행합니다.

1. **시스템 뷰**에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.

2. 규칙 디자이너 또는 템플릿 디자이너 순서도에서 공식 구성요소를 마우스 오른쪽 버튼으로 누르고 **복사**를 선택합니다.

복사할 구성요소가 공유된 경우 공식 구성요소를 마우스 오른쪽 버튼으로 누르고 참조 복사를 선택하여 구성요소 자체가 아니라 공유 구성요소에 대한 참조를 복사할 수 있습니다. [비즈니스 규칙 공식 또는 스크립트 구성요소에 대한 참조 복사 및 붙여넣기](#)를 참조하십시오.

복사할 구성요소에 다른 구성요소(즉, 하위 멤버)가 포함된 경우 Ctrl+누르기를 사용하여 모두 선택하고 마우스 오른쪽 버튼으로 누른 다음 [그룹 복사]를 선택하여 그룹에 포함된 모든 구성요소를 복사할 수 있습니다. [순서도에서 구성요소 그룹 복사하여 붙여넣기](#)를 참조하십시오.

3. 다음 태스크 중 하나를 수행합니다.

- 공식 구성요소를 **동일한** 비즈니스 규칙 또는 템플릿 순서도에 붙여넣으려면 순서도에서 공식 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
- 공식 구성요소를 **다른** 비즈니스 규칙 또는 템플릿 순서도에 붙여넣으려면 구성요소를 붙여넣으려는 비즈니스 규칙 또는 템플릿을 열고, 순서도에서 공식 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누른 다음 **붙여넣기**를 선택합니다.

4.  를 누릅니다.

## 스크립트 구성요소

스크립트 구성요소는 비즈니스 규칙과 템플릿에서 사용할 수 있습니다.

참조:

- [스크립트 구성요소 생성](#)
- [스크립트 구성요소 디자인](#)
- [스크립트 구성요소 열기](#)



- 스크립트 구성요소 편집
- 스크립트 구성요소 삭제
- 스크립트 구성요소 복사 및 붙여넣기

## 스크립트 구성요소 생성

시스템 뷰에서 스크립트 구성요소를 생성할 수 있습니다. 스크립트 구성요소는 비즈니스 규칙과 템플릿에 사용할 수 있습니다.

### 주:

스크립트 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

스크립트 구성요소를 생성하려면 다음을 수행합니다.

1. 스크립트의 이름 및 애플리케이션 유형을 입력합니다.
2. **애플리케이션 이름**을 선택합니다.  
애플리케이션 이름은 Financial Management 또는 Planning 애플리케이션 관리에서 생성된 애플리케이션이거나 Essbase 블록 저장영역 애플리케이션이어야 합니다.
3. 다음 중 한 가지 태스크를 수행합니다.
  - a. Financial Management를 선택한 경우 **계산 유형**을 선택합니다.
  - b. Planning을 선택한 경우 **계획 유형**을 선택합니다.
  - c. Essbase를 선택한 경우 **데이터베이스**를 선택합니다.

### 주:

시스템 뷰에서 스크립트를 마우스 오른쪽 버튼으로 누르고 [새로 작성]을 선택하여 새 스크립트를 생성하면 [새 스크립트] 대화상자에 작업 중인 애플리케이션 유형, 애플리케이션 및 계산 유형(Financial Management 스크립트), 계획 유형(Planning 스크립트) 또는 데이터베이스(Essbase 스크립트)가 입력됩니다.

## 스크립트 구성요소 디자인

스크립트 구성요소는 비즈니스 규칙과 템플릿에 사용할 수 있습니다. 시스템 뷰나 다른 모든 뷰에서 스크립트 구성요소를 생성합니다. 비즈니스 규칙 또는 템플릿을 디자인하는 동안 규칙 또는 템플릿 디자이너에서 스크립트 구성요소를 생성할 수도 있습니다.

 주:

스크립트 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

 주:

비즈니스 규칙 또는 템플릿에서 스크립트 구성요소를 디자인할 수도 있습니다.

스크립트 구성요소를 디자인하려면 다음을 수행합니다.

1. 시스템 뷰에서 스크립트를 마우스 오른쪽 버튼으로 누르고 **새로 작성**을 선택합니다.
2. 스크립트의 이름, 애플리케이션 유형 및 애플리케이션을 입력합니다. 그런 다음 다음 태스크 중 하나를 수행합니다.
  - **Financial Management 사용자만 해당:** Financial Management 스크립트를 생성하는 경우 계산 유형을 입력합니다.
  - **Planning 사용자만 해당:** Planning 스크립트를 생성하는 경우 계획 유형을 입력합니다.
  - **Essbase 사용자만 해당:** Essbase 스크립트를 생성하는 경우 데이터베이스를 입력합니다.
3. **확인**을 누릅니다.
4. 아이콘을 사용하여 스크립트 구성요소를 디자인합니다.

Financial Management 스크립트는 Visual Basic으로 작성해야 하고 Planning 및 Essbase 스크립트는 Essbase로 작성해야 합니다.

 주:

규칙, 템플릿 및 기타 구성요소와 달리, 스크립트를 열 때는 그래픽 형식으로 (순서도 내) 볼 수 없습니다. 스크립트 형식으로만 볼 수 있습니다.

스크립트를 사용하여 다음 태스크를 수행할 수 있습니다.

- **라인 번호 숨기기** 아이콘을 눌러 스크립트 라인 번호를 숨기거나 표시합니다. 라인 번호는 기본적으로 표시됩니다.
- **함수와 매개변수 삽입** 아이콘을 눌러 스크립트에 함수를 삽입합니다. **함수 작업**을 참조하십시오.
- **차원에서 선택한 멤버 삽입** 아이콘을 눌러 스크립트에 멤버를 삽입합니다. **구성요소에 멤버 및 함수 추가**을 참조하십시오.
- **변수 삽입** 아이콘을 눌러 스크립트에 변수를 삽입합니다. **변수 작업**을 참조하십시오.

 주:

Visual Basic 스크립트는 필요한 모든 내용을 자체적으로 포함하고 있어야 합니다. 스크립트에서 규칙 세트에 정의된 대체 또는 실행 변수를 참조할 수 없습니다. 스크립트에서 변수를 사용해야 하는 경우 스크립트 구성요소 내에서 선언해야 합니다.

- **Planning 사용자만 해당:** 스마트 목록 삽입을 눌러 스크립트에 스마트 목록을 삽입합니다. [스마트 목록 작업](#)을 참조하십시오.
- 설명 및 설명 제거 아이콘을 눌러 스크립트에서 설명 라인을 추가하고 제거합니다.
- 설명 숨기기 아이콘을 눌러 스크립트에서 시스템 생성 설명을 숨깁니다.
- 구문 확인 아이콘을 눌러 스크립트 구문에 오류가 있는지 확인합니다.

 주:

구문 오류가 있으면 오류가 포함된 코드가 빨간색 텍스트로 바뀌며, 오류 메시지는 표시되지 않습니다. 구문 오류가 없으면 "발견된 구문 오류가 없습니다."라는 메시지가 표시됩니다.

- 페이지 바깥쪽으로 스크롤되는 스크립트의 긴 라인을 동일한 페이지의 여러 라인에 표시하려면 [줄 바꿈](#) 아이콘을 누릅니다.
  - 코드 완료 OFF 설정 아이콘을 눌러 코드를 완료하기 위한 제안을 제외합니다. 이 아이콘은 토글로 작동합니다. 코드 완료를 다시 켜려면 아이콘을 다시 누릅니다. 코드 완료는 기본적으로 ON으로 설정되어 있습니다.
  - 바꾸기 아이콘을 눌러 스크립트에서 텍스트 문자열을 [찾아서 바꿉니다](#). [비즈니스 규칙 스크립트에서 텍스트 문자열 검색](#)을 참조하십시오.
  - 찾기 아이콘을 눌러 스크립트에서 텍스트 문자열을 [찾습니다](#). [비즈니스 규칙 스크립트에서 텍스트 문자열 검색](#)을 참조하십시오.
  - 찾기에 검색 텍스트를 입력하고 이전 또는 다음을 눌러 텍스트의 이전(스크립트에서 위로 검색) 인스턴스 또는 다음(스크립트에서 아래로 검색) 인스턴스를 찾습니다.
  - 행으로 이동에 라인 번호를 입력하고 화살표를 눌러 스크립트에서 라인 번호를 강조 표시합니다.
  - 라인 왼쪽에 있는 더하기 또는 빼기 기호를 눌러 스크립트에서 라인을 확장 또는 축소합니다. 예를 들어 FIX 명령문은 표시될 때 기본적으로 명령문의 모든 라인이 표시되지만 라인 왼쪽에 있는 빼기 기호를 눌러 FIX 명령문의 라인을 숨길 수 있습니다.
  - 복사할 스크립트 라인을 선택하고 [클립보드에 복사](#) 아이콘을 누릅니다.
  - 잘라낼 스크립트 라인을 선택하고 [클립보드에 잘라내기](#) 아이콘을 누릅니다.
  - 붙여넣을 스크립트 위치에 커서를 놓고 [클립보드에서 붙여넣기](#) 아이콘을 누릅니다.
  - 함수에 대해 자동 제안이 제공됩니다. 함수의 처음 몇 글자를 입력하고 **Ctrl+스페이스**를 누르면 제안이 표시됩니다. 제안 중 하나를 선택하고 **Enter** 키를 눌러 함수를 완성합니다.
5. 등록정보에서 다음 단계를 완료합니다.

- a. 스크립트 구성요소를 생성할 때는 **공유**가 선택되지 않습니다. 스크립트 구성요소를 공유로 설정하려면 **등록정보** 탭, **공유** 순으로 선택하고 이름을 입력합니다.

시스템 뷰에서 스크립트 구성요소를 생성할 때는 **공유**가 기본적으로 선택되며 변경할 수 없습니다. 대신, 비즈니스 규칙에서 스크립트를 복사하려는 경우 공유 스크립트를 비즈니스 규칙으로 끌어온 다음 **등록정보** 탭에서 **공유** 확인란의 선택을 취소합니다. 그러면 비즈니스 규칙에서 공유 스크립트 복사본이 생성됩니다. 공유 스크립트를 변경해도 공유 스크립트 복사본은 변경되지 않습니다. 비즈니스 규칙을 저장하면 스크립트에 더 이상 이름이 없습니다.

- b. **선택 사항:** 최대 50자의 새 이름을 입력하여 스크립트 이름을 편집합니다. 이름은 기본적으로 [새 스크립트] 대화상자에서 가져옵니다.

 **주:**

공유 스크립트에만 이름을 지정할 수 있습니다. 스크립트가 공유되지 않는 경우에는 [이름] 필드에 입력할 수 없습니다.

- c. 스크립트에 대한 설명을 최대 255자까지 입력합니다.
  - d. 스크립트에 대한 캡션을 입력합니다. 캡션은 **규칙 디자이너** 및 **템플릿 디자이너** 순서도에서 스크립트 아래에 표시됩니다.
  - e. 스크립트에 대한 설명을 입력합니다. 예를 들어 스크립트의 용도를 알려 줄 수 있습니다.
6. **Financial Management 사용자만 해당: 타이머 사용**을 선택합니다. 그러면 규칙이 실행될 때 스크립트를 처리하는 데 걸린 시간이 로그 파일에 기록됩니다. 규칙, 규칙 세트 및 구성요소에 대해 타이머를 사용할 수 있습니다. 타이머가 활성화된 모든 객체에 대한 처리 시간이 로그 파일에 포함됩니다. 예를 들어 타이머가 사용되는 규칙이 있고 이 규칙에 타이머가 사용되는 스크립트 세 개가 포함된 경우 규칙과 규칙의 각 스크립트를 처리하는 데 걸린 시간이 기록됩니다.
7. **Financial Management 사용자만 해당:** Visual Basic 스크립트를 생성할 때 스크립트 구성요소를 사용하는 규칙 내에 스크립트 구성요소 콘텐츠를 포함하도록 **포함**을 선택합니다.
8. **Financial Management 사용자만 해당: 매개변수**에서 스크립트 매개변수를 입력합니다.
9. **Financial Management, Planning 및 Essbase 블록 저장영역 사용자만 해당:** 스크립트에 스크립트 구성요소의 캡션을 입력합니다.
10. **템플릿에 대한 스크립트 구성요소를 생성하는 사용자만 해당:** 스크립트 구성요소에서 디자인 타임 프롬프트를 사용하려는 경우 **디자인 프롬프트 사용**을 선택합니다. 그런 다음 조건 그리드에서 다음 태스크를 수행하여 디자인 타임 프롬프트에 대한 조건을 정의합니다.
- a. **DTP**에서 디자인 타임 프롬프트를 선택합니다.
  - b. **연산자**에서 연산자 = = 또는 <>를 선택합니다.
  - c. **값**의 드롭다운 목록을 사용하여 값을 선택합니다. 이 필드에는 입력할 수 없습니다.
  - d. 이 단계를 반복하여 조건에 필요한 만큼 명령문을 생성합니다.

 **팁:**

더하기(+) 및 빼기(-) 아이콘을 눌러 그리드에 행을 추가 및 삭제합니다.

11. 독립 스크립트 구성요소(즉, 비즈니스 규칙이나 템플릿에 포함되지 않은 스크립트 구성요소) 또는 비즈니스 규칙에 대한 스크립트 구성요소를 생성하는 사용자만 해당: 사용에서 스크립트 구성요소를 사용하는 규칙을 볼 수 있습니다.

 주:

이 탭의 정보는 편집할 수 없습니다.

12.  를 누릅니다.

## 스크립트 구성요소 열기

시스템 뷰에서 또는 규칙 디자이너나 템플릿 디자이너의 순서도에서 스크립트 구성요소를 열 수 있습니다.

 주:

스크립트 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

스크립트 구성요소를 열려면 다음을 수행합니다.

1. 시스템 뷰에서 스크립트를 확장합니다.
2. 스크립트를 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.

 주:

공유 스크립트 구성요소가 비즈니스 규칙에 사용되는 경우 스크립트 구성요소를 마우스 오른쪽 버튼으로 누르고 [열기]를 선택하거나 스크립트 구성요소를 두 번 눌러 규칙 디자이너의 규칙 순서도에서 열 수도 있습니다.

## 스크립트 구성요소 편집

스크립트 구성요소를 작성하는 데 사용하는 함수, 변수 및 멤버를 편집할 수 있고, 스크립트 구성요소의 모든 공식 명령문 계산 결과를 로그 파일에 포함할지 여부(Oracle Hyperion Financial Management 애플리케이션만 해당) 및 스크립트 구성요소의 설명(comment), 캡션, 이름, 설명(description)을 편집할 수 있습니다.

 주:


스크립트 구성요소는 Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

스크립트 구성요소를 편집하려면 다음을 수행합니다.

1. 시스템 뷰에서 스크립트를 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.

 주:

규칙, 템플릿 및 기타 구성요소와 달리, 스크립트를 열 때는 그래픽 형식으로 (순서도 내) 볼 수 없습니다. 스크립트 형식으로만 볼 수 있습니다.

2. 원하는 스크립트 등록정보를 편집하고  를 누릅니다.

[스크립트 구성요소 디자인](#)을 참조하십시오.

## 스크립트 구성요소 삭제

규칙이나 템플릿에서 사용되고 있지 않은 스크립트 구성요소만 삭제할 수 있습니다. 규칙이나 템플릿에 스크립트 구성요소가 사용되고 있는지 확인하려면 사용을 표시하면 됩니다. [공식 및 스크립트 구성요소의 사용 표시](#)를 참조하십시오. 스크립트 구성요소가 규칙이나 템플릿에 사용되고 있을 때 해당 규칙이나 템플릿에서 더 이상 사용할 필요가 없으면 규칙이나 템플릿에서 스크립트 구성요소를 제거한 다음 삭제할 수 있습니다. 또한 규칙이나 템플릿을 삭제할 수도 있으며 이 경우 그 안의 스크립트 구성요소도 삭제됩니다.

 주:

스크립트 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

스크립트 구성요소를 삭제하려면 다음을 수행합니다.

1. 시스템 뷰에서 스크립트를 마우스 오른쪽 버튼으로 누르고 **사용 표시**를 선택합니다.
2. 규칙 또는 템플릿에서 스크립트 구성요소를 사용하고 있지 않은지 확인하십시오.  
사용 중인 경우에는 해당 규칙이나 템플릿에서 스크립트 구성요소를 제거해야 합니다.  
[순서도에서 구성요소 제거](#)를 참조하십시오.
3. 스크립트를 다시 마우스 오른쪽 버튼으로 누르고 **삭제**를 선택합니다.

## 스크립트 구성요소 복사 및 붙여넣기

규칙이나 템플릿에서 스크립트 구성요소를 복사하여 같거나 다른 규칙 또는 템플릿에 붙여넣을 수 있습니다. [스크립트 구성요소 내의 스크립트](#)를 복사하여 같거나 다른 스크립트 구성요소에

붙여넣을 수도 있습니다. 스크립트 구성요소를 복사하여 다른 스크립트 구성요소나 다른 구성요소 유형에 붙여넣을 수는 없습니다.

 주:


스크립트 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

스크립트 구성요소를 복사하고 붙여넣으려면 다음을 수행합니다.

1. 시스템 뷰에서 스크립트 구성요소가 포함된 규칙 또는 템플릿을 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너 순서도에서 스크립트 구성요소를 마우스 오른쪽 버튼으로 누르고 **복사**를 선택합니다.

 주:

복사할 구성요소가 공유된 경우 스크립트 구성요소를 마우스 오른쪽 버튼으로 누르고 참조 복사를 선택하여 구성요소 자체가 아니라 공유 구성요소에 대한 참조를 복사할 수 있습니다. [비즈니스 규칙 공식 또는 스크립트 구성요소에 대한 참조 복사 및 붙여넣기](#)를 참조하십시오.

3. 다음 태스크 중 하나를 수행합니다.
  - 스크립트 구성요소를 *동일한* 비즈니스 규칙 또는 템플릿에 붙여넣으려면 순서도에서 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
  - 스크립트 구성요소를 *다른* 비즈니스 규칙 또는 템플릿에 붙여넣으려면 스크립트 구성요소를 붙여넣으려는 비즈니스 규칙 또는 템플릿을 열고, 순서도에서 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누른 다음 **붙여넣기**를 선택합니다.
4.  를 누릅니다.

## 조건 구성요소

조건 구성요소는 true 또는 false인 조건문으로 구성됩니다.

**참조:**

- [조건 구성요소 정보](#)
- [조건 구성요소 생성](#)
- [조건 구성요소 열기](#)
- [조건 구성요소 편집](#)
- [조건 구성요소 삭제](#)
- [조건 구성요소 복사 및 붙여넣기](#)

## 조건 구성요소 정보

조건 구성요소는 true 또는 false인 조건문(즉, IF...THEN 문)으로 구성됩니다. 조건이 true이면 시스템은 사용자가 지정한 작업을 수행하고, 조건이 false이면 사용자가 지정한 다른 작업을 수행합니다. 조건은 메타데이터 조건이거나 데이터 조건일 수 있습니다. 조건 구성요소는 공유할 수 없습니다.

### 주:

조건 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

## 조건 구성요소 생성

비즈니스 규칙 또는 템플릿에서 조건 구성요소를 생성합니다. 스크립트 및 공식 구성요소와 달리, 조건 구성요소는 독립 객체로 생성할 수 없습니다. 해당 객체를 생성하는 비즈니스 규칙이나 템플릿에 링크되며, 공유될 수 없습니다.

### 주:

조건 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.


조건 구성요소를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 조건을 생성할 순서도의 위치를 결정한 후 새 객체 팔레트에서 조건 구성요소를 순서도의 해당 위치로 끌어 놓습니다.  
조건 객체는 순서도에서 물음표가 있는 다이아몬드 모양으로 표시됩니다.
3. **선택 사항:** 조건에, 조건 구성요소를 확인할 캡션을 입력합니다. 캡션은 해당 조건을 사용하는 모든 규칙 또는 템플릿의 순서도에서 구성요소 위에 표시됩니다.
4. **템플릿에 대한 조건 구성요소를 생성하는 사용자만 해당:** 조건 구성요소에서 디자인 타임 프롬프트를 사용하려는 경우 디자인 프롬프트 사용을 선택합니다. 그런 다음 조건 그리드에서 다음 태스크를 수행하여 디자인 타임 프롬프트에 대한 조건을 정의합니다.
  - a. DTP에서 디자인 타임 프롬프트를 선택합니다.
  - b. 연산자에서 연산자 == 또는 <>를 선택합니다.
  - c. 값의 드롭다운 목록을 사용하여 값을 선택합니다. 이 필드에는 입력할 수 없습니다.
  - d. 이 단계를 반복하여 조건에 필요한 만큼 명령문을 생성합니다.



 주:

디자인 타임 프롬프트를 사용하여 조건을 정의하는 경우 조건 상자에 조건을 입력하거나 조건 빌더를 사용하여 조건을 빌드할 수 없습니다.

5. **Financial Management 사용자만 해당: 타이머 사용**을 선택합니다. 그러면 비즈니스 규칙이 실행될 때 조건 구성요소를 처리하는 데 걸린 시간이 로그 파일에 기록됩니다. 규칙, 규칙 세트 및 구성요소에 대해 타이머를 사용할 수 있습니다. 타이머가 활성화된 모든 객체에 대한 처리 시간이 로그 파일에 포함됩니다. 예를 들어 타이머가 사용되는 규칙이 있고 이 규칙에 타이머가 사용되는 조건 구성요소 세 개가 포함된 경우 규칙과 규칙의 각 조건 구성요소를 처리하는 데 걸린 시간이 기록됩니다.
6. **Planning 및 Essbase 사용자만 해당: 멤버 블록**을 생성하려면 **생략 기호** 아이콘을 누르고 **멤버** 또는 **함수**를 선택합니다. 기본적으로 조건 구성요소 계산 처리는 사용자화 그리드에 입력하는 첫 번째 멤버에서 시작됩니다. 다른 멤버에서 처리를 시작하려면 **멤버 블록**에 **멤버** 또는 **함수** 이름을 입력하거나 **출입표** 아이콘을 눌러 멤버 선택기에서 하나를 선택합니다. **구성요소에 멤버 및 함수 추가 정보**를 참조하십시오.
7. 다음 태스크 중 하나를 수행합니다.
  - [조건] 텍스트 상자에 조건문을 입력합니다.
  - **조건 추가**를 눌러 조건 빌더를 사용하여 조건문을 생성합니다. **조건 빌더를 사용하여 조건문 생성**을 참조하십시오.
8. 조건 구성요소에 대한 설명을 입력하고  를 누릅니다.

## 조건 구성요소 열기

조건 구성요소는 해당 구성요소가 속한 비즈니스 규칙 또는 템플리트의 순서도에서 엽니다. 공식 및 스크립트 구성요소와 달리 시스템 뷰에서는 열 수 없습니다.

 주:

조건 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플리트에서 사용할 수 있습니다.

조건 구성요소를 열려면 다음을 수행합니다.

1. **시스템 뷰**에서 규칙 또는 템플리트를 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙이나 템플리트가 열려 있는 경우 순서도의 조건 구성요소를 선택하여 조건 등록정보를 표시합니다.


## 조건 구성요소 편집

조건 구성요소를 생성할 때 사용하는 함수, 변수 및 멤버를 편집할 수 있고, 조건 구성요소 처리에 걸린 시간을 로그 파일에 기록할지 여부(Oracle Hyperion Financial Management 애플리케이션만 해당) 및 조건 구성요소의 설명과 캡션을 편집할 수 있습니다.

 주:

조건 구성요소는 Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

조건 구성요소를 편집하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 [열기]를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너의 조건에서 조건을 선택하여 해당 등록정보를 편집합니다. 편집할 수 있는 등록정보는 다음과 같습니다. [조건 구성요소 생성](#)을 참조하십시오.
  - 캡션
  - 조건문
  - 설명
  - **템플릿 사용자만 해당:** 디자인 설계 프롬프트
  - **Financial Management 사용자만 해당:** 로그 및 타이머 옵션
3.  를 누릅니다.


## 조건 구성요소 삭제

조건 구성요소를 해당 구성요소가 속한 비즈니스 규칙이나 템플릿에서 제거하여 삭제합니다. 조건 구성요소는 하나의 비즈니스 규칙 또는 템플릿에서만 사용될 수 있으므로 해당 비즈니스 규칙이나 템플릿에서 제거하면 삭제됩니다.

 주:

조건 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

조건 구성요소를 삭제하려면 다음을 수행합니다.

1. 시스템 디자이너에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너의 순서도에서 삭제할 조건 구성요소를 선택합니다.
3. 조건을 마우스 오른쪽 버튼으로 누르고 **제거**를 선택합니다.  
조건 구성요소는 공유할 수 없으므로 비즈니스 규칙이나 템플릿에서 조건 구성요소를 제거하면 데이터베이스에서 삭제됩니다.
4. 구성요소 삭제를 확인하고  를 누릅니다.


## 조건 구성요소 복사 및 붙여넣기

규칙이나 템플릿에서 조건 구성요소를 복사하여 같거나 다른 규칙 또는 템플릿에 붙여넣을 수 있습니다. 조건 구성요소를 복사하여 다른 조건 구성요소나 다른 구성요소 유형에 붙여넣을 수는 없습니다.

### 주:

조건 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

규칙 디자이너 또는 템플릿 디자이너에서 조건 구성요소를 복사하여 붙여넣으려면 다음을 수행합니다.

1. **시스템 뷰**에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너 순서도에서 복사할 조건 구성요소를 마우스 오른쪽 버튼으로 누르고 **복사**를 선택합니다.
3. 다음 태스크 중 하나를 수행합니다.
  - 조건 구성요소를 **동일한** 비즈니스 규칙 또는 템플릿에 붙여넣으려면 순서도에서 조건 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
  - 조건 구성요소를 **다른** 비즈니스 규칙 또는 템플릿에 붙여넣으려면 구성요소를 붙여넣으려는 비즈니스 규칙 또는 템플릿을 열고, 순서도에서 조건 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누른 다음 **붙여넣기**를 선택합니다.
4.  를 누릅니다.

## 멤버 블록 구성요소

멤버 블록 구성요소는 스크립트에서 하나 이상의 명령문을 둘러쌀 멤버를 정의합니다.

### 참조:

- [멤버 블록 구성요소 정보](#)
- [멤버 블록 구성요소 생성](#)
- [멤버 블록 구성요소 열기](#)
- [멤버 블록 구성요소 편집](#)
- [멤버 블록 구성요소 삭제](#)
- [멤버 블록 구성요소 복사 및 붙여넣기](#)

## 멤버 블록 구성요소 정보

멤버 블록 구성요소는 스크립트에서 하나 이상의 명령문을 둘러쌀 멤버를 정의합니다. 예를 들어, If 조건은 멤버 블록으로 둘러싸야 합니다. 멤버 블록은 비즈니스 규칙 및 사용자정의 템플릿 둘 다에서 사용할 수 있습니다.

멤버 블록은 공식의 타겟에 Oracle Essbase 대체 변수 또는 횡차원 멤버를 사용하는 경우에 필요합니다. 멤버가 공식 대상의 연속된 명령문에서 반복되는 경우에도 멤버 블록을 사용할 수 있습니다.

멤버 블록 구성요소는 이 구성요소가 생성되는 비즈니스 규칙이나 템플릿과 별개인 데이터베이스 객체로 존재하지 않고, 구성요소가 속한 템플릿나 비즈니스 규칙의 일부로만 존재합니다. 멤버 블록은 규칙과 템플릿 사이에서 공유할 수 없습니다.


### 주:

멤버 블록 구성요소는 Oracle Hyperion Planning 및 Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

## 멤버 블록 구성요소 생성

멤버 블록 구성요소는 규칙이나 템플릿을 디자인할 때 규칙 또는 템플릿 디자이너에서 생성합니다. 시스템 뷰에서는 멤버 블록 구성요소를 생성할 수 없습니다.

멤버 블록 구성요소를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 멤버 블록을 생성할 순서도 위치를 결정한 후 새 객체에서 [멤버 블록] 객체를 순서도로 끌어서 놓습니다.  
멤버 블록 객체는 순서도에서 연결선이 있는 4개 블록으로 표시됩니다.
3. 선택 사항: 템플릿에 대한 멤버 블록 구성요소를 생성하는 경우 해당 디자인 타임 프롬프트를 생성할 수도 있습니다. 사용자정의 템플릿에 대한 디자인 타임 프롬프트 생성을 참조하십시오.
4. 멤버 블록 탭에서 멤버 블록 옆에 있는 작업 아이콘을 누르고 다음 중 하나를 수행합니다.
  - 변수를 선택하여 변수를 검색하거나 생성하고 확인을 누릅니다. 변수 작업을 참조하십시오.
  - 멤버를 선택하여 멤버를 검색하고 확인을 누릅니다. 구성요소에 멤버 및 함수 추가 정보를 참조하십시오.
5. 오류 및 경고 탭에서 버튼을 눌러 스크립트 진단을 실행합니다.
6. 등록정보에서 사용 안 함을 선택하여 검증 및 계산에서 멤버 블록을 제외합니다. 그런 다음 선택적 설명(description) 및 설명(comment)을 입력합니다.
7.  를 누릅니다.

## 멤버 블록 구성요소 열기

멤버 블록 구성요소는 이 구성요소가 사용되는 비즈니스 규칙 또는 템플리트의 순서도에서 열립니다. 시스템 뷰에서는 열 수 없습니다.

### 주:

멤버 블록 구성요소는 Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플리트에서 사용할 수 있습니다.

멤버 블록 구성요소를 열려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플리트를 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙이나 템플리트가 열려 있는 경우 순서도의 멤버 블록 구성요소를 선택하여 멤버 블록 등록정보를 표시합니다.

## 멤버 블록 구성요소 편집

멤버 블록의 멤버를 변경할 수 있고, 멤버 블록 구성요소 처리에 걸리는 시간이 로그 파일에 기록되는지 여부(Oracle Hyperion Financial Management 애플리케이션 사용자만 해당) 및 멤버 블록 구성요소의 디자인 타임 프롬프트(사용자정의 템플리트 사용자만 해당)를 변경할 수 있습니다.

### 주:

멤버 블록 구성요소는 Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플리트에서 사용할 수 있습니다.

멤버 블록 구성요소를 편집하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플리트를 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙 디자이너 또는 템플리트 디자이너의 멤버 블록에서 순서도의 멤버 블록을 선택하여 해당 등록정보를 편집합니다.

멤버 블록에 대해 선택한 멤버 또는 변수, 멤버 블록에 대한 설명(description) 및 설명(comment), 멤버 블록이 비즈니스 규칙이나 템플리트의 검증 및 계산에 포함되지 않도록 비활성화할지 여부를 변경할 수 있습니다.

3.  를 누릅니다.

## 멤버 블록 구성요소 삭제

멤버 블록 구성요소를 이 구성요소가 속한 비즈니스 규칙이나 템플리트에서 제거하여 삭제합니다. 멤버 블록 구성요소는 하나의 비즈니스 규칙 또는 템플리트에서만 사용할 수 있습니다. 다른 비즈니스 규칙이나 템플리트 사이에서는 공유할 수 없습니다.

 주:

멤버 블록 구성요소는 Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

멤버 블록 구성요소를 삭제하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너의 순서도에서 삭제할 멤버 블록 구성요소를 선택합니다.
3. 멤버 블록을 마우스 오른쪽 버튼으로 누르고 제거를 선택합니다.

멤버 블록은 멤버 블록이 생성된 비즈니스 규칙이나 템플릿에만 있으므로 멤버 블록을 제거하면 데이터베이스에서 삭제됩니다.


## 멤버 블록 구성요소 복사 및 붙여넣기

규칙 또는 템플릿의 멤버 블록 구성요소를 복사하여 같거나 다른 규칙 또는 템플릿에 붙여넣을 수 있습니다. 멤버 블록 구성요소를 복사하여 다른 멤버 블록 구성요소나 다른 구성요소 유형에 붙여넣을 수는 없습니다.

 주:

멤버 블록 구성요소는 Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

규칙 디자이너 또는 템플릿 디자이너에서 멤버 블록 구성요소를 복사하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너 순서도에서 복사할 멤버 블록 구성요소를 마우스 오른쪽 버튼으로 누르고 복사를 선택합니다.
3. 다음 태스크 중 하나를 수행합니다.
  - 멤버 블록 구성요소를 같은 비즈니스 규칙 또는 템플릿에 붙여넣으려면 해당 구성요소를 붙여넣을 순서도의 위치를 마우스 오른쪽 버튼으로 누르고 붙여넣기를 선택합니다.
  - 멤버 블록 구성요소를 다른 비즈니스 규칙 또는 템플릿에 붙여넣으려면 구성요소를 붙여넣을 비즈니스 규칙이나 템플릿을 열고 멤버 블록 구성요소를 붙여넣을 순서도의 위치를 마우스 오른쪽 버튼으로 누른 다음 붙여넣기를 선택합니다.
4.  를 누릅니다.

## 멤버 범위 구성요소

멤버 범위 구성요소는 Planning 차원의 멤버 범위로 구성된 루프 유형입니다.

참조:

- [멤버 범위 구성요소 정보](#)

- 멤버 범위 구성요소 생성
- 멤버 범위 구성요소 열기
- 멤버 범위 구성요소 편집
- 멤버 범위 구성요소 삭제
- 멤버 범위 구성요소 복사 및 붙여넣기

## 멤버 범위 구성요소 정보

멤버 범위 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 또는 Oracle Essbase 차원의 멤버 범위로 구성된 일종의 루프입니다. 멤버 범위 구성요소는 공유할 수 없으므로 비즈니스 규칙이나 템플릿에 추가할 때마다 새 멤버 범위 구성요소를 생성해야 합니다.



주:

멤버 범위 구성요소는 Financial Management, Planning 및 Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

## 멤버 범위 구성요소 생성

규칙 또는 템플릿을 디자인할 때 규칙 또는 템플릿 디자이너에서 멤버 범위 구성요소를 생성합니다. 사용되는 비즈니스 규칙 및 템플릿과 독립적으로 존재하는 스크립트 및 공식 구성요소와 달리, 멤버 범위 구성요소는 시스템 뷰에서 생성할 수 없습니다. 멤버 범위 구성요소는 해당 구성요소가 속한 비즈니스 규칙 및 템플릿에 링크되며 공유할 수 없습니다.

멤버 범위 구성요소를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 새 객체에서 멤버 범위 객체를 순서대로 끌어서 놓습니다.  
멤버 범위 객체는 연결선이 있는 두 원으로 표시됩니다.
3. 사용자정의 템플릿 사용자만 해당: 템플릿에 대한 멤버 범위 구성요소를 생성하는 경우 디자인 타임 프롬프트 탭에서 해당 디자인 타임 프롬프트를 생성합니다.  
[사용자정의 템플릿에 대한 디자인 타임 프롬프트 생성](#)을 참조하십시오.
4. 멤버 범위 탭에서 다음 중 하나를 수행합니다.
  - 변수 선택기를 선택하고 변수를 사용하여 멤버 범위를 정의합니다. 그런 다음 변수를 입력하거나 선택합니다. [변수 작업](#)을 참조하십시오.

 주:

이 옵션을 선택하고 멤버를 변경하면 "그리드를 편집하면 동적 변수 링크가 제거됩니다. 계속하시겠습니까?"라는 메시지가 나타납니다. 변수에 대한 링크를 제거하려면 [예]를 누릅니다. [예]를 누르면 변수에 대한 링크가 제거되고 변수에 있던 멤버를 포함하는 멤버 범위는 유지됩니다. 변수에 대한 링크를 유지하려면 [아니오]를 누릅니다.

이 옵션을 선택한 경우 변수가 변경될 때마다 규칙 또는 템플리트의 변수가 이러한 변경과 함께 업데이트되도록 **동적으로 변수 링크**를 선택합니다.

- **Oracle Hyperion Planning 및 Oracle Essbase 사용자만 해당:**

- a. 멤버 범위를 정의할 각 차원의 **값** 열에 멤버를 입력하거나, 각 행을 눌러 **작업** 아이콘을 표시합니다.

 **팁:**

한 번에 한 행씩 각 차원에 대한 멤버를 선택하는 대신 [멤버 선택기] 아이콘을 눌러 그리드에서 모든 차원에 대한 멤버를 선택할 수 있습니다. [멤버 선택기]에서 [확인]을 누르면 선택된 멤버가 멤버를 선택한 그리드의 각 차원에 대해 표시됩니다.


- b. **작업** 아이콘을 누르고 다음 옵션 중 하나를 선택하여 멤버 범위를 정의합니다.
  - 변수([변수 작업](#) 참조)
  - 멤버([구성요소에 멤버 및 함수 추가](#) 참조)
  - 함수([함수 작업](#) 참조)
- c. **설명** 아이콘을 누르고 멤버 범위에 대한 설명을 입력합니다.
- d. 멤버 범위를 정의할 각 차원에 대해 이 단계를 반복합니다.
- e. **그리드 값 제외**를 선택하여 선택한 멤버를 계산에서 제외합니다.
- f. **그리드 재설정**을 눌러 그리드에 입력한 멤버, 변수 및 함수를 지웁니다.
- g. **병렬 처리 사용**을 선택하여 멤버 범위 명령을 순차적으로 처리하는 대신 동시에 처리합니다. 기본적으로 직렬 처리가 사용되지만 병렬 처리를 사용하면 계산을 최적화할 수 있습니다. 그런 다음 텍스트 상자에 병렬 처리에 사용할 수 있는 스레드 수를 입력하거나 **변수 선택기**를 눌러 숫자 또는 정수 유형 변수(비즈니스 규칙에서 멤버 범위를 생성하는 경우) 또는 숫자 디자인 타임 프롬프트(템플리트에 대한 멤버 범위를 생성하는 경우)를 선택합니다.

기본적으로 병렬 처리에 사용되는 스레드 수는 2개입니다. 32비트 플랫폼의 경우 1에서 4 사이의 정수를 선택합니다. 64비트 플랫폼의 경우 1에서 8 사이의 정수를 선택합니다.

- **Oracle Hyperion Financial Management 사용자만 해당:**

- a. **차원** 열 아래의 첫 번째 행을 누르고 차원을 선택합니다.
- b. **값** 열에 멤버를 입력하거나, 행을 눌러 **작업** 아이콘을 표시합니다. **작업** 아이콘을 누르고 다음 옵션 중 하나를 선택하여 멤버 범위를 정의합니다.
  - 변수([변수 작업](#) 참조)



- 멤버(구성요소에 멤버 및 함수 추가 참조)
  - 함수(함수 작업 참조)
  - c. 변수 열에 실행 변수를 입력하거나, 행을 눌러 **작업** 아이콘을 표시합니다. 작업을 누르고 실행 변수를 선택합니다. **변수 작업**을 참조하십시오.
  - d. **설명** 아이콘을 누르고 멤버 범위에 대한 설명을 입력합니다.
  - e. 멤버 범위를 정의할 각 차원에 대해 이 단계를 반복합니다.
5. **비즈니스 규칙 사용자만 해당: 오류 및 경고** 탭에서 스크립트 진단 실행 버튼을 누릅니다.
6. **선택 사항: 등록정보에서 사용 안 함**을 선택하여 멤버 범위(및 해당 범위 내의 모든 구성요소)를 검증에서 제외합니다. 비즈니스 규칙 또는 템플리트의 구성요소로 인해 비즈니스 규칙 또는 템플리트를 검증할 수 없는 경우 문제 해결을 위해 이 옵션을 사용할 수 있습니다.
- 멤버 범위에 대한 설명(description), 설명(comment) 및 캡션을 입력할 수도 있습니다. 캡션은 해당 캡션이 속한 규칙이나 템플리트의 순서도에서 구성요소 아래에 표시됩니다.
7. **Financial Management 사용자만 해당: 등록정보에서 타이머 사용**을 선택하여 비즈니스 규칙이 실행될 때 멤버 범위 구성요소를 처리하는 데 걸린 시간을 로그 파일에 기록합니다. 규칙, 규칙 세트 및 구성요소에 대해 타이머를 사용할 수 있습니다. 타이머가 활성화된 모든 객체에 대한 처리 시간이 로그 파일에 포함됩니다. 예를 들어 타이머가 사용되는 규칙이 있고 이 규칙에 타이머가 사용되는 멤버 범위 구성요소 세 개가 포함된 경우 규칙과 규칙의 각 멤버 범위 구성요소를 처리하는 데 걸린 시간이 기록됩니다.
8. 을 누릅니다.

## 멤버 범위 구성요소 열기

멤버 범위 구성요소는 해당 구성요소가 사용되는 비즈니스 규칙 또는 템플리트의 순서도에서 열 수 있습니다. 공식 및 스크립트 구성요소와 달리 시스템 뷰에서는 열 수 없습니다.

### 주:

멤버 범위 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플리트에서 사용할 수 있습니다.

멤버 범위 구성요소를 열려면 다음을 수행합니다.

1. **시스템 뷰**에서 규칙 또는 템플리트를 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙이나 템플리트가 열려 있는 경우 순서도의 멤버 범위 구성요소를 선택하여 멤버 범위 등록정보를 표시합니다.


## 멤버 범위 구성요소 편집

멤버 범위 구성요소를 정의할 때 사용하는 차원과 멤버 또는 변수를 편집할 수 있고, 멤버 범위 구성요소 처리에 걸린 시간을 로그 파일에 기록할지 여부(Oracle Hyperion Financial Management 애플리케이션만 해당) 및 멤버 범위 구성요소의 설명과 캡션을 편집할 수 있습니다.

### 주:

멤버 범위 구성요소는 Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

멤버 범위 구성요소를 편집하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너의 **멤버 범위**에서 멤버 범위를 선택하여 해당 등록정보를 편집합니다. 편집할 수 있는 등록정보는 다음과 같습니다. [멤버 범위 구성요소 생성](#)을 참조하십시오.
  - 캡션, 설명(description) 및 설명(comment)
  - 멤버 범위를 정의하기 위해 선택하는 변수
  - 멤버 범위에 포함하는 차원
  - 각 차원의 멤버 범위를 정의하는 멤버
  - **Financial Management 사용자만 해당:** 구성요소 처리에 걸리는 시간을 로그 파일에 포함할지 여부
3. 를 누릅니다.

## 멤버 범위 구성요소 삭제

멤버 범위 구성요소를 이 구성요소가 속한 비즈니스 규칙이나 템플릿에서 제거하여 삭제합니다. 멤버 범위 구성요소는 하나의 비즈니스 규칙 또는 템플릿에서만 사용될 수 있으므로 비즈니스 규칙이나 템플릿에서 제거하면 삭제됩니다.

### 주:

멤버 범위 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

멤버 범위 구성요소를 삭제하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너의 순서도에서 삭제할 멤버 범위 구성요소를 선택합니다.
3. 멤버 범위를 마우스 오른쪽 버튼으로 누르고 **제거**를 선택합니다.

멤버 범위를 제거하면 데이터베이스에서 삭제됩니다.


## 멤버 범위 구성요소 복사 및 붙여넣기

규칙 또는 템플릿의 멤버 범위 구성요소를 복사하여 같거나 다른 규칙 또는 템플릿에 붙여넣을 수 있습니다. 멤버 범위 구성요소를 복사하여 다른 멤버 범위 구성요소나 다른 구성요소 유형에 붙여넣을 수는 없습니다.

### 주:

멤버 범위 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

규칙 디자이너 또는 템플릿 디자이너에서 멤버 범위 구성요소를 복사하려면 다음을 수행합니다.

1. **시스템 뷰**에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **규칙 디자이너** 또는 **템플릿 디자이너**의 순서도에서 복사할 멤버 범위 구성요소를 마우스 오른쪽 버튼으로 누르고 **복사**를 선택합니다.
3. 다음 태스크 중 하나를 수행합니다.
  - 멤버 범위 구성요소를 **동일한** 비즈니스 규칙 또는 템플릿에 붙여넣으려면 순서도에서 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
  - 멤버 범위 구성요소를 **다른** 비즈니스 규칙 또는 템플릿에 붙여넣으려면 구성요소를 붙여넣으려는 비즈니스 규칙 또는 템플릿을 열고, 순서도에서 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누른 다음 **붙여넣기**를 선택합니다.
4.  를 누릅니다.

## 데이터 범위 구성요소(Financial Management 사용자만 해당)

### 참조:

- 데이터 범위 구성요소 정보(Financial Management 사용자만 해당)
- 데이터 범위 구성요소 생성(Financial Management 사용자만 해당)
- 데이터 범위 구성요소 열기(Financial Management 사용자만 해당)
- 데이터 범위 구성요소 편집(Financial Management 사용자만 해당)
- 데이터 범위 구성요소 삭제(Financial Management 사용자만 해당)
- 데이터 범위 구성요소 복사 및 붙여넣기(Financial Management 사용자만 해당)

## 데이터 범위 구성요소 정보(Financial Management 사용자만 해당)

데이터 범위 구성요소는 Oracle Hyperion Financial Management 애플리케이션의 광범위한 데이터 값으로 구성된 일종의 루프입니다. 데이터 범위 구성요소는 공유할 수 없으므로 비즈니스 규칙 또는 템플릿에 추가할 때마다 새 데이터 범위를 생성해야 합니다.

## 데이터 범위 구성요소 생성(Financial Management 사용자만 해당)

Oracle Hyperion Financial Management 비즈니스 규칙이나 템플릿을 디자인할 때 규칙 디자이너 또는 템플릿 디자이너에서 데이터 범위 구성요소를 생성합니다. 스크립트 또는 공식 구성요소와 달리 데이터 범위 구성요소는 해당 구성요소가 속하는 비즈니스 규칙 또는 템플릿의 구성요소로만 존재합니다. 데이터 범위 구성요소는 공유할 수 없습니다.

데이터 범위 구성요소를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 Financial Management 애플리케이션 유형, 애플리케이션, 계산 유형 순으로 확장합니다. 그런 다음 데이터 범위를 규칙에 대해 생성할지 템플릿에 대해 생성할지 여부에 따라 다음 태스크 중 하나를 수행합니다.
  - 규칙에서 데이터 범위를 생성하려면 규칙을 확장하고 규칙을 마우스 오른쪽 버튼으로 누른 후 열기를 선택합니다.
  - 템플릿에서 데이터 범위를 생성하려면 템플릿을 확장하고 템플릿을 마우스 오른쪽 버튼으로 누른 후 열기를 선택합니다.
2. 데이터 범위를 생성할 순서도 위치를 결정한 후 새 객체에서 데이터 범위 객체를 순서도로 끌어 놓습니다.  
데이터 범위 객체는 순서도에서 연결선이 있는 두 원으로 표시됩니다.
3. 사용자정의 템플릿 사용자만 해당: 템플릿에 대한 데이터 범위 구성요소를 생성하는 경우 디자인 타임 프롬프트 탭에서 해당 디자인 타임 프롬프트를 생성합니다. 사용자정의 템플릿에 대한 디자인 타임 프롬프트 생성을 참조하십시오.
4. 데이터 범위 탭에서 다음 태스크 중 하나를 수행합니다.
  - 비즈니스 규칙에 대한 데이터 범위를 생성할 때 변수를 사용하여 데이터 범위를 정의하려면(변수 작업 참조) 다음 태스크를 수행합니다.
    - a. 변수 선택기를 누르고 변수를 입력하거나 생성하여 데이터 범위를 정의합니다.

### 주:


이 옵션을 선택하면 그리드의 차원에 대한 데이터 범위를 정의할 수 없습니다.

- b. 변수를 데이터 범위 구성요소에 동적으로 링크할지 여부를 선택합니다. 동적으로 링크할 경우 Oracle Hyperion Calculation Manager 외부에서 변수가 변경될 때마다 데이터 범위 구성요소에서 변수가 업데이트됩니다.
- c. 값 변수로 POV를 포함하는 대체 문자열 변수를 선택합니다. 이 버튼을 사용하여 변수를 선택할 경우 그리드의 값 열에 멤버를 입력하는 것과 같습니다.
- d. 선택 사항: 차원 정렬에서 정렬에 사용할 차원을 선택합니다.
- e. 선택 사항 이전 단계에서 정렬에 사용할 차원을 선택한 경우 정렬 방법에서 차원을 정렬하는 데 사용할 방법을 선택합니다.

- **오름차순** - 차원 멤버를 A에서 Z까지 순서로 정렬합니다. 기본 설정입니다.
- **내림차순** - 차원 멤버를 Z에서 A까지 순서로 정렬합니다.
- 비즈니스 규칙이나 템플릿을 계산할 때 데이터 범위 처리에 걸리는 시간을 로그 파일에 포함하려면 **타이머 사용**을 선택합니다. 타이머가 활성화된 모든 객체에 대한 처리 시간이 로그 파일에 포함됩니다. 예를 들어 타이머가 사용되는 규칙이 있고 이 규칙에 타이머가 사용되는 고정 루프 세 개가 포함된 경우 규칙과 규칙의 각 고정 루프를 처리하는 데 걸린 시간이 기록됩니다.
- **그리드 재설정**을 눌러 그리드를 기본 설정으로 표시합니다.
- **사용자정의 템플릿에 대한 데이터 범위를 생성할 때 디자인 타임 프롬프트를 사용하여 데이터 범위를 정의하려면 다음을 수행합니다.**
  - a. **변수 선택기**를 누르고 변수를 입력하거나 생성합니다.
  - b. 변수를 데이터 범위 구성요소에 동적으로 링크할지 여부를 선택합니다. 동적으로 링크할 경우 Calculation Manager 외부에서 변수가 변경될 때마다 데이터 범위 구성요소에서 변수가 업데이트됩니다.
  - c. **값 변수**로 데이터 범위 유형의 대체 변수를 선택합니다. 데이터 범위 변수에는 전체 데이터 범위 정의가 포함됩니다.
  - d. **선택 사항: 차원 정렬**에서 정렬에 사용할 차원을 선택합니다.
  - e. **선택 사항** 이전 단계에서 정렬에 사용할 차원을 선택한 경우 **정렬 방법**에서 차원을 정렬하는 데 사용할 방법을 선택합니다.
    - **오름차순** - 차원 멤버를 A에서 Z까지 순서로 정렬합니다. 기본 설정입니다.
    - **내림차순** - 차원 멤버를 Z에서 A까지 순서로 정렬합니다.
    - 비즈니스 규칙이나 템플릿을 계산할 때 데이터 범위 처리에 걸리는 시간을 로그 파일에 포함하려면 **타이머 사용**을 선택합니다. 타이머가 활성화된 모든 객체에 대한 처리 시간이 로그 파일에 포함됩니다. 예를 들어 타이머가 사용되는 규칙이 있고 이 규칙에 타이머가 사용되는 고정 루프 세 개가 포함된 경우 규칙과 규칙의 각 고정 루프를 처리하는 데 걸린 시간이 기록됩니다.
    - **그리드 재설정**을 눌러 그리드를 기본 설정으로 표시합니다.
- **비즈니스 규칙 또는 사용자정의 템플릿에 대한 데이터 범위를 생성할 때 변수, 멤버 및 함수를 사용하여 데이터 범위를 정의하려면 다음을 수행합니다.**
  - a. 데이터 범위를 정의할 첫 번째 차원에 대해 **값** 열에 멤버를 입력하거나, 행을 눌러 **작업** 아이콘을 표시합니다.
  - b. **작업** 아이콘을 누르고 다음 옵션 중 하나를 선택하여 데이터 범위를 정의합니다.
    - 변수([변수 작업](#) 참조)
    - 멤버([구성요소에 멤버 및 함수 추가](#) 참조)
    - 함수([함수 작업](#) 참조)
  - c. **변수** 열에 실행 변수를 입력하거나, 행을 눌러 **작업** 아이콘을 표시합니다. **작업** 아이콘을 누르고 **변수**를 선택하고 실행 변수를 선택합니다. [변수 작업](#)을 참조하십시오.

 주:

시나리오, 연도, 기간, 엔티티 또는 값 차원에 대한 실행 변수는 포함할 수 없습니다.

- d. 설명 아이콘을 누르고 데이터 범위에 대한 설명을 입력합니다.
  - e. 데이터 범위를 정의할 각 차원에 대해 이 단계를 반복합니다.
5. **선택 사항:** 등록정보 탭에서 다음 태스크를 수행합니다.
- 비즈니스 규칙이나 템플릿 검증 및 계산에서 데이터 범위 구성요소를 제외하려면 **사용 안 함**을 선택합니다.
  - 데이터 범위 구성요소를 확인할 캡션을 입력합니다. 캡션은 해당 캡션이 속한 규칙이나 템플릿의 순서도에서 구성요소 아래에 표시됩니다.
  - 데이터 범위 구성요소에 대한 설명(description) 및 설명(comment)을 입력합니다.
6.  를 누릅니다.

## 데이터 범위 구성요소 열기(Financial Management 사용자만 해당)

데이터 범위 구성요소는 해당 구성요소가 사용되는 비즈니스 규칙 또는 템플릿의 순서도에서 열 수 있습니다. 공식 및 스크립트 구성요소와 달리 시스템 뷰에서는 열 수 없습니다.

데이터 범위 구성요소를 열려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너의 순서도에서 데이터 범위 구성요소를 선택하여 데이터 범위 등록정보를 표시합니다.

## 데이터 범위 구성요소 편집(Financial Management 사용자만 해당)

데이터 범위 구성요소를 정의할 때 사용하는 차원과 멤버, 변수 또는 디자인 타임 프롬프트를 편집할 수 있고, 데이터 범위 구성요소 처리에 걸린 시간을 로그 파일에 기록할지 여부 및 데이터 범위 구성요소의 설명과 캡션을 편집할 수 있습니다.

데이터 범위 구성요소를 편집하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너의 데이터 범위에서 순서도의 데이터 범위를 선택하여 해당 등록정보를 편집합니다.


[데이터 범위 구성요소 생성\(Financial Management 사용자만 해당\)](#)을 참조하십시오.

3.  를 누릅니다.

## 데이터 범위 구성요소 삭제(Financial Management 사용자만 해당)

데이터 범위 구성요소를 해당 구성요소가 속한 비즈니스 규칙이나 템플릿에서 제거하여 삭제합니다. 데이터 범위 구성요소는 하나의 비즈니스 규칙 또는 템플릿에서만 사용될 수 있으므로 해당 비즈니스 규칙이나 템플릿에서 제거하면 삭제됩니다.


데이터 범위 구성요소를 삭제하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너의 순서도에서 데이터 범위 구성요소를 마우스 오른쪽 버튼으로 누르고 제거를 선택합니다.  
비즈니스 규칙이나 템플릿에서 데이터 범위 구성요소를 제거하면 데이터베이스에서 삭제됩니다.
3. 구성요소의 삭제를 확인합니다.
4.  를 누릅니다.

## 데이터 범위 구성요소 복사 및 붙여넣기(Financial Management 사용자만 해당)

규칙 또는 템플릿에서 데이터 범위 구성요소를 복사하여 같거나 다른 규칙 또는 템플릿에 붙여넣을 수 있습니다. 데이터 범위 구성요소를 복사하여 다른 데이터 범위 구성요소나 다른 구성요소 유형에 붙여넣을 수는 없습니다.

규칙 디자이너 또는 템플릿 디자이너에서 데이터 범위 구성요소를 복사하여 붙여넣으려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너의 순서도에서 복사할 데이터 범위 구성요소를 마우스 오른쪽 버튼으로 누르고 복사를 선택합니다.
3. 다음 태스크 중 하나를 수행합니다.
  - 데이터 범위 구성요소를 동일한 비즈니스 규칙 또는 템플릿에 붙여넣으려면 순서도에서 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 붙여넣기를 선택합니다.
  - 데이터 범위 구성요소를 다른 비즈니스 규칙 또는 템플릿에 붙여넣으려면 데이터 범위 구성요소를 붙여넣으려는 비즈니스 규칙 또는 템플릿을 열고, 순서도에서 데이터 범위 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누른 다음 붙여넣기를 선택합니다.
4.  를 누릅니다.

## 셀 텍스트 범위 구성요소(Financial Management 사용자만 해당)

### 참조:

- 셀 텍스트 범위 구성요소 정보(Financial Management 사용자만 해당)
- 셀 텍스트 범위 구성요소 생성(Financial Management 사용자만 해당)
- 셀 텍스트 범위 구성요소 열기(Financial Management 사용자만 해당)
- 셀 텍스트 범위 구성요소 편집(Financial Management 사용자만 해당)
- 셀 텍스트 범위 구성요소 삭제(Financial Management 사용자만 해당)
- 셀 텍스트 범위 구성요소 복사 및 붙여넣기(Financial Management 사용자만 해당)

## 셀 텍스트 범위 구성요소 정보(Financial Management 사용자만 해당)

셀 텍스트 범위 구성요소는 Oracle Hyperion Financial Management 애플리케이션의 광범위한 셀 텍스트로 구성된 일종의 루프입니다. 셀 텍스트 범위 구성요소는 공유할 수 없으므로, Business Rule이나 템플릿에 하나를 추가할 때마다 새로 생성해야 합니다.

## 셀 텍스트 범위 구성요소 생성(Financial Management 사용자만 해당)

Oracle Hyperion Financial Management 비즈니스 규칙이나 템플릿을 디자인할 때 규칙 디자이너 또는 템플릿 디자이너에서 셀 텍스트 범위 구성요소를 생성합니다. 스크립트 또는 공식 구성요소와 달리 셀 텍스트 범위 구성요소는 해당 구성요소가 속하는 비즈니스 규칙 또는 템플릿의 구성요소로만 존재합니다. 셀 텍스트 범위 구성요소는 공유할 수 없습니다.

셀 텍스트 범위 구성요소를 생성하려면 다음을 수행합니다.

1. **시스템 뷰**에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 셀 텍스트 범위를 생성할 순서도 위치를 결정한 후 **새 객체**에서 셀 텍스트 범위 객체를 순서대로 끌어 놓습니다.  
셀 텍스트 범위 객체는 순서도에서 연결선이 있는 두 원으로 표시됩니다.
3. **사용자정의 템플릿 사용자만 해당**: 템플릿에 대한 셀 텍스트 범위 구성요소를 생성하는 경우 **디자인 타임 프롬프트** 탭에서 해당 디자인 타임 프롬프트를 생성합니다. **사용자정의 템플릿에 대한 디자인 타임 프롬프트 생성**을 참조하십시오.
4. **셀 텍스트 범위** 탭에서 다음 태스크 중 하나를 수행합니다.
  - **비즈니스 규칙**에 대한 셀 텍스트 범위를 생성할 때 변수를 사용하여 셀 텍스트 범위를 정의하려면 **변수 작업** 참조) 다음 태스크를 수행합니다.
    - a. **변수 선택기**를 누르고 변수를 입력하거나 생성하여 셀 텍스트 범위를 정의합니다.

### 주:

이 옵션을 선택하면 그리드의 차원에 대한 셀 텍스트 범위를 정의할 수 없습니다.

- b. 변수를 셀 텍스트 범위 구성요소에 동적으로 링크할지 여부를 선택합니다. 동적으로 링크할 경우 Oracle Hyperion Calculation Manager 외부에서 변수가 변경될 때마다 셀 텍스트 범위 구성요소에서 변수가 업데이트됩니다.
- c. **값 변수**로 POV를 포함하는 대체 문자열 변수를 선택합니다. 이 버튼을 사용하여 변수를 선택할 경우 그리드의 값 열에 멤버를 입력하는 것과 같습니다.
- d. **선택 사항**: **차원 정렬**에서 정렬에 사용할 차원을 선택합니다.
- e. **선택 사항** 이전 단계에서 정렬에 사용할 차원을 선택한 경우 **정렬 방법**에서 차원을 정렬하는 데 사용할 방법을 선택합니다.
  - **오름차순** - 차원 멤버를 A에서 Z까지 순서로 정렬합니다. 기본 설정입니다.
  - **내림차순** - 차원 멤버를 Z에서 A까지 순서로 정렬합니다.
- **사용자정의 템플릿에 대한 셀 텍스트 범위를 생성할 때 디자인 타임 프롬프트**를 사용하여 셀 텍스트 범위를 정의하려면 다음을 수행합니다.
  - a. **변수 선택기**를 누르고 변수를 입력하거나 생성합니다.



- b. 변수를 셀 텍스트 범위 구성요소에 동적으로 링크할지 여부를 선택합니다. 동적으로 링크할 경우 계산 관리자 외부에서 변수가 변경될 때마다 셀 텍스트 범위 구성요소에서 변수가 업데이트됩니다.
- c. **값** 변수로 셀 텍스트 범위 유형의 대체 변수를 선택합니다. (셀 텍스트 범위 변수에는 전체 셀 텍스트 범위 정의가 포함되어 있습니다.)
- d. **선택 사항: 차원 정렬**에서 정렬에 사용할 차원을 선택합니다.
- e. **선택 사항** 이전 단계에서 정렬에 사용할 차원을 선택한 경우 **정렬 방법**에서 차원을 정렬하는 데 사용할 방법을 선택합니다.
  - **오름차순** - 차원 멤버를 A에서 Z까지 순서로 정렬합니다. 기본 설정입니다.
  - **내림차순** - 차원 멤버를 Z에서 A까지 순서로 정렬합니다.
- **비즈니스 규칙 또는 사용자정의 템플릿에 대한 셀 텍스트 범위를 생성할 때 변수, 멤버 및 함수를 사용하여 셀 텍스트 범위를 정의하려면 다음을 수행합니다.**
  - a. 셀 텍스트 범위를 정의할 첫 번째 차원에 대해 **값** 열에 멤버를 입력하거나, 행을 눌러 **작업** 아이콘을 표시합니다.
  - b. **작업** 아이콘을 누르고 다음 옵션 중 하나를 선택하여 셀 텍스트 범위를 정의합니다.
    - 변수(**변수 작업** 참조)
    - 멤버(**구성요소에 멤버 및 함수 추가** 참조)
    - 함수(**함수 작업** 참조)
  - c. **변수** 열에 실행 변수를 입력하거나, 행을 눌러 **작업** 아이콘을 표시합니다. **작업** 아이콘을 누르고 **변수**를 선택하고 실행 변수를 선택합니다. **변수 작업**을 참조하십시오.

 **주:**

시나리오, 연도, 기간, 엔티티 또는 값 차원에 대한 실행 변수는 포함할 수 없습니다.

- d. **설명** 아이콘을 누르고 셀 텍스트 범위에 대한 설명을 입력합니다.
- e. 그리드에서 셀 텍스트 범위를 정의할 각 차원에 대해 이 단계를 반복합니다.
- f. **<셀 텍스트 레이블>**을 지정하기 위해 **값** 열에 변수 값을 입력하거나 **작업** 아이콘을 눌러 셀 텍스트 레이블 값으로 사용할 변수를 선택합니다. 그런 다음 **변수** 열에 변수를 입력하거나 **작업** 아이콘을 누르고 변수를 선택하여 범위를 정의합니다.
- g. **<POV>**, **<인덱스>** 및 **<셀 텍스트>**를 지정하기 위해 **변수** 열에 변수를 입력하거나 **작업** 아이콘을 눌러 변수를 선택합니다.
- 5. 비즈니스 규칙이나 템플릿을 계산할 때 셀 텍스트 범위 처리에 걸리는 시간을 로그 파일에 포함하려면 **타이머 사용**을 선택합니다. 타이머가 활성화된 모든 객체에 대한 처리 시간이 로그 파일에 포함됩니다.
- 6. **그리드 재설정**을 눌러 그리드를 기본 설정으로 표시합니다.
- 7. **선택 사항: 등록정보** 탭에서 다음 태스크를 수행합니다.
  - 비즈니스 규칙이나 템플릿 검증 및 계산에서 셀 텍스트 범위 구성요소를 제외하려면 **사용 안 함**을 선택합니다.

- 셀 텍스트 범위 구성요소를 확인할 캡션을 입력합니다.  
캡션은 해당 캡션이 속한 규칙이나 템플리트의 순서도에서 구성요소 아래에 표시됩니다.
- 셀 텍스트 범위 구성요소에 대한 설명(description) 및 설명(comment)을 입력합니다.

8. 를 누릅니다.

## 셀 텍스트 범위 구성요소 열기(Financial Management 사용자만 해당)

셀 텍스트 범위 구성요소는 해당 구성요소가 사용되는 비즈니스 규칙 또는 템플리트의 순서도에서 열 수 있습니다. 공식 및 스크립트 구성요소와 달리 시스템 뷰에서는 열 수 없습니다.

셀 텍스트 범위 구성요소를 열려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플리트를 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙 디자이너 또는 템플리트 디자이너의 순서도에서 셀 텍스트 범위 구성요소를 선택하여 셀 텍스트 범위 등록정보를 표시합니다.

## 셀 텍스트 범위 구성요소 편집(Financial Management 사용자만 해당)

셀 텍스트 범위 구성요소를 정의할 때 사용하는 차원과 멤버, 변수 또는 디자인 타임 프롬프트를 편집할 수 있고, 셀 텍스트 범위 구성요소 처리에 걸린 시간을 로그 파일에 기록할지 여부 및 셀 텍스트 범위 구성요소의 설명과 캡션을 편집할 수 있습니다.

셀 텍스트 범위 구성요소를 편집하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플리트를 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙 디자이너 또는 템플리트 디자이너의 셀 텍스트 범위에서 순서도의 셀 텍스트 범위를 선택하여 해당 등록정보를 편집합니다.

[셀 텍스트 범위 구성요소 생성\(Financial Management 사용자만 해당\)](#)을 참조하십시오.

3. 를 누릅니다.


## 셀 텍스트 범위 구성요소 삭제(Financial Management 사용자만 해당)

셀 텍스트 범위 구성요소는 하나의 Business Rule 또는 템플리트에서만 사용될 수 있으므로, 현재 속해 있는 Business Rule이나 템플리트에서 제거하면 삭제됩니다.

셀 텍스트 범위 구성요소를 삭제하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플리트를 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙 디자이너 또는 템플리트 디자이너의 순서도에서 셀 텍스트 범위 구성요소를 마우스 오른쪽 버튼으로 누르고 제거를 선택합니다.


Business Rule이나 템플리트에서 셀 텍스트 범위 구성요소를 제거하면 데이터베이스에서 삭제됩니다.

3. 구성요소 삭제를 확인하고 를 누릅니다.

## 셀 텍스트 범위 구성요소 복사 및 붙여넣기(Financial Management 사용자만 해당)

규칙 또는 템플릿에서 셀 텍스트 범위 구성요소를 복사하여 같거나 다른 규칙 또는 템플릿에 붙여넣을 수 있습니다. 셀 텍스트 범위 구성요소를 복사하여 다른 셀 텍스트 범위 구성요소나 다른 구성요소 유형에 붙여넣을 수는 없습니다.

규칙 디자이너 또는 템플릿 디자이너에서 셀 텍스트 범위 구성요소를 복사하여 붙여넣으려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너의 순서도에서 복사할 셀 텍스트 범위 구성요소를 마우스 오른쪽 버튼으로 누르고 **복사**를 선택합니다.
3. 다음 태스크 중 하나를 수행합니다.
  - 셀 텍스트 범위 구성요소를 **동일한** 비즈니스 규칙 또는 템플릿에 붙여넣으려면 순서도에서 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
  - 셀 텍스트 범위 구성요소를 **다른** 비즈니스 규칙 또는 템플릿에 붙여넣으려면 셀 텍스트 범위 구성요소를 붙여넣으려는 비즈니스 규칙 또는 템플릿을 열고, 순서도에서 데이터 범위 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누른 다음 **붙여넣기**를 선택합니다.
4. 를 누릅니다.

## 고정 루프 구성요소

고정 루프 구성요소는 메타데이터 멤버 목록을 정해진 횟수만큼 순환하는 객체입니다.

참조:

- [고정 루프 구성요소 정보](#)
- [고정 루프 구성요소 생성](#)
- [고정 루프 구성요소 열기](#)
- [고정 루프 구성요소 편집](#)
- [고정 루프 구성요소 삭제](#)
- [고정 루프 구성요소 복사 및 붙여넣기](#)

## 고정 루프 구성요소 정보

고정 루프 구성요소는 메타데이터 멤버 또는 데이터 레코드(Oracle Hyperion Financial Management 애플리케이션만 해당) 목록을 지정된 횟수만큼 순환하는 객체입니다. 예를 들어 계정 목록을 10회 순환하는 고정 루프를 생성할 수 있습니다.

 주:

고정 루프 구성요소는 Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

## 고정 루프 구성요소 생성


고정 루프 아이콘을 규칙 디자이너 또는 템플릿 디자이너 순서도로 끌어 놓아 비즈니스 규칙 또는 템플릿에서 고정 루프 구성요소를 생성합니다. 고정 루프 구성요소는 이 구성요소가 생성된 비즈니스 규칙 또는 템플릿 내에서만 존재합니다. 고정 루프 구성요소는 비즈니스 규칙 또는 템플릿 간에 공유할 수 없습니다.

 주:

고정 루프 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

고정 루프 구성요소를 생성하려면 다음을 수행합니다.

1. **시스템 뷰**에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 고정 루프를 생성할 비즈니스 규칙 또는 템플릿 순서도 위치를 결정한 후 **새 객체**에서 **고정 루프** 객체를 순서도로 끌어 놓습니다.  
고정 루프는 선으로 연결된 두 원으로 표시됩니다.
3. **사용자정의 템플릿 사용자만 해당**: 템플릿에 대한 고정 루프 구성요소를 생성하는 경우 **디자인 타임 프롬프트** 탭에서 해당 디자인 타임 프롬프트를 생성합니다. **사용자정의 템플릿에 대한 디자인 타임 프롬프트 생성**을 참조하십시오.
4. **고정 루프** 탭에서 다음 태스크를 수행합니다.
  - a. **값**에 루프가 메타데이터 또는 데이터를 순환할 횟수를 입력합니다. 또는 **작업** 아이콘을 누르고 **변수**(비즈니스 규칙으로 작업 중인 경우), **DTP**(사용자정의 템플릿으로 작업 중인 경우) 또는 **함수**(비즈니스 규칙 또는 사용자정의 템플릿으로 작업 중인 경우)를 선택하여 루프를 정의할 변수, 디자인 타임 프롬프트 또는 함수를 선택합니다. **변수 작업** 및 **함수 작업**을 참조하십시오.
  - b. **중간 합계 변수**에 변수를 입력하거나 **변수** 아이콘(비즈니스 규칙으로 작업 중인 경우) 또는 **DTP** 아이콘(사용자정의 템플릿으로 작업 중인 경우)을 눌러 각각 고정 루프를 종료할 숫자 변수나 디자인 타임 프롬프트를 선택합니다. 중간 합계 변수의 값은 1이어야 합니다.
  - c. **오류 및 경고** 탭에서 버튼을 눌러 스크립트 진단을 실행합니다.
5. **Financial Management** 비즈니스 규칙 및 **사용자정의 템플릿 사용자만 해당**: 고정 루프 탭에서 다음 태스크를 수행합니다.
  - a. **값**에 루프가 메타데이터 또는 데이터를 순환할 횟수를 입력합니다. 또는 **작업** 아이콘을 누르고 **변수**(비즈니스 규칙으로 작업 중인 경우), **DTP**(사용자정의 템플릿으로 작업 중인 경우) 또는 **함수**(비즈니스 규칙 또는 사용자정의 템플릿으로 작업 중인 경우)를 선택하여 루프를 정의할 변수, 디자인 타임 프롬프트 또는 함수를 선택합니다. **변수 작업** 및 **함수 작업**을 참조하십시오.

- b. 변수에 변수를 입력하거나 변수 아이콘(비즈니스 규칙으로 작업 중인 경우) 또는 DTP 아이콘(사용자정의 템플릿으로 작업 중인 경우)을 눌러 각각 숫자 변수나 디자인 타임 프롬프트를 선택합니다.
6. **선택 사항: 등록정보 탭에서 다음 태스크를 수행합니다.**
    - 비즈니스 규칙이나 템플릿 검증 및 계산에서 고정 루프 구성요소를 제외하려면 **사용 안 함**을 선택합니다.
    - 고정 루프 구성요소를 확인할 캡션을 입력합니다. 캡션은 해당 캡션이 속한 규칙이나 템플릿의 순서도에서 구성요소 아래에 표시됩니다.
    - 고정 루프 구성요소에 대한 설명(description) 및 설명(comment)을 입력합니다.
  7. **Financial Management 사용자만 해당: 타이머 사용**을 선택하여 비즈니스 규칙이 실행될 때 고정 루프 처리에 걸린 시간을 로그 파일에 포함합니다. 규칙, 규칙 세트 및 구성요소에 대해 타이머를 사용할 수 있습니다. 타이머가 활성화된 모든 객체에 대한 처리 시간이 로그 파일에 포함됩니다. 예를 들어 타이머가 사용되는 규칙이 있고 이 규칙에 타이머가 사용되는 고정 루프 세 개가 포함된 경우 규칙과 규칙의 각 고정 루프를 처리하는 데 걸린 시간이 기록됩니다.
  8. 을 누릅니다.

## 고정 루프 구성요소 열기

고정 루프 구성요소는 해당 구성요소가 속한 비즈니스 규칙 또는 템플릿의 순서도에서 열립니다. 공식 및 스크립트 구성요소와 달리 시스템 뷰에서는 열 수 없습니다.

### 주:

고정 루프 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

고정 루프 구성요소를 열려면 다음을 수행합니다.

1. **시스템 뷰**에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙이나 템플릿이 열려 있는 경우 순서도의 고정 루프 구성요소를 선택하여 고정 루프 등록정보를 표시합니다.

## 고정 루프 구성요소 편집


고정 루프 구성요소에 지정하는 값과 이 구성요소에 대해 선택하는 변수를 편집할 수 있습니다. 비즈니스 규칙에 대한 고정 루프를 생성하는 경우 캡션을 편집할 수도 있습니다. Oracle Hyperion Financial Management 비즈니스 규칙에 대한 고정 루프를 생성하는 경우 고정 루프 처리에 걸린 시간을 로그 파일에 기록할지 여부를 편집할 수 있습니다. Oracle Hyperion Planning 또는 Oracle Essbase 비즈니스 규칙에 대한 고정 루프를 생성하는 경우 중간 합계 변수를 편집할 수 있습니다. 중간 합계 변수는 고정 루프를 종료할 시기를 지정합니다.

 주:

고정 루프 구성요소는 Financial Management, Planning 및 Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

고정 루프 구성요소를 편집하려면 다음을 수행합니다.

1. 시스템 뷰에서 고정 루프 구성요소가 포함된 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너의 순서도에서 고정 루프를 선택하여 고정 루프 탭의 등록정보를 편집합니다. 편집할 수 있는 등록정보는 다음과 같습니다 (고정 루프 구성요소 생성을 참조하십시오.)
  - Financial Management, Planning 및 Essbase 비즈니스 규칙 및 템플릿에 있는 고정 루프 구성요소의 경우: 고정 루프에 지정하는 값과 이 고정 루프에 대해 선택하는 변수
  - Financial Management, Planning 및 Essbase 비즈니스 규칙 및 템플릿에 있는 고정 루프 구성요소의 경우: 고정 루프를 검증 및 계산에서 비활성화할지 여부와 고정 루프의 캡션, 설명(description) 및 설명(comment)  
고정 루프를 검증 및 계산에서 비활성화할지 여부와 고정 루프의 캡션, 설명(description) 및 설명(comment)
  - Planning 및 Essbase 비즈니스 규칙에 있는 고정 루프 구성요소의 경우: 중간 합계 변수
  - Financial Management 비즈니스 규칙에 있는 고정 루프 구성요소의 경우: 고정 루프 처리에 걸린 시간을 로그 파일에 기록할지 여부

규칙 디자이너 또는 템플릿 디자이너의 순서도에서 고정 루프를 선택하여 고정 루프 탭의 등록정보를 편집합니다. 고정 루프에 제공하는 값, 이 고정 루프에 대해 선택하는 변수 및 중간 합계 변수를 편집할 수 있습니다. (고정 루프 구성요소 생성을 참조하십시오.)
3.  를 누릅니다.

## 고정 루프 구성요소 삭제


고정 루프 구성요소를 해당 구성요소가 속한 비즈니스 규칙이나 템플릿에서 제거하여 삭제합니다. 고정 루프 구성요소는 하나의 비즈니스 규칙 또는 템플릿에서만 사용될 수 있으므로 해당 비즈니스 규칙이나 템플릿에서 제거하면 삭제됩니다.

 주:

고정 루프 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

고정 루프 구성요소를 삭제하려면 다음을 수행합니다.

1. 시스템 뷰에서 삭제할 고정 루프 구성요소가 포함된 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너의 순서도에서 삭제할 고정 루프 구성요소를 선택합니다.

- 고정 루프 구성요소를 마우스 오른쪽 버튼으로 누르고 **제거**를 선택합니다.
- 구성요소 삭제를 확인하고  를 누릅니다.


## 구성 루프 구성요소 복사 및 붙여넣기

규칙 또는 템플릿에서 고정 루프 구성요소를 복사하여 같거나 다른 규칙 또는 템플릿에 붙여넣을 수 있습니다. 고정 루프 구성요소를 복사하여 다른 고정 루프 구성요소나 다른 구성요소 유형에 붙여넣을 수는 없습니다.

### 주:

고정 루프 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 사용자정의 템플릿에서 사용할 수 있습니다.

규칙 디자이너 또는 템플릿 디자이너에서 고정 루프 구성요소를 복사하여 붙여넣으려면 다음을 수행합니다.

- 시스템 뷰**에서 복사할 고정 루프 구성요소가 포함된 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
- 규칙 디자이너** 또는 **템플릿 디자이너**의 순서도에서 복사할 고정 루프 구성요소를 마우스 오른쪽 버튼으로 누르고 **복사**를 선택합니다.
- 다음 태스크 중 하나를 수행합니다.
  - 고정 루프 구성요소를 **동일한** 비즈니스 규칙 또는 템플릿에 붙여넣으려면 순서도에서 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
  - 고정 루프 구성요소를 **다른** 비즈니스 규칙에 붙여넣으려면 구성요소를 붙여넣으려는 비즈니스 규칙을 열고, 순서도에서 고정 루프 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누른 다음 **붙여넣기**를 선택합니다.
-  를 누릅니다.

## 메타데이터 루프 구성요소

메타데이터 루프 구성요소를 사용하면 함수(메타데이터) 또는 시작 및 끝 값(고정)을 사용하여 여러 멤버에 값을 지정할 수 있습니다.

### 참조:

- [메타데이터 루프 구성요소 정보](#)
- [메타데이터 루프 구성요소 생성](#)
- [메타데이터 루프 구성요소 열기](#)
- [메타데이터 루프 구성요소 삭제](#)
- [메타데이터 루프 구성요소 복사 및 붙여넣기](#)

## 메타데이터 루프 구성요소 정보

### 주:

메타데이터 루프 구성요소는 Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역, 사용자정의 템플릿에서 사용할 수 있습니다.

메타데이터 루프 구성요소를 사용하면 함수(메타데이터) 또는 시작 및 끝 값(고정)을 사용하여 여러 멤버에 값을 지정할 수 있습니다.

예를 들어 상위 "총 이익"의 모든 "레벨 0" 계정에 #missing을 지정하려 할 수 있습니다. 이 예에서는 메타데이터 루프에서 다음 항목을 선택합니다.

- 차원 - "계정"
- 함수 - @RELATIVE("GP", 0)
- 변수 - "멤버" 유형 디자인 타임 프롬프트

## 메타데이터 루프 구성요소 생성

메타데이터 루프 구성요소 아이콘을 사용자정의 템플릿의 순서도로 끌어 놓아 메타데이터 루프 구성요소를 생성합니다. 메타데이터 루프 구성요소는 이 구성요소가 생성된 템플릿 내에서만 존재합니다. 템플릿 간에 메타데이터 루프 구성요소를 공유할 수 없습니다.

메타데이터 루프 구성요소를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 사용자정의 템플릿을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 템플릿 디자이너의 새 객체에서 메타데이터 루프 객체를 선택하여 순서도로 끕니다.
3. 메타데이터 루프 탭에서 다음 루프 유형 중 하나를 선택합니다.

- **메타데이터**

메타데이터를 선택한 경우 다음 정보를 입력합니다.




- **인덱스 DTP** - 메타데이터 루프 인덱스로 작동하는 기존 디자인 타임 프롬프트를 선택합니다. 메타데이터 루프가 시작되기 전에 디자인 타임 프롬프트의 값이 0으로 재설정되고 루프마다 1씩 증가합니다.
- **차원** - 메타데이터 루프에 사용할 상위 멤버가 포함된 차원을 선택합니다.
- **함수** - 차원 멤버에 변수를 적용하는 방법을 지정하는 함수를 선택합니다.
- **변수** - 변수에 사용할 디자인 타임 프롬프트 *멤버*를 생성합니다. 이 디자인 타임 프롬프트는 공식에서 사용되며 Oracle Hyperion Calculation Manager는 디자인 타임 프롬프트를 지정된 함수에서 생성한 각 멤버로 바꿉니다.

- **고정**

고정을 선택한 경우 다음 정보를 입력합니다.

- **인덱스 DTP** - 메타데이터 루프 인덱스로 작동하는 기존 단계 디자인 타임 프롬프트를 선택합니다. 메타데이터 루프가 시작되기 전에 디자인 타임 프롬프트의 값이 0으로 재설정되고 루프마다 1씩 증가합니다.



- **시작 인덱스** - 루프가 시작되는 값입니다. 를 눌러 시작 인덱스로 사용될 디자인 타임 프롬프트를 선택합니다.
  - **끝 인덱스** - 루프가 끝나는 값입니다. 를 눌러 끝 인덱스로 사용될 디자인 타임 프롬프트를 선택합니다.
4. 메타데이터 루프에 대한 등록정보를 입력합니다.
    - 검증 및 계산을 위해 메타데이터 루프 구성요소를 템플릿에서 제외하려면 **사용 안 함**을 선택합니다.  
비즈니스 규칙이 검증되지 않으며 검증 오류의 소스를 찾아야 하는 경우 구성요소를 비활성화할 수 있습니다. 구성요소를 비활성화하면 순서도에서 구성요소가 회색으로 표시됩니다.
    - 메타데이터 루프에 대한 **캡션**을 입력합니다.  
캡션은 순서도에서 메타데이터 루프 구성요소 아이콘 아래에 표시됩니다.
    - **설명(Description)** 및 **설명(Comment)**을 입력합니다.
  5. 를 누릅니다.

## 메타데이터 루프 구성요소 열기

속하는 사용자정의 템플릿의 순서도에서 메타데이터 루프 구성요소를 엽니다. 공식 및 스크립트 구성요소와 달리 메타데이터 루프 구성요소는 **시스템 뷰**에서 직접 열 수 없습니다.

메타데이터 루프 구성요소를 열려면 다음을 수행합니다.

1. **시스템 뷰**에서 사용자정의 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **템플릿 디자이너**의 순서도에서 해당 등록정보를 표시할 메타데이터 루프 구성요소를 선택합니다.

## 메타데이터 루프 구성요소 삭제


속하는 사용자정의 템플릿 내에서 메타데이터 루프 구성요소를 삭제할 수 있습니다.

메타데이터 루프 구성요소를 삭제하려면 다음을 수행합니다.

1. **시스템 뷰**에서 메타데이터 루프가 포함된 사용자정의 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **규칙 디자이너**의 순서도에서 해당 메타데이터 루프 구성요소를 마우스 오른쪽 버튼으로 누르고 **제거**를 선택합니다.

### 주:

이러한 단계는 템플릿의 메타데이터 루프만 삭제합니다. 메타데이터 루프와 함께 사용하기 위해 생성한 디자인 타임 프롬프트도 삭제해야 합니다. 디자인 타임 프롬프트를 삭제하려면 **디자인 타임 프롬프트** 탭에서 디자인 타임 프롬프트가 포함된 행을 마우스 오른쪽 버튼으로 누르고 **행 삭제**를 선택하거나 디자인 타임

프롬프트가 포함된 행을 선택하고 를 누릅니다.


## 메타데이터 루프 구성요소 복사 및 붙여넣기

사용자정의 템플릿에서 메타데이터 루프 구성요소를 복사하여 동일하거나 다른 사용자정의 템플릿에 붙여넣을 수 있습니다.

메타데이터 루프 구성요소를 복사하고 붙여넣으려면 다음을 수행합니다.

1. **시스템 뷰**에서 메타데이터 루프 구성요소가 포함된 사용자정의 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **규칙 디자이너**의 순서도에서 복사할 메타데이터 루프 구성요소를 마우스 오른쪽 버튼으로 누르고 **복사**를 선택합니다.

메타데이터 루프 내 공식 또는 스크립트 구성요소를 끌어온 경우 **그룹 복사**를 선택하여 메타데이터 루프 및 메타데이터 루프 내 구성요소를 모두 복사합니다.

3. 다음 태스크 중 하나를 수행합니다.
  - 메타데이터 루프 구성요소를 *동일한* 템플릿에 붙여넣으려면 순서도에서 원하는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
  - 메타데이터 루프 구성요소를 *다른* 템플릿에 붙여넣으려면 원하는 템플릿을 열고 순서도에서 원하는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
4. 를 누릅니다.

### 주:

디자인 타임 프롬프트가 메타데이터 루프에서 사용되는 경우 디자인 타임 프롬프트를 복사하려면 **디자인 타임 프롬프트** 탭에서 복사할 디자인 타임 프롬프트가 포함된 행을 마우스 오른쪽 버튼으로 누르고 **복사**를 선택합니다.

모든 디자인 시간 프롬프트를 복사하려면 **그리드 복사**를 선택하고 디자인 시간 프롬프트를 붙여넣으려는 템플릿을 엽니다. **디자인 시간 프롬프트** 탭에서 그리드 안을 누른 다음 마우스 오른쪽 버튼을 누르고 **붙여넣기**를 선택합니다.

## DTP 지정 구성요소

DTP 지정 구성요소를 사용하여 사용자정의 템플릿에 있는 디자인 타임 프롬프트에 DTP, 멤버, 함수 또는 입력된 텍스트를 지정합니다.

### 참조:

- [DTP 지정 구성요소 정보](#)
- [DTP 지정 구성요소 생성](#)
- [DTP 지정 구성요소 열기](#)
- [DTP 지정 구성요소 편집](#)
- [DTP 지정 구성요소 삭제](#)
- [DTP 지정 구성요소 복사 및 붙여넣기](#)
- [DTP 지정 구성요소에서 디자인 타임 프롬프트 함수 사용](#)

## DTP 지정 구성요소 정보

DTP 지정 구성요소를 사용하여 사용자정의 템플릿에 있는 디자인 타임 프롬프트에 DTP, 멤버, 함수 또는 입력된 텍스트를 지정할 수 있습니다. DTP 지정 구성요소는 템플릿 순서도의 복잡성을 줄이고 템플릿 논리 개발 및 유지 관리를 용이하게 합니다.

### 주:

DTP 지정 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 사용자정의 템플릿에서 사용할 수 있습니다.

## DTP 지정 구성요소 생성

DTP 지정 구성요소의 아이콘을 템플릿 디자이너 순서도로 끌어 놓아 템플릿에서 DTP 지정 구성요소를 생성할 수 있습니다.


DTP 지정 구성요소를 생성하는 경우 다음 사항에 유의하십시오.



- DTP 지정 구성요소는 해당 구성요소가 생성된 템플릿 내에서만 존재합니다.
- 순서도에서 DTP 조건을 사용하지 않는 멤버 범위, 멤버 블록 또는 조건 구성요소 안을 제외한 어디에든 DTP 구성요소를 배치할 수 있습니다.

DTP 지정 구성요소를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 사용자정의 템플릿을 엽니다.
2. 템플릿 디자이너의 새 객체에서 **DTP 지정** 객체를 선택하여 순서도로 끕니다.
3. **선택 사항:** 구성요소의 디자인 타임 프롬프트를 입력합니다. [사용자정의 템플릿에 대한 디자인 타임 프롬프트 생성](#)을 참조하십시오.
4. **Oracle Hyperion Financial Management 사용자만 해당: 타이머 사용**을 선택하여 템플릿이 포함된 비즈니스 규칙이 실행될 때 DTP 지정 처리에 걸린 시간을 로그 파일에 포함합니다. 타이머가 활성화된 모든 객체에 대한 처리 시간이 로그 파일에 포함됩니다.

예를 들어 타이머가 사용되는 규칙이 있고, 이 규칙에 타이머를 사용으로 설정한 사용자정의 템플릿이 포함되어 있으며, 이 템플릿에 타이머가 사용으로 설정된 DTP 지정 구성요소 세 개가 포함되어 있는 경우 규칙, 사용자정의 템플릿 및 템플릿의 각 DTP 지정을 처리하는 데 걸린 시간이 기록됩니다.

5. **Financial Management 사용자만 해당: 로깅 사용**을 선택합니다. 그러면 템플릿이 포함된 비즈니스 규칙에 로그 텍스트가 있는 경우, 규칙이 실행될 때 해당 텍스트가 로그 파일에 포함됩니다. 규칙, 규칙 세트 및 구성요소에 대해 로깅을 사용할 수 있습니다. 그러나 템플릿의 공식 명령문을 사용하여 포함할 로그 텍스트를 지정합니다.
6. 공식 탭에서 템플릿의 디자인 타임 프롬프트 조건을 정의합니다.
  - 등호 왼쪽의 드롭다운에서 값을 정의할 디자인 타임 프롬프트를 선택합니다. 드롭다운에 디자인 타임 프롬프트가 없는 경우 작업 중인 템플릿에 대해 아무것도 생성되지 않습니다.
  - 등호 오른쪽에 있는 텍스트 상자에 값을 입력하거나 를 눌러 변수, 멤버 또는 함수를 선택합니다.

-  를 눌러 추가 디자인 타임 프롬프트 값을 지정합니다.
7. DTP 지정 구성요소에 대한 등록정보를 입력합니다.
    - 검증 및 계산을 위해 메타데이터 루프 구성요소를 템플릿에서 제외하려면 **사용 안 함**을 선택합니다.  
비즈니스 규칙이 검증되지 않으며 검증 오류의 소스를 찾아야 하는 경우 구성요소를 비활성화할 수 있습니다. 구성요소를 비활성화하면 순서도에서 구성요소가 회색으로 표시됩니다.
    - DTP 지정 구성요소에 대한 **캡션**을 입력합니다.  
캡션은 순서도에서 DTP 지정 구성요소 아이콘 아래에 표시됩니다
    - **설명(Description)** 및 **설명(Comment)**을 입력합니다.
  8.  를 누릅니다.

## DTP 지정 구성요소 열기

DTP 지정 구성요소는 해당 구성요소가 속한 템플릿의 순서도에서 엽니다. 공식 및 스크립트 구성요소와 달리 DTP 지정 구성요소는 **시스템 뷰**에서 직접 열 수 없습니다.


DTP 지정 구성요소를 열려면 다음을 수행합니다.

1. **시스템 뷰**에서 사용자정의 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **템플릿 디자이너**의 순서도에서 해당 등록정보를 표시할 DTP 지정 구성요소를 선택합니다.

## DTP 지정 구성요소 편집

DTP 지정 구성요소의 설명(Description), 캡션 및 설명(Comment)을 편집할 수 있습니다. DTP 지정 구성요소를 사용하여 템플릿에서 디자인 타임 프롬프트에 지정하는 값을 편집할 수도 있습니다.

DTP 지정 구성요소를 편집하려면 다음을 수행합니다.

1. **시스템 뷰**에서 편집할 DTP 지정 구성요소가 포함된 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **템플릿 디자이너**의 순서도에서 DTP 지정 구성요소를 선택합니다.
3. 원하는 대로 변경하고  를 누릅니다.

## DTP 지정 구성요소 삭제

DTP 지정 구성요소가 속한 템플릿에서 이 구성요소를 삭제합니다.


DTP 지정 구성요소를 삭제하려면 다음을 수행합니다.

1. **시스템 뷰**에서 DTP 지정 구성요소가 포함된 사용자정의 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **규칙 디자이너**의 순서도에서 해당 DTP 지정 구성요소를 마우스 오른쪽 버튼으로 누르고 **제거**를 선택합니다.

## DTP 지정 구성요소 복사 및 붙여넣기

템플릿에서 DTP 지정 구성요소를 복사하여 동일한 템플릿 또는 다른 템플릿에 붙여넣을 수 있습니다.

DTP 지정 구성요소를 복사하고 붙여넣으려면 다음을 수행합니다.

1. 시스템 뷰에서 DTP 지정 구성요소가 포함된 사용자정의 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙 디자이너의 순서도에서 복사할 DTP 지정 구성요소를 마우스 오른쪽 버튼으로 누르고 **복사**를 선택합니다.
3. 다음 태스크 중 하나를 수행합니다.
  - DTP 지정 구성요소를 *동일한* 템플릿에 붙여넣으려면 순서도에서 원하는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
  - DTP 지정 구성요소를 *다른* 템플릿에 붙여넣으려면 원하는 템플릿을 열고 순서도에서 원하는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.  
DTP 지정을 새 템플릿에 붙여넣는 경우 DTP 지정 구성요소에 사용되는 디자인 타임 프롬프트도 생성하거나 새 템플릿에 복사해야 합니다.
4. 를 누릅니다.

## DTP 지정 구성요소에서 디자인 타임 프롬프트 함수 사용

참조:

- [디자인 타임 프롬프트 함수 정보](#)
- @Compare
- @Compliment
- @Concat
- @밀집Member
- @DimMember
- @DimName
- @DimType
- @EndsWith
- @Intersect
- @Member
- @Notin
- @Plandim
- @Quote
- @RemoveQuote
- @ReplaceAll
- @ReplaceFirst
- @SmartListFromIndex

- @SmartListFromValue
- @SparseMember
- @StartsWith
- @SubString
- @ToLowerCase
- @ToUpperCase
- @Trim
- @Union

## 디자인 타임 프롬프트 함수 정보

사용자정의 템플릿의 DTP 지정 구성요소에 디자인 타임 프롬프트 함수를 사용할 수 있습니다. 이러한 함수를 사용하면 멤버를 다른 멤버와 비교하거나, 두 디자인 타임 프롬프트 간에 공통되거나 공통되지 않은 멤버 목록을 가져오거나, 문자열에서 따옴표를 추가 또는 제거하거나, 문자열의 문자를 대문자 또는 소문자로 변환하는 등의 태스크를 수행할 수 있습니다.

### @Compare

@COMPARE는 멤버 간 비교, 차원 간 비교, 비밀번호 간 비교 또는 문자열과 제한 목록을 비교할 때 "true" 또는 "false"를 반환합니다. 제한 목록의 경우 @COMPARE는 대체 값이 아니라 규칙 빌더 값을 사용하여 비교합니다.

#### DTP 유형:

@COMPARE는 *부울* 유형의 DTP에 지정될 수 있습니다.

#### 구문:

@COMPARE(*String*, *compareString*, *IgnoreCase*)

#### 매개변수:

매개변수	설명
<i>String</i>	디자인 타임 프롬프트 또는 텍스트
<i>compareString</i>	디자인 타임 프롬프트 또는 텍스트
<i>IgnoreCase</i>	<b>선택 사항입니다.</b> "true"인 경우 매개변수 1 및 2의 대소문자 구분을 무시합니다. "false"인 경우 매개변수 1 및 2의 대소문자 구분이 일치해야 합니다. <i>IgnoreCase</i> 가 비어 있는 경우 기본값은 "true"입니다.

#### 예:

```
[DTP1]=@COMPARE([MBR1],[MBR2],true)
```

#### 설명:

- [DTP1]은 프롬프트할 수 없는 *부울* 유형의 DTP입니다.
- [MBR1]=Jan
- [MBR2]=Jan

이 예에서 [DTP1]은 "true"를 반환합니다.

## @Compliment

@COMPLIMENT는 DTP2에 없는 DTP1의 멤버와 DTP1에 없는 DTP2의 멤버를 반환합니다. 즉, @COMPLIMENT는 DTP1과 DTP2 사이에 공통되지 않은 멤버를 반환합니다.

### DTP 유형:

@COMPLIMENT는 *멤버 범위* 또는 *황차원* 유형의 DTP에 지정될 수 있습니다.

### 구문:

```
@COMPLIMENT (Argument1,Argument2)
```

### 매개변수:

매개변수	설명
Argument1	멤버 범위 또는 황차원을 반환하는 디자인 타임 프롬프트 또는 함수
Argument2	멤버 범위 또는 황차원을 반환하는 디자인 타임 프롬프트 또는 함수

### 예 1:

```
[DTP_MR]=@COMPLIMENT ([MR1] , [MR2])
```

#### 설명:

- [DTP\_MR]은 선택된 프롬프트가 아닌 *멤버 범위* 유형의 DTP입니다.
- [MR1]="Salaries","FY14","Local","Tennessee","USA"
- [MR2]="Jan", "Feb", "Mar","FY15","Actual",  
"Budget","Working","Florida","California","000", "G100"

이 예에서 DTP\_MR은 "Salaries","Jan", "Feb", "Mar", "Actual",  
"Budget","Working","Local","000", "G100","USA"를 반환합니다.

### 예 2:

```
[DTP_CD]=@COMPLIMENT ([CD1] , [CD2])
```

#### 설명:

- [DTP\_CD]는 선택된 프롬프트가 아닌 *황차원* 유형의 DTP입니다.
- [CD1]="Salaries"->"Jan"->"Actual"->"Local"
- [CD2]="50100"->"Feb"->"Budget"->"Working"->"California"->"000"

이 예에서 [DTP\_CD]는 "Working"->"Local"->"California"->"000"을 반환합니다.

## @Concat

@CONCAT는 첫 번째 매개변수의 끝에 두 번째 매개변수를 연결합니다.

### DTP 유형:

@CONCAT는 *멤버*, *차원*, *문자열*, *제한된 목록*, *비밀번호* 또는 *속성* 유형의 DTP에 지정될 수 있습니다.

**구문:**

```
@CONCAT(Text, Concat String, Seperator)
```

**매개변수:**

매개변수	설명
<i>Text</i>	디자인 타임 프롬프트 또는 텍스트
<i>Concat String</i>	디자인 타임 프롬프트 또는 텍스트 <b>참고:</b> <i>concatString</i> 은 <i>String</i> 끝에 추가됩니다.
<i>Seperator</i>	<b>선택 사항:</b> 구분자가 사용된 경우 <i>Text</i> 에 값이 있는 경우에만 <i>Text</i> 및 <i>Concat String</i> 사이에 구분자가 추가됩니다.

**예:**

```
[DTP_CTmbr]=@CONCAT([MBR1],[MBR2])
```

**설명:**

- [DTP\_CTmbr]은 선택된 프롬프트가 아닌 **멤버** 유형의 DTP입니다.
- [MBR1]="California"
- [MBR2]="Washington"

이 예에서 [DTP\_CTmbr]은 "CaliforniaWashington"을 반환합니다.

## @밀집Member

@밀집MEMBER는 횡차원의 첫 번째 밀집 차원 멤버를 반환합니다.

**DTP 유형:**

@밀집MEMBER는 **멤버**, **멤버(둘 이상)**, **차원**, **차원(둘 이상)**, **멤버 범위**, **횡차원** 또는 **문자열** 유형의 DTP에 지정될 수 있습니다.

**구문:**

```
@밀집MEMBER(Members)
```

**매개변수:**

매개변수	설명
<i>Members</i>	<b>횡차원</b> 또는 <b>멤버 범위</b> 를 반환하는 디자인 타임 프롬프트, 멤버 또는 함수

**예:**

```
[밀집Mbr]=@밀집MEMBER([CD1])
```

**설명:**

- [밀집Mbr]은 선택된 프롬프트가 아닌 **횡차원** 유형의 DTP입니다.
- [CD1]은 프롬프트할 수 있는 **횡차원** 유형의 DTP입니다.



- [CD1]="Salaries"->"Jan"->"Working"->"Tennessee"->"G401"

이 예에서 [Densembr]은 "Salaries"를 반환합니다. "Salaries"가 횡차원 목록의 첫번째 밀집 멤버이기 때문입니다.

## @DimMember

@DIMMEMBER에서는 지정된 차원에 적합한 경우 멤버를 반환합니다.

### DTP 유형:

@DIMMEMBER는 *멤버*, *멤버(둘 이상)*, *횡차원*, *멤버 범위*, *차원* 또는 *차원(둘 이상)* 유형의 DTP에 지정될 수 있습니다.

### 구문:

@DIMMEMBER(*Dimension*,*Member*)

### 매개변수:

매개변수	설명
<i>Dimension</i>	디자인 타임 프롬프트 또는 차원
<i>Member</i>	디자인 타임 프롬프트, 멤버 또는 차원을 선택할 수 있는 애플리케이션의 차원 목록이 표시된 함수 드롭다운

### 예:

```
[DIM_mbr1]=@DIMMEMBER([DIM],[Member])
```

### 설명:

- [DIM\_mbr1]은 선택된 프롬프트가 아닌 *멤버* 유형의 DTP입니다.
- [DIM]은 선택된 프롬프트인 *차원* 유형의 DTP입니다.
- [Member]는 선택된 프롬프트인 *멤버* 유형의 DTP입니다.
- [Dim]="Entity"
- [Member]="Washington"

이 예에서 [DIM\_mbr1]은 "Washington"을 반환합니다. Washington이 Entity 차원의 멤버이기 때문입니다.

## @DimName

@DIMNAME은 데이터베이스에 적합한 경우 차원 이름을 반환합니다.

### DTP 유형:

@DIMNAME은 *차원* 유형의 DTP에 지정될 수 있습니다.

### 구문:

@DIMNAME(*Dimension*)

**매개변수:**

매개변수	설명
<i>Dimension</i>	디자인 타임 프롬프트, 차원 또는 입력된 텍스트입니다. 사용 가능한 차원 목록을 표시하려면 차원을 누릅니다.

**예:**

```
[DIMN_Product]=@DIMNAME(["Product"])
```

**설명:**

- [DIMN\_Product]는 선택된 프롬프트가 아닌 *차원* 유형의 DTP입니다.
- "Product"가 함수에 입력됩니다.

"Product"가 이 계획 유형의 적합한 차원 이름인 경우 [DIMN\_Product]는 "Product"를 반환합니다.

"Product"가 이 계획 유형의 적합한 차원이 아닌 경우 [DIMN\_Product]는 "empty"를 반환합니다.

## @DimType

@DIMTYPE에서는 지정된 유형과 일치하는 차원을 반환합니다.

**DTP 유형:**

@DIMTYPE은 *차원*, *차원(둘 이상)* 또는 *문자열* 유형의 DTP에 지정될 수 있습니다.

**구문:**

```
@DIMTYPE(Dimension Type)
```

**매개변수:**

매개변수	설명
<i>Dimension Type</i>	애플리케이션에 사용되는 차원 유형

**예:**

```
[Dims]=@DIMTYPE(Account)
```

**설명:**

- [Dims]는 선택된 프롬프트가 아닌 *차원* 유형의 DTP입니다.
- [Account]는 드롭다운에서 선택됩니다.

이 예에서 [Dims]는 "Account"를 반환합니다. 애플리케이션에서 계정 차원이 Measures로 이름 지정된 경우 "Measures"를 반환합니다.

## @EndsWith

@ENDSWITH는 문자열이 지정된 접미어로 끝나는지 확인하기 위해 테스트합니다.

**DTP 유형:**

@ENDSWITH는 *부울* 유형의 DTP에 지정될 수 있습니다.

**구문:**

```
@ENDSWITH(String,Suffix)
```

**매개변수:**

매개변수	설명
<i>String</i>	디자인 타임 프롬프트 또는 텍스트
<i>Suffix</i>	디자인 타임 프롬프트 또는 텍스트

**예:**

```
[DTP_end]=@ENDSWITH([Mbr1],[String1])
```

**설명:**

- [Mbr1]은 선택된 프롬프트인 *멤버* 유형의 DTP입니다.
- String1은 선택된 프롬프트인 *문자열* 유형의 DTP입니다.

다음 값이 입력된다고 가정합니다.

- [Mbr1]: "Joe Smith"
- [String1]: "Smith"

이 예에서 [DTP\_end]는 "true"를 반환합니다.

## @Intersect

@INTERSECT에서는 공통 차원의 멤버를 반환합니다.

**DTP 유형:**

@INTERSECT는 *멤버 범위* 또는 *황차원* 유형의 DTP에 지정될 수 있습니다.

**구문:**

```
@INTERSECT(Argument1,Argument2)
```

**매개변수:**

매개변수	설명
<i>Argument1</i>	디자인 타임 프롬프트, 함수 또는 텍스트
<i>Argument2</i>	디자인 타임 프롬프트, 함수 또는 텍스트

**예:**

```
[IntersectMR]=@INTERSECT([MR1],[MR2])
```

**설명:**

- [MR1]은 선택된 프롬프트인 *멤버 범위* 유형의 DTP입니다.
- [MR2]는 선택된 프롬프트인 *멤버 범위* 유형의 DTP입니다.

MR1 입력이 다음과 같다고 가정합니다.

- 계정: "Salaries"
- 버전: "Target"
- 엔티티: "Tennessee", "Florida"

MR2 입력은 다음과 같습니다.

- 계정: "50590", "50350"
- 연도: "FY10"
- 시나리오: "Budget"
- 버전: "First Pass"

이 예에서 [IntersectMR]은 "50590", "50350", "Salaries", "First Pass", "Target"을 반환합니다. "Entity"의 멤버는 MR1에만 입력되었으며 "Scenario"의 멤버는 MR2에만 입력되었으므로, 이러한 멤버는 [IntersectMR]에 포함되지 않습니다.

## @Member

@MEMBER는 적절한 멤버인 경우에 한해 멤버를 반환합니다.

### DTP 유형:

@MEMBER는 *멤버* 유형의 DTP에 지정될 수 있습니다.

### 구문:

@MEMBER (*Member*)

### 매개변수:

매개변수	설명
<i>Member</i>	디자인 타임 프롬프트, 멤버 또는 다른 디자인 타임 프롬프트 함수

### 예:

```
[Mbr_mr1]=@UNION([MR1],@MEMBER([Member]))
```

### 설명:

- [Mbr\_mr1]은 선택된 프롬프트가 아닌 *멤버 범위* 유형의 DTP입니다.
- [@UNION]은 입력된 멤버가 적합한 멤버인 경우에 한해 DTP, MR1 및 DTP 멤버의 범위를 생성합니다.
- [MR1]은 선택된 프롬프트인 *멤버 범위* 유형의 DTP입니다.
- [Member]는 선택된 프롬프트인 *멤버* 유형의 DTP입니다.

다음 값이 입력된다고 가정합니다.

- [MR1]: 계정 "Salaries", 시나리오 "Actual", 엔티티 "Tennessee"

- [Member]: "000", 제품 차원의 멤버
- 이 예에서 [Mbr\_mr1]은 "Salaries","Actual","Tennessee","000"을 반환합니다.

## @Notin

@NOTIN에서는 한 표현식에는 있지만 다른 표현식에는 없는 멤버를 반환합니다. 즉, @NOTIN은 표현식 사이에 공통되지 않은 멤버를 반환합니다.

### DTP 유형:

@NOTIN은 **멤버 범위** 또는 **횡차원** 유형의 DTP에 지정될 수 있습니다.

### 구문:

```
@NOTIN (Argument1,Argument2)
```

### 매개변수:

매개변수	설명
<i>Argument1</i>	디자인 타임 프롬프트, 함수 또는 멤버
<i>Argument2</i>	디자인 타임 프롬프트, 함수 또는 멤버

### 예:

```
[NOTIN1]=@NOTIN ([MR1], [MR2])
```

### 설명:

- [NOTIN1]은 선택된 프롬프트가 아닌 **멤버 범위** 유형의 DTP입니다.
- [MR1]은 선택된 프롬프트인 **멤버 범위** 유형의 DTP입니다.
- [MR2]는 선택된 프롬프트인 **멤버 범위** 유형의 DTP입니다.

MR1 입력이 다음과 같다고 가정합니다.

- 기간: "Jan", "Feb", "Mar"
- 시나리오: "Actual"
- 통화: "Local"
- 제품: "000"

MR2 입력이 다음과 같다고 가정합니다.

- 기간: "Feb"
- 버전: "Working "
- 통화: "Local"
- 제품: "000"

이 예에서 [@NOTIN1]은 "Actual"을 반환합니다. MR2에는 없으나 MR1에는 있는 유일한 멤버이기 때문입니다.

## @Plandim

@PLANDIM에서는 차원 이름이 애플리케이션의 데이터베이스에 있는 경우 차원 이름을 반환합니다.

### DTP 유형:

@PLANDIM은 **멤버** 또는 **차원** 유형의 DTP에 지정될 수 있습니다. @PLANDIM은 두 번째 매개변수에 값이 하나만 사용되는 경우 적용됩니다.

### 구문:

```
@PLANDIM(Database Name, Dimension)
```

### 매개변수:

매개변수	설명
<i>Database Name</i>	디자인 타임 프롬프트 또는 텍스트
<i>Dimension</i>	디자인 타임 프롬프트, 차원 또는 텍스트

### 예:

```
[PDIM_mbr]=@PLANDIM([PlTy],[Dim])
```

### 설명:

- [PDIM\_mbr]은 선택된 프롬프트가 아닌 **멤버** 유형의 DTP입니다.
- [PlTy]는 선택된 프롬프트인 **문자열** 유형의 DTP입니다.
- [Dim]은 선택된 프롬프트인 **차원** 유형의 DTP입니다.

다음 값이 입력된다고 가정합니다.

- [PlTy]: "Plan1"
- Dim: "Product"

이 예에서 [PDIM\_mbr]은 Product가 "Plan1"의 적합한 차원이므로 "Product"를 반환합니다.

## @Quote

@QUOTE는 문자열을 큰따옴표로 묶습니다.

### DTP 유형:

@QUOTE는 **멤버**, **문자열** 또는 **비밀번호** 유형의 DTP에 지정될 수 있습니다.

### 구문:

```
@QUOTE(String)
```

### 매개변수:

매개변수	설명
<i>String</i>	디자인 타임 프롬프트 또는 텍스트

예:

```
[Quote_St]=@QUOTE ([String1])
```

설명:

- [Quote\_St]는 선택된 프롬프트가 아닌 *문자열* 유형의 DTP입니다.
- [String1]은 선택된 프롬프트인 *문자열* 유형의 DTP입니다.

[String1] 입력이 "Yellow"라고 가정합니다.

이 예에서 [Quote\_St]는 "Yellow"를 반환합니다.

## @RemoveQuote

@REMOVEQUOTE는 문자열에서 큰따옴표를 제거합니다.

**DTP 유형:**

@REMOVEQUOTE는 *멤버*, *문자열* 또는 *비밀번호* 유형의 DTP에 지정될 수 있습니다.

**구문:**

```
@REMOVEQUOTE (String)
```

**매개변수:**

매개변수	설명
String	디자인 타임 프롬프트 또는 텍스트

예:

```
[RemoveQuote_St]=@REMOVEQUOTE ([String1])
```

설명:

- [RemoveQuote\_St]는 *문자열* 유형의 DTP이며 선택된 프롬프트가 아닙니다.
- [String1]은 선택된 프롬프트인 *문자열* 유형의 DTP입니다.

[String1] 입력이 "Yellow"라고 가정합니다.

이 예에서 [RemoveQuote\_St]는 "Yellow"를 반환합니다.

## @ReplaceAll

@REPLACEALL은 표현식이 포함된 문자열의 일부를 대체 문자열로 바꿉니다.

**DTP 유형:**

@REPLACEALL은 *문자열*, *비밀번호*, *멤버*, *멤버(둘 이상)*, *차원* 또는 *차원(둘 이상)* 유형의 DTP에 지정될 수 있습니다.

**구문:**

```
@REPLACEALL (String, Regular Expression, replaceString)
```

**매개변수:**

매개변수	설명
<i>String</i>	디자인 타임 프롬프트 또는 텍스트
<i>Regular Expression</i>	디자인 타임 프롬프트 또는 텍스트
<i>replaceString</i>	디자인 타임 프롬프트 또는 텍스트

**예:**

```
[ReplaceAll_Str]=@REPLACEALL([String1],[Rep_String],[new_String])
```

**설명:**

- [ReplaceAll\_Str]은 선택된 프롬프트가 아닌 *문자열* 유형의 DTP입니다.
- [String1]은 선택된 프롬프트인 *문자열* 유형의 DTP입니다.
- [Rep\_String]은 선택된 프롬프트인 *문자열* 유형의 DTP입니다.
- [new\_String]은 선택된 프롬프트인 *문자열* 유형의 DTP입니다.

매개변수가 다음과 같다고 가정합니다.

- 문자열: "This is a test of a testing string"
- 정규 표현식: "Test"
- 대체 문자열 "XYZ"

이 예에서 [ReplaceAll\_Str]은 "This is a XYZ of a XYZing string for one XYZ and two XYZ"를 반환합니다. "test with XYZ"로 대체되었습니다.

## @ReplaceFirst

@REPLACEFIRST는 발견된 첫 번째 정규 표현식을 대체 문자열로 바꿉니다.

**DTP 유형:**

@REPLACEFIRST는 *문자열*, *비밀번호*, *멤버*, *멤버(둘 이상)*, *차원* 또는 *차원(둘 이상)* 유형의 DTP에 지정될 수 있습니다.

**구문:**

```
@REPLACEFIRST(String,Regular Expression,replaceString)
```

**매개변수:**

매개변수	설명
<i>String</i>	디자인 타임 프롬프트 또는 텍스트
<i>Regular Expression</i>	디자인 타임 프롬프트 또는 텍스트
<i>replaceString</i>	디자인 타임 프롬프트 또는 텍스트



**예:**

```
[ReplaceFirst_Str]=@REPLACEFIRST([String1],[Rep_String],[new_String])
```

**설명:**

- [ReplaceFirst\_Str]은 선택된 프롬프트가 아닌 문자열 유형의 DTP입니다.
- [String1]은 선택된 프롬프트인 문자열 유형의 DTP입니다.
- [Rep\_String]은 선택된 프롬프트인 문자열 유형의 DTP입니다.
- [new\_String]은 선택된 프롬프트인 문자열 유형의 DTP입니다.

매개변수가 다음과 같다고 가정합니다.

- 문자열: "This is a testing of the new test of a test today"
- 정규 표현식: "Test"
- 대체 문자열: "XYZ"

이 예에서 [ReplaceFirst\_Str]은 "This is a XYZing of the new test of a test today"를 반환합니다. 발견된 모든 항목이 아니라 첫 번째 발견된 test만 바꿉니다.

## @SmartListFromIndex

@SMARTLISTFROMINDEX는 스마트 목록의 멤버 순서대로 스마트 목록 멤버를 반환합니다.

**DTP 유형:**

@SMARTLISTFROMINDEX는 문자열, 비밀번호, 멤버 또는 멤버(둘 이상) 유형의 DTP에 지정될 수 있습니다.

**구문:**

```
@SMARTLISTFROMINDEX(SmartList,Index)
```

**매개변수:**

매개변수	설명
<i>SmartList</i>	디자인 타임 프롬프트 또는 차원. 차원은 애플리케이션의 스마트 목록 차원을 표시합니다.
<i>Index</i>	디자인 타임 프롬프트 또는 텍스트

**예:**

```
[sl_ind_Str]=@SmartListFromIndex([String1],[index_ST])
```

**설명:**

- [sl\_ind\_Str]은 선택된 프롬프트가 아닌 문자열 유형의 DTP입니다.
- [String1]은 선택된 프롬프트인 문자열 유형의 DTP입니다.
- [index\_ST]는 선택된 프롬프트인 숫자 유형의 DTP입니다.

매개변수가 다음과 같다고 가정합니다.

- 스마트 목록: "CreditRatings"

- 인덱스: "2"  
이 예에서 스마트 목록 "CreditRatings"에는 다음 데이터가 있습니다.

이름	레이블
AA	ID_ENUMNETLABEL_AA
AAA	ID_ENUMNETLABEL_AAA
AAMinus	ID_ENUMNETLABEL_AA22
A	ID_ENUMNETLABEL_A
AMinus	ID_ENUMNETLABEL_AA22
AAPlus	ID_ENUMNETLABEL_AA11
APlus	ID_ENUMNETLABEL_A11
B	ID_ENUMNETLABEL_B
BPlus	ID_ENUMNETLABEL_B11
BMinus	ID_ENUMNETLABEL_B22
BB	ID_ENUMNETLABEL_BB
BBPlus	ID_ENUMNETLABEL_BB11
BBMinus	ID_ENUMNETLABEL_BB22
BBB	ID_ENUMNETLABEL_BBB
AAAMinus	ID_ENUMNETLABEL_AAA-
BBBPlus	ID_ENUMNETLABEL_BBB+
BBBMinus	ID_ENUMNETLABEL_BBB-
CCCPlus	ID_ENUMNETLABEL_CCC+
CCC	ID_ENUMNETLABEL_CCC

[sl\_ind\_str]에서는 "AAMinus"를 반환합니다. 두 번째 인덱스 위치에 있는 스마트 목록 멤버의 이름이기 때문입니다.

## @SmartListFromValue

@SMARTLISTFROMVALUE는 스마트 목록의 멤버 ID에 따라 스마트 목록 멤버를 반환합니다.

### DTP 유형:

@SMARTLISTFROMVALUE는 멤버가 하나만 입력되는 경우 문자열, 비밀번호, 멤버 또는 멤버(둘 이상) 유형의 DTP에 지정될 수 있습니다.

### 구문:

@SMARTLISTFROMVALUE (*SmartList*, *Value*)

### 매개변수:

매개변수	설명
<i>Smartlist</i>	디자인 타임 프롬프트 또는 차원 차원은 애플리케이션의 스마트 목록 차원을 표시합니다.
<i>Value</i>	디자인 타임 프롬프트 또는 번호

### 예:

```
[sl_val_str]=@SmartListFromValue([String1],[value_ST])
```

**설명:**

- [sl\_val\_str]은 선택된 프롬프트가 아닌 문자열 유형의 DTP입니다.
- [String1]은 선택된 프롬프트인 문자열 유형의 DTP입니다.
- [value\_ST]는 선택된 프롬프트인 숫자 유형의 DTP입니다.

매개변수가 다음과 같다고 가정합니다.

- 스마트 목록: "CreditRatings"
- 값: "5"

이 예에서 스마트 목록 "CreditRatings"에는 다음 데이터가 있습니다.

이름	레이블
AA	ID_ENUMNETLABEL_AA
AAA	ID_ENUMNETLABEL_AAA
AMinus	ID_ENUMNETLABEL_AA22
A	ID_ENUMNETLABEL_A
AMinus	ID_ENUMNETLABEL_A22
APlus	ID_ENUMNETLABEL_AA11
APlus	ID_ENUMNETLABEL_A11
B	ID_ENUMNETLABEL_B
BPlus	ID_ENUMNETLABEL_B11
BMinus	ID_ENUMNETLABEL_B22
BB	ID_ENUMNETLABEL_BB
BBPlus	ID_ENUMNETLABEL_BB11
BBMinus	ID_ENUMNETLABEL_BB22
BBB	ID_ENUMNETLABEL_BBB
AAAMinus	ID_ENUMNETLABEL_AAA-
BBBPlus	ID_ENUMNETLABEL_BBB+
BBBMinus	ID_ENUMNETLABEL_BBB-
CCCPlus	ID_ENUMNETLABEL_CCC+
CCC	ID_ENUMNETLABEL_CCC

[sl\_val\_str]에서 "AMinus"를 반환합니다. AMinus는 ID가 5인 SmartList 멤버의 이름이기 때문입니다.

## @SparseMember

@SPARSEMEMBER는 횡차원의 첫 번째 희소 차원 멤버를 반환합니다.

### DTP 유형:

@SPARSEMEMBER는 *멤버*, *멤버(둘 이상)*, *차원*, *차원(둘 이상)*, *멤버 범위*, *횡차원* 또는 *문자열* 유형의 DTP에 지정될 수 있습니다.

### 구문:

@SPARSEMEMBER (*Members*)

**매개변수:**

매개변수	설명
<i>Members</i>	디자인 타임 프롬프트, 멤버 또는 멤버 목록을 반환하는 함수

**예:**

```
[SparseMbr]=@SPARSEMEMBER([CD1])
```

**설명:**

- [SparseMbr]은 선택된 프롬프트가 아닌 **황차원** 유형의 DTP입니다.
- [CD1]은 프롬프트할 수 있는 **황차원** 유형의 DTP입니다.
- [CD1]은 "Gross Sales"->"Jan"->"FY14"->"Working"->"California"입니다.

이 예에서 [Sparsembr]은 FY14가 황차원 목록의 첫 번째 회소 멤버이므로 "FY14"를 반환합니다.

## @StartsWith

@STARTSWITH는 문자열이 지정된 접두어로 시작하는지 확인하기 위해 테스트합니다.

**DTP 유형:**

@STARTSWITH는 **부울** 유형의 DTP에 지정될 수 있습니다.

**구문:**

```
@STARTSWITH(String,Prefix)
```

**매개변수:**

매개변수	설명
<i>String</i>	디자인 타임 프롬프트 또는 텍스트
<i>Prefix</i>	디자인 타임 프롬프트 또는 텍스트

**예:**

```
[DTP_Start]=@STARTSWITH([Mbr1],[String1])
```

**설명:**

- [Mbr1]은 선택된 프롬프트인 **멤버** 유형의 DTP입니다.
- [String1]은 선택된 프롬프트인 **문자열** 유형의 DTP입니다.

다음 값이 입력된다고 가정합니다.

- [Mbr1]: "Joe Smith"
- [String1]: "Smith"

이 예에서 [DTP\_Start]는 "false"를 반환합니다.

## @SubString

@SUBSTRING은 시작 인덱스의 문자로 시작하여 종료 인덱스 전의 문자열까지 확장되는 하위 문자열을 반환합니다. 시작 인덱스는 0으로 시작하고 반환되는 텍스트에는 끝 인덱스가 포함되지 않습니다. 종료 인덱스가 문자열의 문자 수보다 크면 아무 것도 반환되지 않습니다.

### DTP 유형:

@SUBSTRING은 **멤버**, **멤버(둘 이상)**, **차원**, **차원(둘 이상)**, **문자열** 또는 **비밀번호** 유형의 DTP에 지정될 수 있습니다.

### 구문:

```
@SUBSTRING(String,Beginning Index,Ending Index)
```

### 매개변수:

매개변수	설명
<i>String</i>	디자인 타임 프롬프트 또는 텍스트
<i>Beginning Index</i>	디자인 타임 프롬프트 숫자 또는 텍스트
<i>Ending Index</i>	숫자 또는 텍스트 디자인 타임 프롬프트

### 예:

```
[SUBSTRING_Str]=@SUBSTRING([String1],[start_ind],[end_ind])
```

### 설명:

- [SUBSTRING\_Str]은 선택된 프롬프트가 아닌 **문자열** 유형의 DTP입니다.
- [String1]은 선택된 프롬프트인 **문자열** 유형의 DTP입니다.
- [start\_ind]는 선택된 프롬프트인 **숫자** 유형의 DTP입니다.
- [end\_ind]는 선택된 프롬프트인 **숫자** 유형의 DTP입니다.

매개변수가 다음과 같다고 가정합니다.

- 문자열: "This is a test of testing today."
- 시작 인덱스: "2"
- 끝 인덱스: "16"

이 예에서 [SUBSTRING\_Str]은 "this is a test o"를 반환합니다. 문자열의 2번째 문자는 "this"의 "h"이며 15번째 문자는 "of" 단어 시작 부분의 "o"이기 때문입니다.

## @ToLowerCase

@TOLOWERCASE는 문자열의 문자를 소문자로 변경합니다.

### DTP 유형:

@TOLOWERCASE는 **멤버**, **멤버(둘 이상)**, **차원**, **차원(둘 이상)**, **문자열** 또는 **비밀번호** 유형의 DTP에 지정될 수 있습니다.

**구문:**

```
@TOLOWERCASE (String)
```

**매개변수:**

매개변수	설명
<i>String</i>	디자인 타임 프롬프트 또는 텍스트

**예:**

```
[TOLOWERCASE_Str]=@TOLOWERCASE([String1])
```

**설명:**

- [TOLOWERCASE\_Str]은 선택된 프롬프트가 아닌 *문자열* 유형의 DTP입니다.
- [String1]은 선택된 프롬프트인 *문자열* 유형의 DTP입니다.

매개변수가 다음과 같다고 가정합니다.

문자열: "Testing for Today"

이 예에서 [TOLOWERCASE\_Str]은 "Testing for Today"를 반환합니다.

## @ToUpperCase

@TOUPPERCASE는 문자열의 문자를 대문자로 변경합니다.

**DTP 유형:**

@TOUPPERCASE는 *멤버*, *멤버(둘 이상)*, *차원*, *차원(둘 이상)*, *문자열* 또는 *비밀번호* 유형의 DTP에 지정될 수 있습니다.

**구문:**

```
@TOUPPERCASE (String)
```

**매개변수:**

매개변수	설명
<i>String</i>	디자인 타임 프롬프트 또는 텍스트

**예:**

```
[TOUPPERCASE_Str]=@TOUPPERCASE([String1])
```

**설명:**

- [TOUPPERCASE\_Str]은 선택된 프롬프트가 아닌 *문자열* 유형의 DTP입니다.
- [String1]은 선택된 프롬프트인 *문자열* 유형의 DTP입니다.

매개변수가 다음과 같다고 가정합니다.

문자열: "Testing to convert to upper case."

이 예에서 [TOUPPERCASE\_Str]은 "TESTING TO CONVERT TO UPPER CASE"를 반환합니다.

## @Trim

@TRIM은 선행 및 후행 공백을 제거합니다.

### DTP 유형:

@TRIM은 **멤버**, **멤버(둘 이상)**, **문자열** 또는 **비밀번호** 유형의 DTP에 지정될 수 있습니다.

### 구문:

```
@TRIM(String)
```

### 매개변수:

매개변수	설명
<i>String</i>	디자인 타임 프롬프트 또는 텍스트

### 예:

```
[TRIM_ST]=@TRIM([String1])
```

### 설명:

- [TRIM\_ST]는 선택된 프롬프트가 아닌 **문자열** 유형의 DTP입니다.
- [String1]은 선택된 프롬프트인 **문자열** 유형의 DTP입니다.

매개변수가 다음과 같다고 가정합니다.

문자열: "This is a test of leading and trailing spaces"

매개변수의 텍스트 앞뒤에 공백이 있습니다. 다음과 같이 표시됩니다.

```
String1 =      this is a test of leading and trailing spaces
```

## @Union

@UNION은 여러 디자인 타임 프롬프트에서 사용한 멤버의 조합을 반환합니다.

### DTP 유형:

@UNION은 **멤버 범위** 또는 **횡차원** 유형의 DTP에 지정될 수 있습니다.

### 구문:

```
@UNION(Argument1,Argument2)
```

### 매개변수:

매개변수	설명
<i>Argument1</i>	디자인 타임 프롬프트, 함수 또는 멤버
<i>Argument2</i>	디자인 타임 프롬프트, 함수 또는 멤버

예:

```
[UnionMR]=@UNION([MR1],[MR2])
```

설명:

- [UnionMR]은 선택된 프롬프트가 아닌 **멤버 범위** 유형의 DTP입니다.
- [MR1]은 선택된 프롬프트인 **멤버 범위** 유형의 DTP입니다.
- [MR2]는 선택된 프롬프트인 **멤버 범위** 유형의 DTP입니다.

MR1 및 MR2 입력이 다음과 같다고 가정합니다.

- [MR1]: "50350",@Relative("Q1", 0),"California","Tennessee"
- [MR2]: "Apr","Washington", "Tennessee"

이 예에서 UnionMR은 다음을 반환합니다.

```
"50350",@Relative("Q1", 0),"Apr","California","Tennessee","Washington"
```

## 스크립트 및 공식 구성요소 공유

공유 공식 또는 스크립트 구성요소는 공식 레벨과 스크립트 레벨에 있습니다.

참조:

- [스크립트 및 공식 구성요소 공유 정보](#)  
동일한 애플리케이션 유형에 속하는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙 및 템플릿 간에 공식 및 스크립트 구성요소를 공유할 수 있습니다.
- [공식 및 스크립트 구성요소를 공유에서 공유 안 됨으로 변경](#)
- [공식 및 스크립트 구성요소를 공유 안 됨에서 공유로 변경](#)

## 스크립트 및 공식 구성요소 공유 정보

동일한 애플리케이션 유형에 속하는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙 및 템플릿 간에 공식 및 스크립트 구성요소를 공유할 수 있습니다.

 주:

멤버 범위, 조건, 고정 루프, 멤버 블록 또는 DTP 지정 구성요소는 공유할 수 없습니다.

공유 공식 또는 스크립트 구성요소는 공식 레벨과 스크립트 레벨에 있습니다. 공유 공식 및 스크립트 구성요소는 공식이나 스크립트를 변경할 때 공유 구성요소가 사용되는 모든 규칙 및 템플릿에 변경사항을 반영하는 데 사용됩니다. 구성요소를 공유할 때 시스템은 원래 구성요소에 대한 교차 참조를 생성합니다. 원본 구성요소의 사본 대신 교차 참조를 생성하기 때문에 데이터베이스에서 사용되는 공간이 줄어들고 처리 시간도 단축될 수 있습니다.



## 공식 및 스크립트 구성요소를 공유에서 공유 안 됨으로 변경

공유된 공식 또는 스크립트 구성요소를 공유 안 됨으로 변경하기 전에 둘 이상의 비즈니스 규칙 또는 템플릿에서 사용되고 있는 것은 아닌지 확인해야 합니다. [사용 표시] 기능을 사용하여 공식 또는 스크립트 구성요소를 사용하는 비즈니스 규칙 및 템플릿을 볼 수 있습니다. **공식 및 스크립트 구성요소의 사용 표시**를 참조하십시오. 그런 다음 규칙 및 템플릿에서 구성요소의 [공유] 확인란을 선택 취소하여 사용되는 각 비즈니스 규칙 및 템플릿에 대한 공유 구성요소의 사본을 생성할 수 있습니다.

### 주:

공식 및 스크립트 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙과 템플릿에서 사용할 수 있습니다.

공식 또는 스크립트 구성요소를 공유에서 공유 안 됨으로 변경하려면 다음을 수행합니다.

1. **시스템 뷰**에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **규칙 디자이너** 또는 **템플릿 디자이너**에서 다음 태스크 중 하나를 수행합니다.
  - 순서도에 추가하는 공식 또는 스크립트 구성요소를 공유하지 않으려면 공식 또는 스크립트 구성요소를 순서도로 끌어 놓은 후 **등록정보**에서 **공유**를 선택 취소합니다.
  - 순서도에 이미 있는 공식 또는 스크립트 구성요소를 공유하지 않으려면 순서도에서 구성요소를 선택하고 **등록정보**에서 **공유**를 선택 취소합니다.

그러면 규칙 또는 템플릿에 구성요소 사본이 생성됩니다.

3. 를 누릅니다.

## 공식 및 스크립트 구성요소를 공유 안 됨에서 공유로 변경

공식 또는 스크립트 구성요소를 공유 안 됨에서 공유로 변경하려면 **같은 애플리케이션 유형** (Oracle Hyperion Financial Management, Oracle Hyperion Planning 또는 Oracle Essbase) 내에서 이름이 동일한 공유되거나 공유되지 않은 다른 공식 또는 스크립트 구성요소가 없는지 확인해야 합니다. 공유된 객체의 이름은 애플리케이션 전체에서 고유해야 하므로 이미 사용된 이름으로 공유 객체를 생성할 수 없습니다.

공식 또는 스크립트 구성요소를 공유하려면 기존 구성요소를 규칙 또는 템플릿의 순서도로 끌어 놓을 때 구성요소의 등록정보 탭에서 [공유] 확인란이 선택되어 있어야 합니다. ([공유] 확인란은 기본적으로 선택되어 있음).

### 주:

공식 및 스크립트 구성요소는 Financial Management, Planning 및 Essbase 블록 저장영역 비즈니스 규칙과 템플릿에서 사용할 수 있습니다.

공식 및 스크립트 구성요소를 공유 안 됨에서 공유로 변경하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙 디자이너 또는 템플릿 디자이너가 열리면 다음 태스크 중 하나를 수행합니다.
  - 공식 또는 스크립트 구성요소를 공유하려면 다음을 수행합니다.
    - 새 객체에서 공식 또는 스크립트 구성요소를 순서도의 원하는 위치로 끌어서 놓습니다.
    - 구성요소의 **등록정보** 탭에서 **공유**를 선택하고 구성요소에 이름을 지정합니다.
  - 이미 순서도에 있는 공식 또는 스크립트 구성요소를 공유하려면 다음을 수행합니다.
    - a. 순서도에서 구성요소를 선택합니다.
    - b. 구성요소의 **등록정보** 탭에서 **공유**를 선택하고 구성요소에 이름을 지정합니다. 모든 공유 구성요소에 이름이 있어야 합니다.
3. **저장**을 선택합니다.

## 구성요소 복사

비즈니스 규칙 구성요소의 1차 하위를 복사하여 붙여넣거나 비즈니스 규칙 공식 또는 스크립트 구성요소에 대한 참조를 복사하여 붙여넣습니다.


**참조:**

- [비즈니스 규칙 구성요소의 1차 하위 구성요소 복사 및 붙여넣기](#)
- [비즈니스 규칙 공식 또는 스크립트 구성요소에 대한 참조 복사 및 붙여넣기](#)

## 비즈니스 규칙 구성요소의 1차 하위 구성요소 복사 및 붙여넣기

구성요소가 포함된 비즈니스 규칙에서 작업하는 경우 구성요소의 1차 하위 구성요소(순서도에서 한 구성요소 아래에 그룹화된 구성요소)를 복사할 수 있습니다. 구성요소의 하위 멤버를 복사하여 같은 비즈니스 규칙이나 다른 비즈니스 규칙에 붙여넣을 수 있습니다.

비즈니스 규칙 구성요소의 1차 하위 구성요소를 복사하여 붙여넣으려면 다음을 수행합니다.

1. 비즈니스 규칙을 엽니다.
2. 비즈니스 규칙 순서도에서 하위 멤버를 복사하려는 멤버 범위, 고정 루프 또는 조건 구성요소를 마우스 오른쪽 버튼으로 누르고 **그룹 복사**를 선택합니다.
3. 다음 태스크 중 하나를 수행합니다.
  - 구성요소의 하위 멤버를 **동일한** 비즈니스 규칙에 붙여넣으려면 순서도에서 구성요소의 하위 멤버를 표시할 위치 왼쪽에 있는 구성요소를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다. 구성요소의 1차 하위 구성요소는 선택하는 구성요소의 오른쪽에 표시됩니다.
  - 구성요소의 하위 멤버를 **다른** 비즈니스 규칙에 붙여넣으려면 구성요소의 하위 멤버를 붙여넣으려는 비즈니스 규칙을 열고 순서도에서 구성요소의 하위 멤버를 표시할 위치 왼쪽에 있는 구성요소를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
4. 를 누릅니다.

 **주:**

구성요소의 1차 하위 구성요소를 복사하여 붙여넣으면 공유 구성요소도 복사됩니다.

## 비즈니스 규칙 공식 또는 스크립트 구성요소에 대한 참조 복사 및 붙여넣기

비즈니스 규칙 공유 스크립트 또는 공유 공식 구성요소에 대한 참조를 복사하여 붙여넣는 것은 구성요소에 대한 바로가기를 복사하여 붙여넣는 것이며, 구성요소 자체를 복사하는 것이 아닙니다. 참조를 복사한 원래 비즈니스 규칙에는 구성요소의 사본 하나만 있습니다. 참조는 공유 구성요소가 포함된 애플리케이션 및 계획 유형에 대한 포인터로 작동합니다. 공유되지 않는 구성요소에 대해 [참조 복사]를 선택하면 붙여넣은 구성요소는 구성요소의 복사본이고 원래 구성요소의 바로가기를 포함하지 않습니다. 이 경우 참조 복사는 복사와 같이 작동합니다.

공유 공식 또는 공유 스크립트 구성요소에 대한 참조를 복사하여 같은 비즈니스 규칙이나 다른 비즈니스 규칙에 붙여넣을 수 있습니다.

### 주:

Oracle Hyperion Financial Management 비즈니스 규칙 및 구성요소의 경우 구성요소에 대한 참조를 복사하여 동일한 계산 유형이나 일반 계산 유형의 비즈니스 규칙에 붙여넣어야 합니다.

공유 공식 또는 공유 스크립트 구성요소에 대한 참조를 복사하여 붙여넣으려면 다음을 수행합니다.

1. 공유 공식 또는 공유 스크립트 구성요소가 포함된 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙 디자이너 순서도에서 공유 공식 또는 공유 스크립트 구성요소를 마우스 오른쪽 버튼으로 누르고 **참조 복사**를 선택합니다.
3. 다음 태스크 중 하나를 수행합니다.
  - 구성요소의 참조를 **동일한** 비즈니스 규칙에 붙여넣으려면 순서도에서 구성요소의 참조를 표시할 위치 왼쪽에 있는 구성요소를 마우스 오른쪽 버튼으로 누릅니다. 구성요소의 참조는 선택하는 구성요소의 오른쪽에 표시됩니다. **붙여넣기**를 선택합니다.
  - 구성요소의 참조를 **다른** 비즈니스 규칙에 붙여넣으려면 구성요소의 참조를 붙여넣으려는 비즈니스 규칙을 열고 순서도에서 구성요소의 참조를 표시할 위치 왼쪽에 있는 구성요소를 마우스 오른쪽 버튼으로 누릅니다. **붙여넣기**를 누릅니다.

### 주:

애플리케이션이 동일한 애플리케이션 유형에 속하는 한 참조를 복사하여 동일한 애플리케이션이나 다른 애플리케이션에 속하는 비즈니스 규칙에 붙여넣을 수 있습니다.

4.  를 누릅니다.

## 구성요소 저장

구성요소 디자이너에서 공식 및 스크립트 구성요소를 디자인한 다음 이를 저장합니다.


참조:

- [구성요소 저장](#)
- [공식 및 스크립트 구성요소를 다른 이름으로 저장](#)

## 구성요소 저장

구성요소 디자이너에서 공식 및 스크립트 구성요소를 디자인한 다음 이를 저장합니다. 규칙 디자이너 또는 템플릿 디자이너에서 비즈니스 규칙 또는 템플릿을 저장할 때 여기에 속하는 조건, 범위 및 고정 루프 구성요소가 저장됩니다.

Oracle Hyperion Financial Management, Oracle Hyperion Planning 또는 Oracle Essbase 블록 저장영역 구성요소로 작업 중인 경우 사용자정의 템플릿으로 저장할 수 있습니다. 공식 또는 스크립트 구성요소로 작업할 때는 다른 이름으로 저장할 수 있습니다.

구성요소를 저장하려면 구성요소 디자이너, 템플릿 디자이너 또는 규칙 디자이너에서 디자인 완료 후  를 누릅니다.

## 공식 및 스크립트 구성요소를 다른 이름으로 저장

[다른 이름으로 저장]을 사용하여 스크립트 및 공식 구성요소를 다른 이름으로 저장할 수 있습니다. [다른 이름으로 저장]을 사용하면 공식 또는 스크립트 구성요소의 사본이 생성됩니다. [다른 이름으로 저장]을 사용하면 공식 또는 스크립트 구성요소의 사본이 생성됩니다. 공유된 구성요소에 대해 공유를 원하지 않는 경우 구성요소의 사본을 생성할 수 있습니다. [스크립트 및 공식 구성요소 공유](#)를 참조하십시오.

 주:

Oracle Hyperion Financial Management, Oracle Hyperion Planning 또는 Oracle Essbase 블록 저장영역 비즈니스 규칙에서 생성된 공식 및 스크립트 구성요소를 다른 이름으로 저장할 수 있습니다.

스크립트 또는 공식 구성요소를 다른 이름으로 저장하려면 다음을 수행합니다.

1. 시스템 뷰에서 공식 또는 스크립트를 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 구성요소 디자이너에서 파일, 다른 이름으로 저장 순으로 선택합니다.
3. 다른 이름으로 저장에서 공식 또는 스크립트의 새 이름을 입력하고 애플리케이션을 선택합니다. 다음 중 하나를 수행합니다.
  - 공식 또는 스크립트가 Financial Management용인 경우 계산 유형을 선택합니다.
  - 공식 또는 스크립트가 Planning용인 경우 계획 유형을 선택합니다.
  - 공식 또는 스크립트가 Essbase용인 경우 데이터베이스를 선택합니다.

 주:

다른 이름으로 저장하는 공식 또는 스크립트의 애플리케이션 유형은 변경할 수 없습니다. 새 공식 또는 스크립트의 애플리케이션 유형은 이전 공식 또는 스크립트와 동일해야 합니다. 또한 공식 또는 스크립트가 Financial Management용인 경우에는 계산 유형을 변경할 수 없습니다. 새 공식 또는 스크립트의 애플리케이션 유형은 원본 공식 또는 스크립트와 동일해야 합니다.

4. 확인을 누릅니다.

저장한 후 공식 또는 스크립트 구성요소를 보려면 시스템 뷰에서 애플리케이션 목록을 새로고쳐야 할 수 있습니다. [공식 및 스크립트 구성요소 새로고치기](#)를 참조하십시오.

## 공식 및 스크립트 구성요소 새로고치기

공식 또는 스크립트 구성요소를 생성한 후에는 시스템 뷰에서 애플리케이션 목록을 새로고쳐야 공식 또는 스크립트 노드에서 확인할 수 있습니다.

애플리케이션 유형, 애플리케이션 또는 공식이나 스크립트 구성요소가 속하는 계산이나 계획 유형을 새로고치면 기본적으로 공식 및 스크립트 구성요소가 새로고쳐집니다. 하지만 공식 또는 스크립트 구성요소를 새로고치면 애플리케이션 목록의 상위 레벨(즉, 계산 또는 계획 유형, 애플리케이션 또는 애플리케이션 유형)이 새로고쳐지지 않습니다.

 주:

공식 및 스크립트 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙에서 생성할 수 있습니다.

공식 또는 스크립트 구성요소를 새로고치려면 시스템 뷰에서 스크립트 또는 공식을 마우스 오른쪽 버튼으로 누르고 **새로고침**을 선택합니다.

 주:

데이터베이스 아웃라인의 상위 레벨을 새로고쳐 상위 레벨에 포함된 객체를 새로고칠 수도 있습니다. 예를 들어 애플리케이션에 대한 스크립트와 공식을 새로고치려면 애플리케이션 이름을 마우스 오른쪽 버튼으로 누르고 **새로고침**을 선택합니다.

## 공식 및 스크립트 구성요소의 사용 표시

스크립트 또는 공식 구성요소의 사용을 표시할 수 있습니다.

### 주:

공식 및 스크립트 구성요소는 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 비즈니스 규칙에서 생성할 수 있습니다.

스크립트 또는 공식 구성요소의 사용을 표시하려면 다음을 수행합니다.

1. 시스템 뷰에서 스크립트 또는 공식을 마우스 오른쪽 버튼으로 누르고 **사용 표시**를 선택합니다.
2. 사용 창에서 정보를 검토하고 **확인**을 누릅니다.

## 순서도의 구성요소 작업

순서도의 구성요소에 대해 작업을 수행할 수 있습니다.

### 참조:

- [순서도의 구성요소 작업 정보](#)
- [순서도에서 구성요소 확장 및 축소](#)
- [순서도에서 구성요소 제거](#)
- [순서도에서 구성요소 복사하여 붙여넣기](#)
- [순서도에서 구성요소에 대한 참조 복사하여 붙여넣기](#)
- [순서도에서 구성요소 그룹 복사하여 붙여넣기](#)

## 순서도의 구성요소 작업 정보


순서도에서 구성요소를 확장 및 축소하여 세부정보를 표시하거나 숨기고, 구성요소를 제거하거나 템플릿으로 저장하고, 구성요소, 구성요소 그룹 및 구성요소 참조를 복사하여 붙여넣는 등의 구성요소 관련 작업을 수행할 수 있습니다. 템플릿 디자이너 및 규칙 디자이너 순서도에서 작업하는 동안 구성요소에 대한 여러 변경사항을 실행취소할 수도 있습니다.

## 순서도에서 구성요소 확장 및 축소

복합 구성요소가 많이 들어 있는 비즈니스 규칙이나 템플릿이 있는 경우 순서도에서 몇몇 구성요소를 축소하거나 확장할 수 있습니다. 순서도에서 구성요소를 축소 및 확장하여 작업할 구성요소의 표시 공간은 최대화하면서 작업하지 않을 구성요소의 표시 공간은 최소화할 수 있습니다.

순서도에서 구성요소를 축소하려면 다음을 수행합니다.

1. 시스템 뷰에서 비즈니스 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.


2. **규칙 디자이너 또는 템플리트 디자이너** 순서도에서 구성요소에 대해 다음 작업 중 하나를 수행하십시오.
  - 구성요소를 확장하려면 구성요소를 마우스 오른쪽 버튼으로 누르고 **확장**을 선택합니다.
  - 구성요소를 축소하려면 구성요소를 마우스 오른쪽 버튼으로 누르고 **축소**를 선택합니다.
3. 를 누릅니다.

## 순서도에서 구성요소 제거

비즈니스 규칙이나 템플리트의 순서도에서 조건, 멤버 범위, 데이터 범위 또는 고정 루프 구성요소를 제거하면 해당 구성요소가 삭제됩니다. 이러한 구성요소는 공유할 수 없으므로 하나의 비즈니스 규칙이나 템플리트에만 존재합니다.

순서도에서 공식이나 스크립트 구성요소를 제거하면 해당 구성요소가 삭제될 수도 있고 삭제되지 않을 수도 있습니다. *공유되지 않은* 공식이나 스크립트 구성요소를 제거하면 해당 구성요소가 삭제되고, *공유된* 공식이나 스크립트 구성요소를 제거하면 해당 구성요소가 제거된 비즈니스 규칙이나 템플리트에서만 삭제됩니다.

순서도에서 구성요소를 제거하려면 다음을 수행합니다.

1. **시스템 뷰**에서 비즈니스 규칙 또는 템플리트를 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **규칙 디자이너 또는 템플리트 디자이너**의 순서도에서 제거할 구성요소를 마우스 오른쪽 버튼으로 누르고 **제거**를 선택합니다.
3. 를 누릅니다.

## 순서도에서 구성요소 복사하여 붙여넣기

비즈니스 규칙 또는 템플리트의 순서도에서 구성요소를 복사하는 경우 동일한 비즈니스 규칙이나 템플리트의 순서도 내에 있는 다른 위치에 붙여넣을 수도 있고, 비즈니스 규칙이나 템플리트가 동일한 애플리케이션 유형에 속하는 경우 다른 비즈니스 규칙이나 템플리트의 순서도에 붙여넣을 수도 있습니다.

구성요소를 복사하여 붙여넣으려면 다음을 수행합니다.

1. **시스템 뷰**에서 비즈니스 규칙 또는 템플리트를 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **규칙 디자이너 또는 템플리트 디자이너**의 순서도에서 복사할 구성요소를 마우스 오른쪽 버튼으로 누르고 **복사**를 선택합니다.

### **팁:**

편집, 복사를 사용할 수도 있습니다.

3. 다음 태스크 중 하나를 수행합니다.
  - 구성요소를 *동일한* 비즈니스 규칙 또는 템플리트에 붙여넣으려면 순서도에서 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.

- 구성요소를 *다른* 비즈니스 규칙 또는 템플릿에 붙여넣으려면 비즈니스 규칙 또는 템플릿을 열고 순서도에서 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.

4. 를 누릅니다.

## 순서도에서 구성요소에 대한 참조 복사하여 붙여넣기

구성요소 자체를 복사하는 것과 달리, 구성요소에 대한 참조를 복사하면 구성요소에 대한 포인터만 복사됩니다. 구성요소에 대한 참조를 복사하면 구성요소 자체는 생성된 원래 위치에만 있게 됩니다.

구성요소에 대한 참조를 복사하는 경우 동일한 비즈니스 규칙이나 템플릿에 붙여넣을 수도 있고, 동일한 애플리케이션 유형에 속하는 다른 비즈니스 규칙이나 템플릿에 붙여넣을 수도 있습니다.

순서도에서 구성요소에 대한 참조를 복사하여 붙여넣으려면 다음을 수행합니다.


1. **시스템 뷰**에서 비즈니스 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. **규칙 디자이너** 또는 **템플릿 디자이너**의 순서도에서 참조를 복사할 구성요소를 마우스 오른쪽 버튼으로 누르고 **참조 복사**를 선택합니다.

### **팁:**

편집, 참조 복사를 사용할 수도 있습니다.

### **주:**

참조 복사는 다른 규칙이나 템플릿 플로우에서 사용되는 규칙 및 템플릿나 공유 구성요소(즉, 공식 및 스크립트)에만 사용할 수 있습니다.

3. 다음 태스크 중 하나를 수행합니다.
  - 구성요소 참조를 *동일한* 비즈니스 규칙 또는 템플릿에 붙여넣으려면 순서도에서 참조를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
  - 구성요소 참조를 *다른* 비즈니스 규칙 또는 템플릿에 붙여넣으려면 비즈니스 규칙 또는 템플릿을 열고 순서도에서 참조를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
4. 를 누릅니다.

## 순서도에서 구성요소 그룹 복사하여 붙여넣기

특정 구성요소에 다른 구성요소가 포함되어 있는 경우(즉, 한 구성요소 내에 여러 구성요소가 있는 경우) 구성요소 그룹을 복사한 후 같은 순서도 내의 다른 위치에 붙여넣거나 다른 순서도에 붙여넣을 수 있습니다.

순서도에서 구성요소 그룹을 복사하여 붙여넣으려면 다음을 수행합니다.


1. **시스템 뷰**에서 비즈니스 규칙 또는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.



2. 규칙 디자이너 또는 템플릿 디자이너의 순서도에서 복사할 구성요소 그룹을 마우스 오른쪽 버튼으로 누르고 **그룹 복사**를 선택합니다.

 **팁:**

편집, 그룹 복사를 사용할 수도 있습니다.

3. 다음 태스크 중 하나를 수행합니다.
  - 구성요소 그룹을 *동일한* 비즈니스 규칙 또는 템플릿에 붙여넣으려면 순서도에서 그룹을 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
  - 구성요소 그룹을 *다른* 비즈니스 규칙 또는 템플릿에 붙여넣으려면 비즈니스 규칙 또는 템플릿을 열고 순서도에서 그룹을 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다.
4.  를 누릅니다.

# 7

## 집계 저장영역 구성요소를 사용하여 비즈니스 규칙 디자인

### 참조:

- [집계 저장영역 구성요소를 사용하여 비즈니스 규칙 디자인 정보](#)  
집계 저장영역 구성요소를 사용하여 비즈니스 규칙을 디자인합니다.
- [POV 구성요소 작업](#)  
POV 구성요소를 생성하고 편집할 수 있습니다.
- [할당 구성요소 작업](#)  
할당 구성요소를 생성하고 편집할 수 있습니다.
- [POV 또는 할당 구성요소 열기](#)  
POV 또는 할당 구성요소는 해당 구성요소가 속해 있는 비즈니스 규칙의 순서도에서 엽니다. 시스템 뷰에서는 열 수 없습니다.
- [POV 또는 할당 구성요소 삭제](#)  
POV 또는 할당 구성요소가 속해 있는 비즈니스 규칙에서 해당 구성요소를 제거하면 삭제됩니다.
- [POV 또는 할당 구성요소 복사 및 붙여넣기](#)  
비즈니스 규칙에서 POV 또는 할당 구성요소를 복사한 다음 동일하거나 다른 비즈니스 규칙에 구성요소를 붙여넣을 수 있습니다.
- [POV 또는 할당 구성요소 저장](#)  
규칙 디자이너에서 POV 또는 할당 구성요소가 속해 있는 비즈니스 규칙을 저장하면 해당 구성요소가 저장됩니다.
- [집계 저장영역 공식 구성요소 작업](#)  
집계 저장영역 공식 구성요소를 생성, 열기, 편집, 삭제, 복사, 붙여넣기, 사용법을 표시합니다.

## 집계 저장영역 구성요소를 사용하여 비즈니스 규칙 디자인 정보

집계 저장영역 구성요소를 사용하여 비즈니스 규칙을 디자인합니다.

Oracle Essbase 집계 저장영역 애플리케이션의 비즈니스 규칙(Oracle General Ledger에서 사용되는 비즈니스 규칙 포함)은 Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Essbase 블록 저장영역 애플리케이션의 비즈니스 규칙과는 다른 구성요소로 구성됩니다.

Essbase 합산 저장영역 애플리케이션에서 비즈니스 규칙 디자인에 사용되는 구성요소는 다음과 같습니다.

 주:

이러한 구성요소는 Essbase 합산 저장영역 애플리케이션 및 Oracle General Ledger 애플리케이션에서만 사용할 수 있습니다.

- POV 구성요소에는 메타데이터 멤버 목록(예: 계정 목록)이 포함됩니다.

 주:

POV 구성요소를 다른 POV 구성요소 내에 중첩할 수 있습니다.

- 할당 구성요소에는 데이터베이스 아웃라인의 단일 레벨에 있는 멤버의 데이터를 아웃라인의 다른 멤버에 배포하기 위한 계산이 포함됩니다.
- 공식 구성요소에는 멤버, 함수 및 변수를 사용하여 디자인하는 계산 명령문이 포함됩니다.

구성요소를 생성할 때 작업 중인 비즈니스 규칙, 구성요소, 템플릿 및 변수를 열어 두어야 할 수 있습니다. Oracle Hyperion Calculation Manager에서는 이러한 객체를 탭 인터페이스에 표시하므로 구성요소를 생성하는 동안 탭 간에 쉽게 이동할 수 있습니다. Calculation Manager 내에서 탭 10개를 열어 둘 수 있지만 성능을 최적화하려면 동시에 여는 객체 수가 10개를 초과하지 않는 것이 좋습니다.

## POV 구성요소 작업

POV 구성요소를 생성하고 편집할 수 있습니다.

## 참조:

- [POV 구성요소 생성](#)
- [POV 구성요소 편집](#)

## POV 구성요소 생성

 주:

이러한 구성요소는 Oracle Essbase 집계 저장영역 애플리케이션 및 Oracle General Ledger 애플리케이션에서만 사용할 수 있습니다.

비즈니스 규칙 내에서 POV 구성요소를 생성합니다.

POV 구성요소를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.  
비즈니스 규칙이 규칙 디자이너에 표시됩니다.
2. POV 구성요소를 생성할 비즈니스 규칙 순서도 위치를 결정한 후 새 객체 팔레트에서 POV 구성요소를 순서도의 해당 위치로 끌어 놓습니다.

POV 객체가 안에 화살표가 있는 두 개의 원으로 표시됩니다.

3. **POV**에 POV 구성요소를 확인할 캡션을 입력합니다. 캡션은 해당 POV를 사용하는 모든 규칙의 순서도에서 구성요소 위에 표시됩니다.
4. **선택 사항:** 다음 태스크 중 하나를 수행하여 POV의 글로벌 범위를 정의합니다.

#### 주:

POV 구성요소를 생성할 비즈니스 규칙에 대해 글로벌 범위가 정의되어 있으면 [POV] 탭에 비즈니스 규칙의 멤버 선택이 기본적으로 표시됩니다. 비즈니스 규칙에 대해 글로벌 범위가 정의되었는지 여부를 확인하려면 순서도에서 [시작] 또는 [끝] 탭을 선택합니다. 그런 다음 [글로벌 범위] 탭을 눌러 멤버, 함수 또는 변수가 정의되어 있는지 확인합니다.

- 변수 선택기를 눌러 POV를 정의할 변수를 선택하거나 생성합니다. 변수를 선택하는 경우 동적으로 변수 링크를 선택하여 변경 시 변수가 동적으로 업데이트되도록 합니다.
- 멤버 선택기를 눌러 POV를 정의할 멤버를 선택합니다.
- 값 열에서 차원의 행을 눌러 POV를 정의하는 멤버 이름을 입력합니다.
- 차원의 행을 누르고 작업 아이콘을 누른 후 다음 옵션 중 하나를 선택하여 멤버를 입력합니다.

- 멤버([구성요소에 멤버 및 함수 추가 참조](#))
- 변수([변수 작업 참조](#))

Groovy 이외의 규칙인 경우, 하나 이상의 *멤버* 유형의 변수는 ASO(*집계 저장영역 옵션*) 유형의 Oracle Hyperion Planning 큐브에서만 지원되는 변수 유형입니다.

- 함수([함수 작업 참조](#))

입력한 함수는 레벨 멤버만 반환해야 하며 함수 이름 앞에 @ 기호를 포함해야 합니다. List 매개변수가 있는 함수의 경우 입력은 단일 멤버나 멤버 또는 멤버 목록을 반환하는 함수여야 합니다.

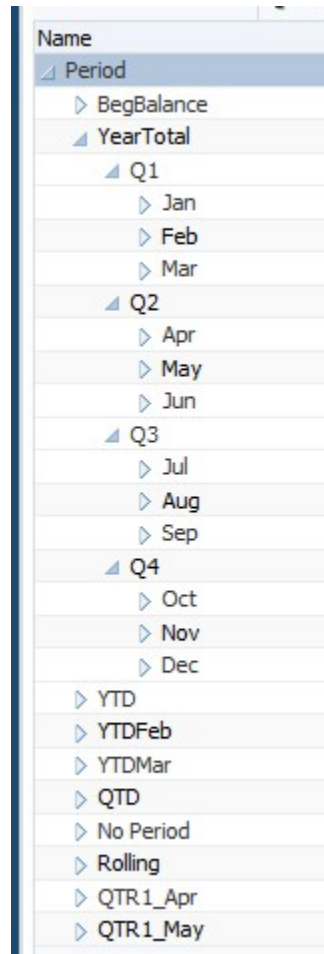
다음 함수를 입력할 수 있습니다.

- \* @ANCESTOR(*Member Name, Index|Layer*)는 입력 멤버가 제공되면 지정된 인덱스 또는 계층의 상위 멤버를 반환합니다. @ANCESTOR는 규칙 POV에서만 사용합니다.
- \* @ATTRIBUTE(*Attribute Member Name*)는 지정된 속성 멤버와 연계된 모든 기준 멤버 목록을 생성합니다.
- \* @COUSIN(*Member Name, Cousin Member Name*)은 다른 상위 멤버의 멤버와 동일한 위치에 있는 1차 하위 멤버를 반환합니다. @COUSIN은 규칙 POV에서만 사용합니다.
- \* @DISTINCT(*List1*)는 세트에서 중복 튜플을 삭제합니다. @DISTINCT는 규칙 POV에서만 사용합니다.
- \* @EXCEPT(*List1, List2*)는 두 세트 간의 차이를 포함하는 서브세트를 반환하며, 선택적으로 중복 항목을 유지합니다. @EXCEPT는 규칙 POV에서만 사용합니다.
- \* @FilterDynamic(*Dimension Name, Member Name*)은 멤버 목록에서 모든 동적 멤버를 제거합니다. @FilterDynamic은 규칙 POV에서만 사용합니다.

- \* @FilterShared(*Dimension Name, Member Name*)는 멤버 목록에서 모든 공유 멤버를 제거합니다. @FilterShared는 규칙 POV에서만 사용합니다.
- \* @FilterSharedAndDynamic(*Dimension Name, Member Name*)은 멤버 목록에서 모든 동적 멤버와 공유 멤버를 제거합니다. @FilterSharedAndDynamic은 규칙 POV에서만 사용합니다.
- \* @FIRSTCHILD(*Member Name*)는 입력 멤버의 첫번째 1차 하위를 반환합니다. @FIRSTCHILD는 규칙 POV에서만 사용합니다.
- \* @FIRSTSIBLING(*Member Name*)은 입력 멤버 상위의 첫번째 1차 하위를 반환합니다. @FIRSTSIBLING은 글로벌 POV에서만 사용합니다.
- \* @GEN(*Member Name*)은 지정된 멤버의 입력 세대 번호로 지정된 멤버를 반환합니다. @GEN은 규칙 POV에서만 사용합니다.
- \* @GENMBRS(*Member Name, Generation*)는 입력 세대 번호로 지정된 세대를 반환합니다. @GENMBRS는 규칙 POV에서만 사용합니다.
- \* @ILSIBLINGS는 지정된 멤버와 해당 왼쪽 동위 멤버를 반환합니다. @ILSIBLINGS는 규칙 POV에서만 사용합니다.
- \* @INTERSECT(*List1, List2*)는 두 입력 세트의 교차점을 반환하며, 선택적으로 중복 항목을 유지합니다. @INTERSECT는 규칙 POV에서만 사용합니다.
- \* @IRSIBLINGS는 지정된 멤버와 해당 오른쪽 동위 멤버를 반환합니다. @IRSIBLINGS는 규칙 POV에서만 사용합니다.
- \* @LAGGEN(*Member Name, Index*)은 데이터베이스 아웃라인에 있는 멤버 순서를 사용하여 동일한 세대에서 제공된 멤버 뒤의 n단계인 멤버를 반환합니다. @LAGGEN은 글로벌 POV에서만 사용합니다.
- \* @LAGLEV(*Member Name, Index*)는 데이터베이스 아웃라인에 있는 멤버 순서를 사용하여 동일한 레벨에서 제공된 멤버 뒤의 n단계인 멤버를 반환합니다. @LAGLEV는 글로벌 POV에서만 사용합니다.
- \* @LASTCHILD(*Member Name*)는 입력 멤버의 마지막 1차 하위를 반환합니다. @LASTCHILD는 규칙 POV에서만 사용합니다.
- \* @LASTSIBLING(*Member Name*)은 입력 멤버 상위의 마지막 1차 하위를 반환합니다. @LASTSIBLING은 글로벌 POV에서만 사용합니다.
- \* @LAYERGEN(*Member Name, Index*)은 지정된 멤버의 세대 기반 계층을 반환합니다.  
  
@LAYERGEN은 매개변수가 계층을 찾는 함수 내에서만 사용할 수 있습니다 (예: @ANCESTOR(*member, layer, index*)). @LAYERGEN을 단독으로 사용할 수는 없습니다. @LAYERGEN은 규칙 POV에서만 사용합니다.

다음 기간 차원에서

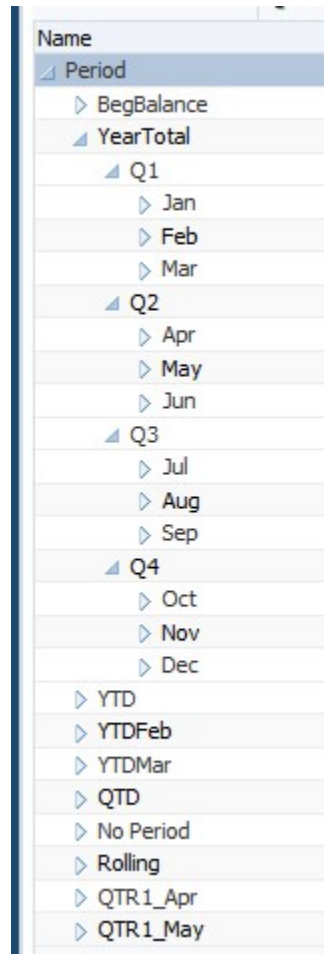
@Level0Descendants(@ANCESTOR(Jul, @LAYERGEN(Period, 3)))는 Jul, Aug, Sep를 반환합니다.




- \* @LAYERLEVEL (*Member Name*, *Index*) 은 지정된 멤버의 레벨 기반 계층을 반환합니다.

@LAYERLEVEL은 매개변수가 계층을 찾는 함수 내에서만 사용할 수 있습니다(예: @ANCESTOR(*member*, **layer**, *index*)). @LAYERLEVEL을 단독으로 사용할 수는 없습니다. @LAYERLEVEL은 규칙 POV에서만 사용합니다.

다음 기간 차원에서 @ANCESTOR(Oct, @LAYERLEVEL(Period, 0))는 멤버 Oct, Nov, Dec를 반환합니다.




- \* @LEADGEN (*Member Name, Index*)은 데이터베이스 아웃라인에 있는 멤버 순서를 사용하여 동일한 세대에서 제공된 멤버 뒤의  $n$ 단계인 멤버를 반환합니다. @LEADGEN은 글로벌 POV에서만 사용합니다.
- \* @LEADLEV (*Member Name, Index*)는 데이터베이스 아웃라인에 있는 멤버 순서를 사용하여 동일한 레벨에서 제공된 멤버 뒤의  $n$ 단계인 멤버를 반환합니다. @LEADLEV는 글로벌 POV에서만 사용합니다.
- \* @Level0Descendant (*Member Name*)는 지정된 멤버의 모든 레벨 0 하위를 포함하도록 확장됩니다.
- \* @LIST (*Argument*)는 멤버 목록을 반환합니다. @LIST는 규칙 POV에서만 사용합니다.
- \* @LSIBLINGS는 지정된 멤버의 왼쪽 동위 멤버를 반환합니다. @LSIBLINGS는 규칙 POV에서만 사용합니다.
- \* @NEXTLEVMBR (*Member Name*)은 데이터베이스 아웃라인에 있는 멤버 순서를 사용하여 동일한 레벨에서 다음 멤버를 반환합니다. @NEXTLEVMBR은 글로벌 POV에서만 사용합니다.
- \* @NEXTMBR (*Member Name*)은 데이터베이스 아웃라인에 있는 멤버 순서를 사용하여 동일한 세대 또는 레벨에서 다음 멤버를 반환합니다. @NEXTMBR은 규칙 POV에서만 사용합니다.

- \* @PARENT (*Member Name*)는 멤버의 상위를 반환합니다. @PARENT는 규칙 POV에서만 사용합니다.
  - \* @PREVLEVMBR (*Member Name*)은 데이터베이스 아웃라인에 있는 멤버 순서를 사용하여 동일한 레벨에서 이전 멤버를 반환합니다. @PREVLEVMBR은 글로벌 POV에서만 사용합니다.
  - \* @PREVMBR (*Member Name*)은 데이터베이스 아웃라인에 있는 멤버 순서를 사용하여 동일한 세대 또는 레벨에서 이전 멤버를 반환합니다. @PREVMBR은 규칙 POV에서만 사용합니다.
  - \* @RSIBLINGS는 지정된 멤버의 오른쪽 동위 멤버를 반환합니다. @RSIBLINGS는 규칙 POV에서만 사용합니다.
  - \* @Siblings (*Member Name*)는 지정된 멤버의 모든 동위를 포함하도록 확장됩니다.
  - \* @UDA (*Dimension Name, User-Defined Attribute String*)는 Essbase 서버에서 UDA(사용자정의 속성)로 정의한 공통 속성을 기준으로 멤버를 선택합니다.
5. 차원에 대해 선택한 멤버에 대한 설명을 입력하려면 **설명**을 누릅니다.
  6. **그리드 재설정**을 눌러 그리드에 입력한 내용을 모두 지웁니다.
  7. 를 누릅니다.

## POV 구성요소 편집

POV 구성요소의 글로벌 범위를 구성하는 멤버, 변수 및 함수를 편집할 수 있습니다. 순서도에서 구성요소 위에 표시되는 캡션과 POV의 각 차원에 선택한 값에 대해 입력된 설명도 편집할 수 있습니다.

POV 구성요소를 편집하려면 다음을 수행합니다.

1. **시스템 뷰**에서 POV가 포함된 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙 디자이너의 순서도에서 편집할 POV 구성요소를 선택하여 해당 등록정보를 표시합니다. POV 구성요소의 다음 등록정보를 모두 편집할 수 있습니다. [POV 구성요소 생성](#)을 참조하십시오.
  - 비즈니스 규칙의 순서도에서 POV 구성요소 위에 표시되는 캡션
  - POV를 정의하는 멤버, 변수 및 함수
  - POV 구성요소에 사용된 변수가 변경 시 동적으로 업데이트되는지 여부
  - POV의 글로벌 범위를 정의하는 차원 및 멤버에 대해 설명이 입력되는지 여부
  - POV 구성요소가 속해 있는 비즈니스 규칙을 검증하거나 실행할 때 POV 구성요소의 멤버 값이 계산되는지 여부
3. 를 누릅니다.

## 할당 구성요소 작업

할당 구성요소를 생성하고 편집할 수 있습니다.

**참조:**

- [할당 구성요소 생성](#)



- 할당 구성요소 편집

## 할당 구성요소 생성

### 주:

이러한 구성요소는 Oracle Essbase 집계 저장영역 애플리케이션 및 Oracle General Ledger 애플리케이션에서만 사용할 수 있습니다.

할당 구성요소를 사용하면 멤버에서 해당 멤버의 레벨 0 하위 멤버로 데이터를 분산할 수 있습니다. 비즈니스 규칙 내에서 할당 구성요소를 생성하며, 이 구성요소는 해당 비즈니스 규칙에만 존재하고 여러 비즈니스 규칙 사이에 공유할 수 없습니다.

할당 구성요소를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.  
비즈니스 규칙이 규칙 디자이너에 표시됩니다.
2. 할당 구성요소를 생성할 비즈니스 규칙 순서도 위치를 결정한 후 새 객체 팔레트에서 할당 구성요소를 순서도의 해당 위치로 끌어 놓습니다.

### 주:

POV 구성요소를 다른 POV 구성요소 내에 놓으면 두 번째 POV가 첫 번째 (상위) POV의 멤버, 변수 및 함수를 상속합니다.

3. 할당 마법사의 POV에 나열된 차원 중 할당 과정에서 그대로 유지하려는 각 차원에 대해 다음 태스크 중 하나를 수행합니다.
  - 사전 정의된 선택 사용에서 사전 정의된 선택을 선택하여, 나열된 차원을 값으로 채웁니다.
  - 멤버 선택기 아이콘을 눌러 나열된 각 차원의 멤버와 변수를 선택합니다.

### 주:

멤버 선택기의 차원에서 마법사의 현재 단계에 나열된 차원을 선택할 수 있습니다. 따라서 마법사의 현재 단계에 나열된 차원의 멤버와 함수를 선택할 수 있습니다.

선택한 모든 멤버가 적합한 레벨 0 멤버인지 확인하십시오.

- 목록에서 차원을 선택하고 작업을 눌러 멤버나 변수를 선택합니다. **멤버 선택**, **변수**, **함수**, **스마트 목록** 및 **Planning 공식 표현식**을 사용하여 구성요소 디자인을 참조하십시오.
4. 할당 마법사의 소스에 나열된 각 차원에 대해 다음 태스크 중 하나를 수행하여 할당할 데이터가 있는 멤버를 선택합니다.

 주:

나열된 각 차원의 멤버를 선택해야 합니다.  
소스 멤버는 레벨 0이 아닌 멤버일 수 있습니다.

- 사전 정의된 선택 사용에서 사전 정의된 선택을 선택하여, 나열된 차원을 값으로 채웁니다.

 주:

나열된 각 차원의 값이 사전 정의된 선택에 의해 입력되지 않는 경우, 비어 있는 차원에 대해 값을 입력해야 합니다.

- **멤버 선택기** 아이콘을 눌러 나열된 각 차원에 대한 멤버를 선택합니다.
- 목록에서 차원을 선택하고 **작업**을 눌러 멤버나 변수를 선택합니다. **멤버 선택, 변수, 함수, 스마트 목록 및 Planning 공식 표현식을 사용하여 구성요소 디자인**을 참조하십시오.

 주:

이 할당 구성요소 단계에서는 함수를 사용할 수 없습니다.

- 특정 값을 할당하려면 **선택 사항**에서 위의 선택 항목이 아니라 할당할 금액을 입력합니다.
5. 할당할 소스 크기가 0이면 드롭다운에서 다음 옵션 중 하나를 선택합니다.
    - 다음 풀 레코드를 선택합니다.
    - 할당 처리를 중지합니다.
  6. 할당 범위에서 다음을 수행합니다.
    - 할당에 사용할 차원의 상위 멤버를 입력합니다.  
상위 멤버를 입력하려면 다음 태스크 중 하나를 수행합니다.
      - 사전 정의된 선택 사용에서 사전 정의된 선택을 선택하여, 나열된 차원을 값으로 채웁니다.
      - **멤버 선택기** 아이콘을 눌러 데이터를 할당할 차원의 상위 멤버를 선택합니다.
      - 상위 멤버를 입력하거나 목록에서 차원을 선택하고 **작업** 아이콘을 눌러 데이터를 할당할 기본 차원의 상위 멤버를 선택합니다. **멤버 선택, 변수, 함수, 스마트 목록 및 Planning 공식 표현식을 사용하여 구성요소 디자인**을 참조하십시오.  
이 데이터는 데이터베이스 아웃라인에서 상위 멤버 아래의 레벨 0 멤버(아웃라인에서 하위 멤버가 없는 최하위 멤버)에 할당됩니다.
    - 할당 프로세스 전에 영역을 지울지 여부를 선택합니다.  
영역을 지우도록 **예**를 선택하는 경우 **영역 지우기** 필드에 할당을 실행하기 전에 지울 멤버를 입력합니다(8단계 참조).
  7. 타겟의 나머지 차원에서 데이터를 할당할 레벨 0 멤버를 선택합니다. 다음 중 한 가지 태스크를 수행합니다.
    - 사전 정의된 선택 사용에서 사전 정의된 선택을 선택하여, 나열된 차원을 값으로 채웁니다.

- **멤버 선택기** 아이콘을 눌러 나열된 각 차원의 멤버를 선택합니다.
  - 목록에서 차원을 선택하고 **작업** 아이콘을 눌러 멤버나 변수를 선택합니다. **멤버 선택, 변수, 함수, 스마트 목록 및 Planning 공식 표현식을 사용하여 구성요소 디자인을 참조하십시오.**
8. **영역 지우기**에 할당 프로세스 전에 논리적 지우기로 지울 레벨 0 멤버를 입력합니다. 다음 중 한 가지 태스크를 수행합니다.
- **사전 정의된 선택 사용**에서 사전 정의된 선택을 선택하여, 나열된 차원을 값으로 채웁니다.
  - **멤버 선택기** 아이콘을 눌러 나열된 각 차원의 멤버를 선택합니다.
  - 목록에서 차원을 선택하고 **작업** 아이콘을 눌러 멤버나 변수를 선택합니다. **멤버 선택, 변수, 함수, 스마트 목록 및 Planning 공식 표현식을 사용하여 구성요소 디자인을 참조하십시오.**



**주:**

영역 지우기가 표시되지 않는 경우 할당 전에 데이터를 지우도록 할당 범위에서 **예**를 선택하지 않은 것입니다.

9. **오프셋**에서 다음 태스크 중 하나를 수행합니다.
- **사전 정의된 선택 사용**에서 사전 정의된 선택을 선택하여, 나열된 차원을 값으로 채웁니다.
  - **멤버 선택기** 아이콘을 눌러 나열된 각 차원의 멤버를 선택합니다.
  - 목록에서 차원을 선택하고 **작업** 아이콘을 눌러 멤버나 변수를 선택합니다. **멤버 선택, 변수, 함수, 스마트 목록 및 Planning 공식 표현식을 사용하여 구성요소 디자인을 참조하십시오.**



**주:**

오프셋에 대한 멤버를 지정해야 합니다. 이 항목을 비워 둘 수 없습니다.

10. **선택사항**: 할당에서 제외할 멤버를 **제외**에서 선택합니다. 다음 중 한 가지 태스크를 수행합니다.
- **사전 정의된 선택 사용**에서 사전 정의된 선택을 선택하여, 나열된 차원을 값으로 채웁니다.
  - **멤버 선택기** 아이콘을 눌러 나열된 각 차원의 멤버를 선택합니다.
  - 목록에서 차원을 선택하고 **작업** 아이콘을 눌러 멤버나 변수를 선택합니다. **멤버 선택, 변수, 함수, 스마트 목록 및 Planning 공식 표현식을 사용하여 구성요소 디자인을 참조하십시오.**
11. **기준**에서 다음 태스크를 수행합니다.
- a. 할당 방법을 선택하여 데이터를 할당하는 방법을 지정합니다.
- **균등 할당**을 선택하여 데이터 값을 할당 범위에 균등하게 할당합니다. 그런 다음, **균등 방법의 기준 옵션**에서 기준이 음수 또는 0이거나 기준에 값이 누락된 경우 또는 모든 멤버가 제외된 경우 수행할 작업을 지정합니다.

- **드라이버를 사용하여 할당**을 선택하여 할당 범위에서 각 멤버에 적용될 백분율을 계산합니다. 그런 다음, **기준 옵션**에서 기준이 음수이거나 0과 같은 경우 수행할 작업을 지정합니다.
  - b. 지정하지 않은 차원 멤버는 이전에 정의한 POV에서 상속되지만, 이러한 POV 선택 사항을 다음 태스크 중 하나를 수행하여 재정의할 수 있습니다.
    - **사전 정의된 선택 사용**에서 사전 정의된 선택을 선택하여, 나열된 차원을 값으로 채웁니다.
    - **멤버 선택기** 아이콘을 눌러 나열된 각 차원에 대한 멤버를 선택합니다.
    - 목록에서 차원을 선택하고 **작업** 아이콘을 눌러 멤버나 변수를 선택합니다. **멤버 선택, 변수, 함수, 스마트 목록 및 Planning 공식 표현식을 사용하여 구성요소 디자인**을 참조하십시오.
  - c. 다음을 누릅니다.
12. 반올림에서 다음 단계를 완료합니다.
- a. 이 할당에 사용할 소수 자릿수를 입력하거나 **작업** 아이콘을 눌러 이 값을 나타내는 멤버나 변수를 선택합니다.
  - b. 반올림 오차를 배치할 위치를 선택합니다.
    - 반올림 오차를 배치할 멤버를 지정하려면 **위치 정의**를 선택합니다.
    - **가장 큰 값 사용**을 선택하여 데이터 값을 가장 큰 값으로 반올림합니다
    - 데이터 값을 가장 작은 값으로 반올림하려면 **가장 작은 값 사용**을 선택합니다.
    - 할당된 데이터 값을 있는 그대로 사용하려면 **반올림 오류 무시**를 선택합니다.
13. 이전 단계에서 **위치 정의**를 선택한 경우 **반올림 멤버**에서 다음 태스크 중 하나를 수행합니다.
- **사전 정의된 선택 사용**에서 사전 정의된 선택을 선택하여, 나열된 차원을 값으로 채웁니다.
  - **멤버 선택기** 아이콘을 눌러 나열된 각 차원에 대한 멤버를 선택합니다.
  - 목록에서 차원을 선택하고 **작업** 아이콘을 눌러 멤버나 변수를 선택합니다. **멤버 선택, 변수, 함수, 스마트 목록 및 Planning 공식 표현식을 사용하여 구성요소 디자인**을 참조하십시오.

 **주:**

이 단계에서 선택한 멤버는 할당 범위에 속해야 합니다.

14. **완료**를 누릅니다.

## 할당 구성요소 편집

할당 구성요소가 속해 있는 비즈니스 규칙을 열면 해당 구성요소를 편집할 수 있습니다. 비즈니스 규칙이 규칙 디자이너에 표시되는 경우 비즈니스 규칙의 순서도에서 할당 구성요소를 선택하여 해당 등록정보를 볼 수 있습니다.

할당 구성요소를 편집하려면 다음을 수행합니다.

1. **시스템 뷰**에서 할당 구성요소가 포함된 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙 디자이너의 순서도에서 편집할 할당 구성요소를 선택하여 해당 등록정보를 표시합니다. 할당 구성요소의 다음 등록정보를 모두 편집할 수 있습니다. **할당 구성요소 생성**을 참조하십시오.

- 할당할 데이터가 있는 멤버
- 데이터를 할당할 레벨 0 멤버
- 할당할 데이터 및 데이터 양
- 할당된 총 데이터 양을 오프셋 멤버에 쓸지 여부
- 데이터를 균일하게 할당할지 또는 드라이버를 사용하여 각각 다른 양을 할당할지 여부
- 할당된 데이터를 반올림할지 여부 및 반올림하는 경우 반올림하는 방법

3.  를 누릅니다.

## POV 또는 할당 구성요소 열기

POV 또는 할당 구성요소는 해당 구성요소가 속해 있는 비즈니스 규칙의 순서도에서 엽니다. 시스템 뷰에서는 열 수 없습니다.

POV 또는 할당 구성요소를 열려면 다음을 수행합니다.


1. 시스템 뷰에서 구성요소가 포함된 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.  
비즈니스 규칙이 규칙 디자이너에 표시됩니다.
2. 규칙이 열려 있는 경우 규칙의 순서도에서 POV 구성요소를 누르거나 할당 구성요소를 두 번 눌러 엽니다.

## POV 또는 할당 구성요소 삭제

POV 또는 할당 구성요소가 속해 있는 비즈니스 규칙에서 해당 구성요소를 제거하면 삭제됩니다.

POV 또는 할당 구성요소는 하나의 비즈니스 규칙에서만 사용될 수 있으므로 해당 구성요소가 속해 있는 비즈니스 규칙에서만 제거하면 삭제됩니다.


POV 또는 할당 구성요소를 삭제하려면 다음을 수행합니다.

1. 시스템 뷰에서 POV 또는 할당 구성요소가 포함된 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 규칙 디자이너의 순서도에서 삭제할 POV 또는 할당 구성요소를 선택합니다.
3. POV 또는 할당 구성요소를 마우스 오른쪽 버튼으로 누르고 **제거**를 선택합니다.  
할당 구성요소가 POV 구성요소 내에 있는 경우 POV 구성요소를 제거하면 할당 구성요소가 제거됩니다.
4.  를 누릅니다.

## POV 또는 할당 구성요소 복사 및 붙여넣기

비즈니스 규칙에서 POV 또는 할당 구성요소를 복사한 다음 동일하거나 다른 비즈니스 규칙에 구성요소를 붙여넣을 수 있습니다.


비즈니스 규칙 순서도에서 POV 또는 할당 구성요소를 복사하여 붙여넣으려면 다음을 수행합니다.

1. 시스템 뷰에서 POV 또는 할당 구성요소가 포함된 규칙을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.  
비즈니스 규칙이 규칙 디자이너에 표시됩니다.
2. 규칙 디자이너의 비즈니스 규칙 순서도에서 복사할 POV 또는 할당 구성요소를 마우스 오른쪽 버튼으로 누르고 복사를 선택하여 해당 구성요소만 복사하거나 그룹 복사를 선택하여 해당 구성요소와 이 구성요소에 포함된 모든 구성요소를 복사합니다.
3. 다음 태스크 중 하나를 수행합니다.
  - 구성요소를 동일한 비즈니스 규칙에 붙여넣으려면 순서도에서 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 붙여넣기를 선택합니다.
  - 구성요소를 다른 비즈니스 규칙에 붙여넣으려면 비즈니스 규칙을 열고 순서도에서 구성요소를 붙여넣으려는 위치를 마우스 오른쪽 버튼으로 누르고 붙여넣기를 선택합니다.
4.  를 누릅니다.

## POV 또는 할당 구성요소 저장

규칙 디자이너에서 POV 또는 할당 구성요소가 속해 있는 비즈니스 규칙을 저장하면 해당 구성요소가 저장됩니다.

공식 구성요소와 달리 POV 및 할당 구성요소는 해당 구성요소가 생성된 비즈니스 규칙에 독립적으로 존재할 수 없습니다.

POV 또는 할당 구성요소를 저장하려면 디자인 완료 후  를 누릅니다.

## 집계 저장영역 공식 구성요소 작업

집계 저장영역 공식 구성요소를 생성, 열기, 편집, 삭제, 복사, 붙여넣기, 사용법을 표시합니다.

참조:

- 집계 저장영역 공식 구성요소 생성
- 집계 저장영역 공식 구성요소 열기
- 집계 저장영역 공식 구성요소 편집
- 집계 저장영역 공식 구성요소 삭제
- 집계 저장영역 공식 구성요소 복사 및 붙여넣기
- 다른 애플리케이션이나 데이터베이스에 집계 저장영역 공식 구성요소 복사
- 집계 저장영역 공식 구성요소의 사용 표시

## 집계 저장영역 공식 구성요소 생성

 주:

집계 저장영역 공식 구성요소는 Oracle Essbase 집계 저장영역 애플리케이션 및 Oracle General Ledger 애플리케이션에서만 사용할 수 있습니다.

집계 저장영역 공식 구성요소는 공식 계산 문으로 구성됩니다. 공식의 계산 문을 생성하려면 멤버 및 함수를 입력하거나 선택합니다. 공식을 생성하면 공식의 각 계산 문이 구성요소 디자이너에서 그리드 내의 행에 나열됩니다.

시스템 뷰나 규칙 디자이너 내에서 공식 구성요소를 생성할 수 있습니다. 공식 구성요소는 데이터베이스에서 독립 객체로 존재하므로 비즈니스 규칙 사이에 공유할 수 있습니다.

집계 저장영역 애플리케이션에 대한 공식 구성요소를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 공식을 마우스 오른쪽 버튼으로 누르고 **새로 작성**을 선택합니다.
2. 공식 이름 및 애플리케이션 유형을 입력합니다.
3. 애플리케이션을 선택합니다.

애플리케이션 이름은 적합한 Essbase 합산 저장영역 또는 Oracle General Ledger 애플리케이션이어야 합니다.

4. 데이터베이스를 선택합니다.

#### 주:

[공식]을 마우스 오른쪽 버튼으로 누르고 [새로 작성]을 선택하여 새 공식을 생성하면 [새 공식] 대화상자에 시스템 뷰에서 작업 중인 애플리케이션 유형, 애플리케이션 및 데이터베이스가 입력됩니다.

5. 확인을 누릅니다.
6. 구성요소 디자이너의 **등록정보**에서 다음 단계를 완료합니다.
  - a. **선택 사항:** 기본적으로 공식은 생성될 때 공유됩니다. [공유] 확인란을 선택하거나 선택 취소할 수 없습니다.

#### 팁:

공유되지 않는 공식을 생성하려면 비즈니스 규칙을 열고 새 공식 구성요소를 비즈니스 규칙 순서도에 끌어 놓습니다. [공유] 확인란이 선택되어 있지 않습니다. 공식을 공유하려면 [공유] 확인란을 선택합니다.

- b. **선택 사항:** 새 이름을 최대 50자까지 입력하여 공식 이름을 편집합니다. 이름은 기본적으로 [새 공식] 대화상자에서 가져옵니다.

#### 주:

Essbase 합산 저장영역 및 Oracle General Ledger 애플리케이션의 멤버 이름은 고유하지 않아도 됩니다. 고유하지 않은 이름이 있으면 멤버가 저장된 객체를 검증할 때 플래그가 지정됩니다. 이 경우 멤버의 전체 경로를 입력해야 합니다. 전체 경로 구문은 [차원 이름]. [상위 이름]. [멤버 이름]입니다.

- c. **선택 사항:** 공식에 대한 설명을 최대 255자까지 입력합니다.
  - d. **선택 사항:** 공식의 캡션을 입력합니다. 캡션은 규칙 디자이너 순서도에서 공식 아래에 표시됩니다.

 주:

캡션을 입력하지 않으면 순서도에 구성요소 이름이 표시됩니다.

- e. **선택 사항:** 공식에 대한 설명을 입력합니다. 예를 들어 공식 구성요소의 사용 방법을 알리는 설명을 사용할 수 있습니다.
- 7. 공식에 공식의 캡션을 입력합니다.
- 8. **선택 사항:** 오프셋 멤버에서 공식 구성요소에 있는 모든 공식의 **합계** 금액을 오프셋 멤버나 횡차원 멤버에 기록하려는 경우 **출입표** 아이콘을 눌러 멤버를 선택합니다.

 주:

아래에서 생성하는 공식 내에 오프셋 멤버를 수동으로 정의할 수 있습니다. 공식 구성요소에 정의한 오프셋은 계산된 모든 금액의 합계로 계산됩니다.

- 9. 공식문을 생성하려면 등호 **왼쪽**의 첫 번째 명령문 행을 누릅니다. 멤버 또는 횡차원 멤버 선택을 입력하거나 **작업**을 눌러 다음을 선택합니다.
  - 변수([변수 작업 참조](#))
  - 멤버([구성요소에 멤버 및 함수 추가 참조](#))

 주:

멤버의 횡차원 선택을 입력하려면 각 멤버 이름을 오른쪽 화살표로 구분하여 입력합니다 (예: mem1->mem2->mem3).

- 10. 공식문을 완료하려면 등호 **오른쪽**의 행을 누릅니다. 멤버 또는 횡차원 멤버 선택을 입력하거나 **작업**을 눌러 다음을 선택합니다.
  - 변수([변수 작업 참조](#))
  - 멤버([구성요소에 멤버 및 함수 추가 참조](#))
- 11. **선택 사항:** 각 공식문 행에 대해 **설명** 아이콘을 눌러 공식문에 대한 설명을 입력합니다. **확인**을 누릅니다.
- 12. **사용**에서 공식 구성요소를 사용하는 규칙을 볼 수 있습니다.

 주:

이 탭의 정보는 편집할 수 없습니다.

- 13.  를 누릅니다.

## 집계 저장영역 공식 구성요소 열기

시스템 뷰에서 또는 공식 구성요소를 사용하는 비즈니스 규칙의 규칙 디자이너 순서도에서 집계 저장영역 공식 구성요소를 열 수 있습니다.



집계 저장영역 공식 구성요소를 열려면 **시스템 뷰**에서 공식을 확장한 후 다음 태스크 중 하나를 수행합니다.

- 열려는 공식을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
- 열려는 공식을 두 번 누릅니다.

공식 구성요소가 구성요소 디자이너에서 열립니다.


#### 주:

비즈니스 규칙 내에서 공식 구성요소를 열려면 공식 구성요소를 마우스 오른쪽 버튼으로 누르고 [열기]를 선택하거나 구성요소를 두 번 눌러 규칙 순서도에서 엽니다.

## 집계 저장영역 공식 구성요소 편집

집계 저장영역 공식 구성요소를 구성하는 공식문과 공식 구성요소의 설명(comment), 캡션, 이름, 설명(description) 등을 편집합니다.

집계 저장영역 공식 구성요소를 편집하려면 다음을 수행합니다.

1. **시스템 뷰**에서 공식 구성요소를 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 구성요소 디자이너에서 공식 구성요소의 다음 등록정보를 모두 편집할 수 있습니다. [집계 저장영역 공식 구성요소 생성](#)을 참조하십시오.
  - 캡션
  - 공식문
  - 이름
  - 설명
  - 설명
3. 를 누릅니다.

## 집계 저장영역 공식 구성요소 삭제

어떤 비즈니스 규칙에서도 사용되지 않는 경우에만 집계 저장영역 공식 구성요소를 삭제할 수 있습니다. 비즈니스 규칙에 공식 구성요소가 사용되고 있는지 여부를 확인하려면 공식 구성요소 사용을 표시하면 됩니다. [집계 저장영역 공식 구성요소의 사용 표시](#)를 참조하십시오.

비즈니스 규칙에서 공식 구성요소를 사용하고 있지만 더 이상 사용할 필요가 없는 경우, 규칙에서 구성요소를 제거한 다음 해당 공식 구성요소를 삭제합니다. 공식 구성요소를 비즈니스 규칙에 사용하고 있지만 더 이상 비즈니스 규칙이 필요하지 않은 경우 비즈니스 규칙을 삭제할 수 있습니다.

공식 구성요소가 비즈니스 규칙에 사용되지 않는 경우 해당 구성요소를 삭제할 수 있습니다.

집계 저장영역 공식 구성요소를 삭제하려면 다음을 수행합니다.

1. **시스템 뷰**에서 공식을 마우스 오른쪽 버튼으로 누르고 **사용 표시**를 선택합니다.
2. 비즈니스 규칙에서 공식 구성요소를 사용하고 있지 않은지 확인하십시오.

집계 저장영역 공식 구성요소의 사용 표시를 참조하십시오.

3. 공식 구성요소를 사용하고 있는 비즈니스 규칙에서 해당 구성요소를 제거합니다.  
순서도에서 구성요소 제거를 참조하십시오.
4. 공식을 다시 마우스 오른쪽 버튼으로 누르고 삭제를 선택합니다.
5. 공식의 삭제를 확인합니다.

## 집계 저장영역 공식 구성요소 복사 및 붙여넣기


규칙에서 집계 저장영역 공식 구성요소를 복사하여 같거나 다른 규칙에 붙여넣을 수 있습니다. 또한 공식 구성요소에서 그리드의 콘텐츠를 복사하여 같거나 다른 공식 구성요소에 붙여넣을 수 있습니다. 공식 구성요소를 복사하여 다른 공식 구성요소나 다른 구성요소 유형에 붙여넣을 수는 없습니다.

집계 저장영역 공식 구성요소를 복사하여 붙여넣으려면 다음을 수행합니다.

1. 시스템 뷰에서 공식 구성요소가 포함된 규칙을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 규칙 디자이너 순서도에서 복사할 공식 구성요소를 마우스 오른쪽 버튼으로 누르고 복사를 선택합니다.

### 주:

복사할 구성요소가 공유된 경우 편집, 참조 복사를 사용하여 구성요소 자체가 아니라 공유 구성요소에 대한 참조를 복사할 수 있습니다. **비즈니스 규칙 공식 또는 스크립트 구성요소에 대한 참조 복사 및 붙여넣기**를 참조하십시오.

3. 다음 태스크 중 하나를 수행합니다.
  - 공식 구성요소를 동일한 비즈니스 규칙 순서도에 붙여넣으려면 순서도에서 해당 위치를 마우스 오른쪽 버튼으로 누르고 붙여넣기를 선택합니다.
  - 공식 구성요소를 다른 비즈니스 규칙 순서도에 붙여넣으려면 구성요소를 붙여넣으려는 비즈니스 규칙을 열고 위치를 마우스 오른쪽 버튼으로 누른 다음 붙여넣기를 선택합니다.
4.  를 누릅니다.

## 다른 애플리케이션이나 데이터베이스에 집계 저장영역 공식 구성요소 복사

한 애플리케이션에서 다른 애플리케이션 및 데이터베이스로 또는 한 데이터베이스에서 같은 애플리케이션의 다른 데이터베이스로 집계 저장영역 공식 구성요소를 복사할 수 있습니다.

집계 저장영역 공식 구성요소를 복사하려면 다음을 수행합니다.

1. 시스템 뷰에서 복사할 공식 구성요소를 마우스 오른쪽 버튼으로 누르고 복사 위치를 선택합니다.
2. 다른 이름으로 저장에서 다음 태스크 중 하나를 수행합니다.
  - 공식 구성요소를 다른 애플리케이션으로 복사하려면 애플리케이션에 이름을 입력합니다.

- 공식 구성요소를 다른 애플리케이션 및 데이터베이스로 복사하려면 **애플리케이션 및 데이터베이스**에 이름을 입력합니다.
  - 공식 구성요소를 동일한 애플리케이션 내 다른 데이터베이스로 복사하려면 **데이터베이스**에 이름을 입력합니다.
3. **확인**을 누릅니다.

공식 구성요소를 복사한 애플리케이션 및 데이터베이스의 [공식] 노드에 해당 구성요소가 표시됩니다.

 **주:**

공식 구성요소를 복사한 애플리케이션 또는 데이터베이스 노드를 새로고쳐야 할 수도 있습니다. 애플리케이션 또는 데이터베이스 노드를 마우스 오른쪽 버튼으로 누르고 [새로고침]을 선택합니다.

## 집계 저장영역 공식 구성요소의 사용 표시

시스템 뷰에서 공식 구성요소의 사용을 표시하여 집계 저장영역 공식 구성요소를 사용 중인 비즈니스 규칙 및 비즈니스 규칙에 대한 기타 정보를 볼 수 있습니다.

집계 저장영역 공식 구성요소의 사용을 표시하려면 다음을 수행합니다.

1. **시스템 뷰**에서 사용을 표시할 공식을 마우스 오른쪽 버튼으로 누르고 **사용 표시**를 선택합니다.
2. 공식 구성요소에 대해 다음 정보를 볼 수 있습니다.
  - 공식 구성요소를 사용하는 비즈니스 규칙의 이름
  - 공식 구성요소를 사용하는 비즈니스 규칙의 애플리케이션 이름
  - 공식 구성요소를 사용하는 비즈니스 규칙의 데이터베이스 이름
  - 공식 구성요소의 소유자
  - 공식 구성요소를 사용하는 비즈니스 규칙이 배포되었는지 여부
  - 공식 구성요소를 사용하는 비즈니스 규칙이 검증되었는지 여부
  - 공식 구성요소를 사용하는 비즈니스 규칙의 설명

 **주:**

구성요소 디자이너의 [사용] 탭에서도 공식 구성요소의 사용을 볼 수 있습니다.

# 8

## 멤버 선택, 변수, 함수, 스마트 목록 및 Planning 공식 표현식을 사용하여 구성요소 디자인

### 참조:

- **멤버 선택, 변수, 함수 및 스마트 목록 정보**  
구성요소를 사용하여 비즈니스 규칙을 디자인하는 것처럼, 멤버, 변수 및 함수를 사용하여 구성요소를 디자인합니다.
- **구성요소에 멤버 및 함수 추가**  
공식 구성요소, 스크립트 구성요소, 조건 구성요소, 멤버 및 데이터 범위 구성요소에 멤버를 추가합니다.
- **변수 작업**  
비즈니스 규칙 및 템플릿을 디자인할 때 구성요소에 변수를 사용합니다.
- **함수 작업**  
함수는 데이터 값이나 멤버를 반환하는 멤버 공식을 정의합니다.
- **Calculation Manager에서 지원되는 Essbase 함수**  
다음은 블록 저장영역 애플리케이션의 Oracle Hyperion Calculation Manager에서 지원되는 Oracle Essbase 함수입니다.
- **사용자정의 함수 작업**  
사용자정의 함수를 사용하여 데이터 복사 및 익스포트, 텍스트 문자열에 작은따옴표 또는 큰따옴표 추가 및 제거, 두 텍스트 문자열 비교, 다른 형식으로 날짜 변환과 같은 태스크를 수행합니다.
- **구성요소에 함수 삽입**  
선택할 수 있는 함수 유형은 작업 중인 애플리케이션 유형 및 구성요소 유형에 따라 다릅니다.
- **스마트 목록 작업**  
스마트 목록은 사용자가 Oracle Hyperion Planning 애플리케이션의 Planning 데이터 양식 셀에서 액세스하는 사용자정의 드롭다운 목록입니다.
- **스마트 목록 삽입**  
Oracle Hyperion Planning 비즈니스 규칙, 공식 구성요소 또는 스크립트 구성요소에서 스마트 목록을 사용합니다.
- **Planning 공식 표현식 작업**  
Oracle Hyperion Calculation Manager 그래픽 또는 스크립트 규칙에서 Oracle Hyperion Planning 공식 표현식을 사용합니다.

## 멤버 선택, 변수, 함수 및 스마트 목록 정보

구성요소를 사용하여 비즈니스 규칙을 디자인하는 것처럼, 멤버, 변수 및 함수를 사용하여 구성요소를 디자인합니다.

공식, 스크립트, 조건, 멤버 및 데이터 범위, 멤버 블록 구성요소의 멤버 선택을 사용하여 멤버 목록을 반환하는 함수 및 멤버를 선택하거나(Oracle Hyperion Planning 및 Oracle Essbase 애플리케이션의 경우) 멤버 및 멤버 목록을 선택합니다(Oracle Hyperion Financial Management 애플리케이션의 경우). [구성요소에 멤버 및 함수 추가](#)를 참조하십시오.

변수를 사용하여 공식, 루프 및 조건 구성요소에 대한 공식을 작성합니다. 변수에는 비즈니스 규칙이 실행되면 계산되는 실행 변수와 좀 더 복잡한 공식이나 함수를 대신하는 대체 변수의 두 가지 유형이 있습니다. [변수 작업](#)을 참조하십시오.

다음 유형의 변수를 생성할 수 있습니다.

- 특정 애플리케이션 유형에 속하는 모든 애플리케이션에서 사용하기 위한 글로벌 변수
- 한 애플리케이션에서만 사용하기 위한 애플리케이션 변수
  - 한 계획 유형 또는 데이터베이스에서 사용하기 위한 계획 유형 또는 데이터베이스 변수
- 한 비즈니스 규칙에서만 사용하기 위한 규칙 변수

함수는 루프, 조건 및 공식 구성요소에서 사용할 수 있는 사전 정의된 공식입니다. 구성요소에서 사용할 수 있는 함수는 Financial Management, Planning 및 Essbase 애플리케이션에 대해 서로 다릅니다. Financial Management, Planning 및 Essbase 애플리케이션에 사용할 수 있는 함수 유형에 대한 설명을 보려면 [함수 작업](#)을 참조하십시오. Financial Management 구성요소에서 사용할 수 있는 함수의 전체 목록 및 설명을 보려면 *Oracle Hyperion Financial Management 관리자 가이드*를 참조하십시오. Planning 및 Essbase 애플리케이션에서 사용할 수 있는 함수의 전체 목록 및 설명을 보려면 *Oracle Essbase Technical Reference*를 참조하십시오. 함수를 사용하여 다음과 같은 계산을 수행할 수 있습니다.

- 날짜 문자열을 숫자로 변환
- 범위 내에서 멤버의 평균 값 계산
- 특정 기간 동안의 자산에 대한 감가상각 계산
- 시간 차원에서 멤버의 기간누계 값 계산

Planning에서 데이터 양식 셀에 옵션을 입력하는 대신 스마트 목록을 사용하여 사전 정의된 옵션을 선택합니다.

## 구성요소에 멤버 및 함수 추가

공식 구성요소, 스크립트 구성요소, 조건 구성요소, 멤버 및 데이터 범위 구성요소에 멤버를 추가합니다.

### 참조:

- [구성요소에 멤버 및 함수 추가 정보](#)
- [하나 이상의 차원에 있는 하나의 멤버 또는 함수를 구성요소에 추가](#)
- [하나 이상의 차원에 있는 여러 멤버 또는 함수를 구성요소에 추가](#)
- [구성요소에서 멤버 및 함수 제거](#)
- [멤버 선택기 내에서 멤버 검색](#)
- [멤버 선택기에서 이름, 별칭 또는 등록정보로 멤버 검색](#)

## 구성요소에 멤버 및 함수 추가 정보

공식 구성요소, 스크립트 구성요소, 조건 구성요소, 멤버 및 데이터 범위 구성요소에 멤버를 추가할 수 있습니다. 또한 멤버 목록을 반환하는 함수를 공식, 스크립트 및 조건 구성요소에 추가할 수도 있습니다. 구성요소가 속해 있는 애플리케이션의 차원에서 멤버 및 함수를 선택할 수 있습니다.

작업 중인 구성요소에 따라, 한 차원에서 하나 이상의 멤버나 함수를 선택하거나, 여러 차원에서 하나 이상의 멤버나 차원을 선택할 수 있습니다.

- 다음은 여러 차원에 대해 단일 멤버를 선택할 수 있는 컨텍스트입니다.
  - 공식 구성요소의 공식 그리드에서
  - 필수 매개변수가 단일 멤버인 함수에서
- 멤버 유형의 변수 값을 정의할 때 복수 차원에 대한 복수 멤버를 선택할 수 있습니다.
- 다음은 단일 차원에 대해 여러 멤버를 선택할 수 있는 컨텍스트입니다.
  - 글로벌 범위의 비즈니스 규칙에서
  - 디자인 타임 프롬프트 유형이 데이터 교차인 템플릿에서
  - 유형이 멤버인 변수 값을 정의할 때
  - 유형이 멤버인 변수 제한을 정의할 때
  - 루프 구성요소에서
  - 매개변수가 멤버인 함수에서
- 다음은 단일 차원에 대해 단일 멤버를 선택할 수 있는 컨텍스트입니다.
  - 유형이 멤버인 변수 값을 정의할 때
  - 매개변수가 멤버인 함수에서

## 하나 이상의 차원에 있는 하나의 멤버 또는 함수를 구성요소에 추가

멤버 선택기를 사용하여 차원 내의 멤버와 함수를 선택할 수 있습니다. 차원 내의 멤버를 확장하고 축소할 때는 [+] 및 [-]를 사용합니다.

[멤버 선택기] 대화상자에는 세 가지 탭이 있습니다. [멤버] 탭과 [함수] 탭에는 선택한 차원의 모든 멤버와 함수가 나타나며, 세 번째 탭인 [검색] 탭은 멤버 또는 멤버 설명을 검색하는 데 사용됩니다. 선택한 멤버와 함수는 오른쪽에 있는 [선택 항목] 아래에 나열됩니다.

하나 이상의 차원에 있는 멤버를 구성요소에 추가하려면 다음을 수행합니다.

1. **시스템 뷰**에서 템플릿 또는 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 비즈니스 규칙이나 템플릿이 열리면 해당 순서도에서 멤버를 추가하려는 구성요소를 선택합니다.

### 주:

고정 루프 구성요소에는 멤버를 추가할 수 없습니다.

3. **멤버 선택기** 아이콘을 누릅니다.
4. **멤버 선택기**의 **차원(둘 이상)**에서 차원을 선택합니다.  
선택한 차원의 멤버는 **멤버**에 표시됩니다.
5. 다음 태스크 중 하나를 수행합니다.

- **멤버(둘 이상)**에서 멤버를 선택하고 **오른쪽 화살표** 버튼을 눌러 **선택 항목** 목록으로 이동합니다. 이 표의 옵션을 사용하여 선택 항목을 보다 구체적으로 정의할 수도 있습니다.

**표 8-1** 멤버 선택기 버튼

버튼	설명
특별 추가	<p><b>(Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 애플리케이션 사용자만 해당)</b> 다음 중 하나를 선택하여 탭에서 선택한 멤버 또는 함수와 관련된 멤버나 함수를 추가합니다.</p> <ul style="list-style-type: none"> <li>- 멤버</li> <li>- 1차 하위 멤버</li> <li>- 1차 하위 멤버(포함)</li> <li>- 하위 멤버</li> <li>- 하위 멤버(포함)</li> <li>- 동위 멤버</li> <li>- 동위 멤버(포함)</li> <li>- 1차 상위 멤버</li> <li>- 1차 상위 멤버(포함)</li> <li>- 상위 멤버</li> <li>- 상위 멤버(포함)</li> <li>- 상대</li> <li>- 레벨 0(기준)</li> <li>- 포함</li> </ul> <p><b>(Essbase 합산 저장영역 애플리케이션 사용자만 해당)</b> 탭에서 선택한 멤버나 함수와 연관된 다른 멤버나 함수를 추가하려면 다음 중 하나를 선택합니다.</p> <ul style="list-style-type: none"> <li>- 동위 멤버</li> <li>- 레벨 0 하위 멤버</li> </ul>

 **주:**

Planning의 [특별 추가] 선택에는 [레벨 0(기준)] 또는 [포함]이 없습니다.

선택	선택 목록으로 멤버 또는 함수를 이동하려면 선택합니다.
선택 취소	선택 목록에서 멤버 또는 함수를 제거하려면 선택합니다.
모두 선택 취소	선택 목록에서 모든 멤버 또는 함수를 제거하려면 선택합니다.

- **함수**에서 다음 태스크를 수행합니다.

**Planning 및 Essbase 블록 저장영역 애플리케이션 사용자만 해당:**

- a. 함수를 선택합니다.
- b. 아래 표에 따라 함수의 필수 값을 입력합니다.

표 8-2 함수 및 값

함수	매개변수	설명
@ALLANCESTORS	멤버 이름 - 멤버 이름을 입력하거나 [멤버]를 눌러 멤버를 선택합니다.	공유 멤버로 나타나는 지정된 멤버의 상위 멤버를 비롯하여 지정된 멤버의 모든 상위 멤버를 포함하도록 선택 항목을 확장합니다.
@ANCEST	<ul style="list-style-type: none"> <li>- 차원 이름 - 차원에서 선택한 차원 이름입니다.</li> <li>- 세대 레벨 번호 - 상위 멤버 값이 반환되는 세대 또는 레벨 번호를 정의하는 정수 값입니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> <li>- 멤버 이름 - 임의 멤버 이름 또는 멤버 조합입니다.</li> </ul>	지정된 차원에서 계산되는 현재 멤버의 지정된 세대 또는 레벨에 있는 상위 멤버를 반환합니다. 선택적 멤버 이름을 지정하는 경우 해당 상위 멤버가 지정된 멤버와 결합됩니다.
@ANCESTORS	<ul style="list-style-type: none"> <li>- 멤버 이름 - 멤버 이름 또는 멤버 조합입니다.</li> <li>- 세대 레벨 번호 - 멤버를 포함할 절대 세대 또는 레벨 번호를 정의하는 정수 값입니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> <li>- 세대 레벨 이름 - 선택 항목에 멤버를 포함할 최대 레벨 이름 또는 세대 이름입니다.</li> </ul>	지정된 멤버의 모든 상위 멤버를 포함하도록 선택 항목을 확장합니다.
@ATTRIBUTE	속성 멤버 이름 - 입력한 차원에 대해 선택 항목에 포함할 속성 멤버 이름 또는 멤버 조합을 입력합니다.	지정된 속성 멤버(mbrName)와 연계된 모든 기존 멤버 목록을 생성합니다.
@CHILDREN	멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.	지정된 멤버의 모든 1차 하위를 포함하도록 확장합니다.
@CURRMBR	차원 이름 - 차원 이름입니다.	지정된 차원에서 현재 계산되는 멤버를 반환합니다.
@DESCENDANTS	<ul style="list-style-type: none"> <li>- 멤버 이름 - 멤버 이름 또는 멤버 조합입니다.</li> <li>- 세대 레벨 번호 - 멤버를 포함할 절대 세대 또는 레벨 번호를 정의하는 정수 값입니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> <li>- 세대 레벨 이름 - 선택 항목에 멤버를 포함할 최대 레벨 이름 또는 세대 이름입니다.</li> </ul>	지정된 멤버의 모든 하위를 포함하도록 확장합니다.
@GENMBRS	<ul style="list-style-type: none"> <li>- 차원 이름 - 차원 이름입니다.</li> <li>- genName - dimName의 세대 이름입니다. 양의 정수는 세대 번호를 정의합니다.</li> </ul>	지정된 멤버의 모든 하위를 포함하도록 확장합니다.
@IALLANCESTORS	멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.	공유 멤버로 나타나는 지정된 멤버의 상위 멤버를 비롯하여 지정된 멤버 및 모든 상위 멤버를 포함하도록 선택 항목을 확장합니다.



표 8-2 (계속) 함수 및 값

함수	매개변수	설명
@IANCESTORS	<ul style="list-style-type: none"> <li>멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.</li> <li>세대 레벨 번호 - 선택 항목에 멤버를 포함할 최대 절대 세대 또는 레벨 번호를 정의하는 정수 값입니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> </ul>	지정된 멤버와 지정된 멤버의 모든 상위 멤버 또는 지정된 세대나 레벨까지의 모든 상위 멤버를 포함하도록 선택 항목을 확장합니다.
@ICHILDREN	멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.	지정된 멤버 및 바로 아래 레벨의 모든 멤버를 포함하도록 확장합니다.
@IDESCENDANTS	<ul style="list-style-type: none"> <li>멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.</li> <li>세대 레벨 번호 - 선택 항목에 멤버를 포함할 최대 절대 세대 또는 레벨 번호를 정의하는 정수 값입니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> </ul>	지정된 멤버와 지정된 멤버의 모든 하위 또는 지정된 세대나 레벨까지의 하위를 포함하도록 확장합니다.
@ILSIBLINGS	멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.	지정된 멤버 및 해당 멤버의 왼쪽 동위 멤버를 모두 포함하도록 확장합니다.
@IRDESCENDANTS	<ul style="list-style-type: none"> <li>멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.</li> <li>세대 레벨 번호 - 선택 항목에 멤버를 포함할 최대 절대 세대 또는 레벨 번호를 정의하는 정수 값입니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> </ul>	지정된 멤버와 지정된 멤버의 모든 하위 또는 선택적으로 지정된 세대나 레벨까지의 모든 하위를 반환합니다.
@IRSIBLINGS	멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.	멤버 및 지정된 멤버의 오른쪽 동위 멤버를 모두 포함하도록 확장합니다.
@ISIBLINGS	멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.	멤버의 모든 동위 멤버를 포함하도록 확장합니다.
@LEVELODESCENDANT	멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.	지정된 멤버의 모든 레벨 0 하위를 포함하도록 확장합니다.
@LEVMBRS	<ul style="list-style-type: none"> <li>차원 이름 - 차원 이름입니다.</li> <li>레벨 이름 - 레벨 이름 또는 레벨 번호를 정의하는 정수 값입니다. 정수 값은 0 또는 양의 정수여야 합니다.</li> </ul>	지정된 차원에서 지정된 레벨 번호 또는 레벨 이름을 가진 모든 멤버를 포함하도록 선택 항목을 확장합니다.
@LIST	인수 - 상위 함수에서 처리할 수 있도록 수집된 후 하나의 인수로 취급되는 인수 목록입니다. 인수는 멤버 이름, 멤버 조합, 멤버 세트 함수, 범위 함수 및 숫자 표현식일 수 있습니다.	목록 인수가 필요한 함수에서 처리되는 목록을 생성 및 구분합니다.
@LSIBLINGS	멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.	지정된 멤버의 왼쪽 동위 멤버를 모두 포함하도록 확장합니다.

표 8-2 (계속) 함수 및 값

함수	매개변수	설명
@MATCH	<ul style="list-style-type: none"> <li>멤버, 세대 - 선택 항목의 기준으로 사용할 멤버의 기본 이름 또는 사용자정의 이름입니다. 시스템에서 지정된 멤버 및 해당 하위 멤버의 멤버 이름 및 별칭 이름을 검색합니다.</li> <li>genName - 선택 항목의 기준으로 사용할 세대의 기본 이름 또는 사용자정의 이름입니다. 시스템에서 세대의 모든 멤버 이름 및 멤버 별칭 이름을 검색합니다.</li> <li>일치시킬 패턴 - 와일드카드 문자(또는 ?)를 포함하여 검색할 문자 패턴입니다. ?는 임의의 문자 하나를 대체합니다. ?는 패턴의 어디에서나 사용할 수 있습니다. *는 임의의 문자 수를 대신합니다. *는 패턴 끝에만 사용할 수 있습니다. 문자 패턴에 공백을 포함하려면 패턴을 큰따옴표("")로 묶습니다.</li> </ul>	후행 와일드카드 멤버 선택을 수행합니다.
@MEMBER	동위 - 문자열(큰따옴표로 묶음) 또는 문자열을 반환하는 함수입니다.	문자열로 제공되는 이름이 있는 멤버를 반환합니다
@MERGE	<ul style="list-style-type: none"> <li>목록 1 - 병합할 멤버의 첫번째 목록입니다.</li> <li>목록 2 - 병합할 멤버의 두번째 목록입니다.</li> </ul>	다른 함수에서 처리되는 두 멤버 목록을 병합합니다.
@PARENT	<ul style="list-style-type: none"> <li>차원 이름 - 차원 이름입니다.</li> <li>멤버 이름 - 반환된 상위와 조합될 멤버 이름이나 멤버 조합 또는 멤버나 멤버 조합을 반환하는 함수입니다.</li> </ul>	지정된 차원에서 계산되는 현재 멤버의 상위를 반환합니다.
@RANGE	<ul style="list-style-type: none"> <li>멤버 이름 - 반환된 상위와 조합될 멤버 이름이나 멤버 조합 또는 멤버나 멤버 조합을 반환하는 함수입니다.</li> <li>범위 목록 - 멤버 이름, 십표로 구분된 멤버 이름 목록, 멤버 세트 함수 또는 범위 함수입니다. rangeList가 지정되지 않은 경우 시스템에서 Time으로 태그가 지정된 차원의 레벨 0 멤버를 사용합니다.</li> </ul>	한 차원의 지정된 멤버(mbrName)와 다른 차원의 지정된 멤버 범위(rangeList)를 교차하는 멤버 목록을 반환합니다.
@RDESCENDANTS	<ul style="list-style-type: none"> <li>멤버 이름 - 반환된 상위와 조합될 멤버 이름이나 멤버 조합 또는 멤버나 멤버 조합을 반환하는 함수입니다.</li> <li>세대 레벨 번호 - 멤버를 선택할 최소 절대 세대 또는 레벨 번호를 정의하는 정수 값입니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> </ul>	지정된 멤버의 모든 하위 또는 지정된 세대나 레벨까지의 하위를 반환합니다.
@RELATIVE	<ul style="list-style-type: none"> <li>멤버 이름 - 반환된 상위와 조합될 멤버 이름이나 멤버 조합 또는 멤버나 멤버 조합을 반환하는 함수입니다.</li> <li>세대 레벨 번호 - 멤버를 선택할 최소 절대 세대 또는 레벨 번호를 정의하는 정수 값입니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> </ul>	지정된 멤버 위 또는 아래에 있는 지정된 세대 또는 레벨의 모든 멤버를 선택할 수 있습니다.

표 8-2 (계속) 함수 및 값

함수	매개변수	설명
@REMOVE	<ul style="list-style-type: none"> <li>목록 1 - 병합할 멤버의 첫번째 목록입니다.</li> <li>목록 2 - 병합할 멤버의 두번째 목록입니다.</li> </ul>	메시지를 디버그, 정보, 경고 또는 오류로 반환합니다.
@RSIBLINGS	멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.	멤버의 오른쪽 동위 멤버를 모두 포함하도록 확장합니다.
@SHARE	범위 목록 - 멤버, 멤버를 반환하는 함수 또는 멤버 범위의 심표로 구분된 목록입니다. rangeList의 모든 멤버는 같은 차원에서 가져와야 합니다.	rangeList의 각 멤버를 검사하여 공유 멤버가 포함되어 있는지 확인하고 해당 공유 멤버 목록을 반환합니다.
@SIBLING	멤버 이름 - 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수입니다.	지정된 멤버의 모든 동위를 포함하도록 확장합니다.
@UDA	<ul style="list-style-type: none"> <li>차원 이름 - 사용자 정의 속성과 연계된 차원의 이름입니다.</li> <li>사용자 정의 속성 - 데이터베이스 아웃라인에 표시된 사용자 정의 속성의 이름입니다.</li> </ul>	Essbase 서버에서 사용자 정의 속성(UDA)으로 정의한 공통 속성을 기준으로 멤버를 선택합니다.
@WITHATTR	<ul style="list-style-type: none"> <li>차원 이름 - 속성 차원의 이름입니다.</li> <li>연산자 - 큰따옴표("")로 묶은 연산자 지정입니다.</li> <li>값 - 연산자와 조합되어 충족해야 하는 조건을 정의하는 값입니다. 값은 속성 멤버, 상수 또는 날짜 형식의 함수(즉, @TODATE)일 수 있습니다.</li> </ul>	지정한 조건을 충족하는 속성과 연계된 모든 기준 멤버를 반환합니다.
@XRANGE	<ul style="list-style-type: none"> <li>멤버 이름 1 - 멤버 이름, 멤버 조합 또는 단일 멤버를 반환하는 함수입니다.</li> <li>멤버 이름 2 - 멤버 이름, 멤버 조합 또는 단일 멤버를 반환하는 함수입니다. <i>mbrName1</i>이 횡차원 멤버(예: Actual-&gt;Jan)인 경우 <i>mbrName2</i>도 횡차원 멤버여야 하며 차원 순서가 <i>mbrName1</i>에 사용된 순서와 일치해야 합니다.</li> </ul>	동일한 레벨에서 지정된 두 단일 또는 횡차원 멤버(포함) 사이의 멤버 범위를 반환합니다.

Oracle Hyperion Financial Management, Planning 및 Essbase 블록 저장영역 애플리케이션에 지원되는 함수 유형 목록은 [함수 작업](#)을 참조하십시오.

- c. **오른쪽 화살표** 버튼을 눌러 함수를 **선택 항목** 목록으로 이동합니다. 표 1의 옵션을 사용하여 선택 항목을 보다 구체적으로 정의할 수도 있습니다.
- **검색**에서 다음 태스크를 수행하여 멤버를 검색합니다.
  - a. **차원**에서 멤버를 검색할 차원을 선택합니다.
  - b. **찾기**에서 검색할 멤버 유형, 멤버 이름 또는 설명을 선택합니다.
  - c. 검색할 멤버 이름 또는 설명을 입력하거나, 차원에 있는 모든 멤버를 표시하려면 기본 와일드카드(\*)를 그대로 사용합니다.
  - d. 필드에 입력한 멤버를 검색하려면 **검색**을 선택합니다. **멤버 선택기** 내에서 **멤버 검색**을 참조하십시오.
  - e. 고급 검색 옵션에 액세스하려면 **고급 검색**을 선택합니다. **멤버 선택기** 내에서 **멤버 검색**을 참조하십시오.

- f. 멤버를 선택하고 **오른쪽 화살표**를 눌러 멤버를 **선택 항목** 목록으로 이동합니다. **표 1**의 옵션을 사용하여 선택 항목을 보다 구체적으로 정의할 수도 있습니다.
- **Financial Management 사용자만 해당:** 목록에서 다음 태스크를 수행하십시오.
  - a. 선택한 차원에 대한 멤버 목록을 선택하고 이 표에 따라 목록의 매개변수를 입력합니다.

**표 8-3 Financial Management 멤버 목록 및 매개변수**

차원	멤버 목록 및 매개변수
계정	<ul style="list-style-type: none"> <li>- 계층, 하위 멤버, 상위 멤버, 상위, 1차 하위 및 기본에 대해 다음 매개변수의 값을 입력합니다.                             <ul style="list-style-type: none"> <li>* 최상위 멤버</li> <li>* 활성 엔티티</li> <li>* 시나리오</li> <li>* 연도</li> <li>* 기간</li> </ul> </li> <li>- 시스템에 대해 매개변수를 입력하지 않습니다.</li> </ul>
엔티티	<ul style="list-style-type: none"> <li>- 계층, 하위 멤버, 상위 멤버, 상위, 1차 하위 및 기본에 대해 다음 매개변수의 값을 입력합니다.                             <ul style="list-style-type: none"> <li>* 최상위 멤버</li> <li>* 활성 엔티티</li> <li>* 시나리오</li> <li>* 연도</li> <li>* 기간</li> </ul> </li> <li>- 조정 엔티티 및 상위 조정 엔티티에 대해 값을 입력하지 않습니다.</li> </ul>
ICP	<ul style="list-style-type: none"> <li>- 계층, 하위 멤버, 상위 멤버, 상위, 1차 하위 및 기본에 대해 다음 매개변수의 값을 입력합니다.                             <ul style="list-style-type: none"> <li>* 최상위 멤버</li> <li>* 활성 엔티티</li> <li>* 시나리오</li> <li>* 연도</li> <li>* 기간</li> </ul> </li> <li>- 시스템에 대해 값을 입력하지 않습니다.</li> </ul>
기간	<ul style="list-style-type: none"> <li>- 계층, 하위 멤버, 상위 멤버, 상위 및 1차 하위에 대해 다음 매개변수의 값을 입력합니다.                             <ul style="list-style-type: none"> <li>* 최상위 멤버</li> <li>* 활성 엔티티</li> <li>* 시나리오</li> <li>* 연도</li> <li>* 기간</li> </ul> </li> <li>- 첫번째 세대, 두번째 세대, 세번째 세대, 네번째 세대, 다섯번째 세대 및 여섯번째 세대에 대해 값을 입력하지 않습니다.</li> </ul>

표 8-3 (계속) Financial Management 멤버 목록 및 매개변수

차원	멤버 목록 및 매개변수
시나리오	<ul style="list-style-type: none"> <li>- 계층, 하위 멤버, 상위 멤버 및 상위에 대해 다음 매개변수의 값을 입력합니다. <ul style="list-style-type: none"> <li>* 최상위 멤버</li> <li>* 활성 엔티티</li> <li>* 시나리오</li> <li>* 연도</li> <li>* 기간</li> </ul> </li> <li>- 프로세스 관리 지원에 대해 값을 입력하지 않습니다.</li> </ul>
값	<ul style="list-style-type: none"> <li>- 계층 및 하위 멤버에 대해 다음 매개변수의 값을 입력합니다. <ul style="list-style-type: none"> <li>* 최상위 멤버</li> <li>* 활성 엔티티</li> <li>* 시나리오</li> <li>* 연도</li> <li>* 기간</li> </ul> </li> <li>- 입력, 조정, 합계 및 기본 통화에 대해 값을 입력하지 않습니다.</li> </ul>
뷰	<ul style="list-style-type: none"> <li>- 계층 및 하위 멤버에 대해 다음 매개변수의 값을 입력합니다. <ul style="list-style-type: none"> <li>- 최상위 멤버</li> <li>- 활성 엔티티</li> <li>- 시나리오</li> <li>- 연도</li> <li>- 기간</li> </ul> </li> </ul>
연도	<ul style="list-style-type: none"> <li>- 계층 및 하위 멤버에 대해 다음 매개변수의 값을 입력합니다. <ul style="list-style-type: none"> <li>- 최상위 멤버</li> <li>- 활성 엔티티</li> <li>- 시나리오</li> <li>- 연도</li> <li>- 기간</li> </ul> </li> </ul>

b. 오른쪽 화살표를 눌러 멤버 목록을 선택 항목 목록으로 이동합니다. 표 1의 옵션을 사용하여 선택 항목을 보다 구체적으로 정의할 수도 있습니다.

6. 다음 태스크 중 하나를 수행합니다.

- 다음에 대해 멤버, 함수 또는 멤버 목록을 선택하는 경우 7단계를 계속 진행합니다.
  - 유형이 멤버인 변수 값
  - 매개변수가 멤버인 함수
- 다음에 대해 멤버, 함수 또는 멤버 목록을 선택하는 경우 각 차원에 대한 멤버, 함수 또는 멤버 목록을 선택할 때까지 이전 두 단계를 반복한 후 다음 단계를 계속 진행합니다.
  - Planning 또는 Essbase 비즈니스 규칙의 글로벌 범위
  - 디자인 타임 프롬프트 유형이 멤버 교차인 템플릿
  - 유형이 멤버인 변수 값
  - 유형이 멤버인 변수 제한
  - 루프 구성요소
  - 매개변수가 멤버인 함수

## 7. 확인을 누릅니다.

## 하나 이상의 차원에 있는 여러 멤버 또는 함수를 구성요소에 추가

멤버 선택기를 사용하여 차원 내의 멤버를 선택할 수 있습니다. 차원 내의 멤버를 확장하고 축소할 때는 [+] 및 [-]를 사용합니다.

[멤버 선택기] 대화상자에는 세 가지 탭이 있습니다. [멤버] 탭과 [함수] 탭에는 선택한 차원의 모든 멤버와 함수가 나타나며, 세 번째 탭인 [검색] 탭은 멤버 또는 멤버 설명을 검색하는 데 사용됩니다. 선택한 멤버와 함수는 오른쪽에 있는 [선택 항목] 아래에 나열됩니다.

여러 멤버를 선택할 수 있는 구성요소에 있을 때는 Shift + 누르기 및 Ctrl + 누르기를 통해 인접하거나 인접하지 않은 여러 멤버를 선택할 수 있습니다.

하나 이상의 차원에 있는 여러 멤버를 구성요소에 추가하려면 다음을 수행합니다.

1. 시스템 뷰에서 템플릿 또는 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 열기를 선택합니다.
2. 비즈니스 규칙이나 템플릿이 열리면 해당 순서도에서 멤버를 추가하려는 구성요소를 선택합니다.

 주:

고정 루프 구성요소에는 멤버를 추가할 수 없습니다.

3. 멤버 선택기 아이콘을 누릅니다.
4. 멤버 선택기의 차원(둘 이상)에서 차원을 선택합니다.  
선택한 차원의 멤버는 멤버에 표시됩니다.
5. 다음 태스크 중 하나를 수행하여 멤버를 검색하고 선택합니다.
  - 멤버(둘 이상)에서 Ctrl + 누르기 또는 Shift + 누르기를 사용하여 멤버를 선택하고 오른쪽 화살표를 눌러 멤버를 선택 항목 목록으로 이동합니다. 이 표의 옵션을 사용하여 선택 항목을 보다 구체적으로 정의할 수도 있습니다.

표 8-4 멤버 선택기 버튼

버튼	설명
특별 추가	<p><b>(Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 애플리케이션 사용자만 해당)</b> 탭에서 선택한 멤버나 함수와 연관된 다른 멤버나 함수를 추가하려면 다음 중 하나를 선택합니다.</p> <ul style="list-style-type: none"> <li>- 멤버</li> <li>- 1차 하위 멤버</li> <li>- 1차 하위 멤버(포함)</li> <li>- 하위 멤버</li> <li>- 하위 멤버(포함)</li> <li>- 동위 멤버</li> <li>- 동위 멤버(포함)</li> <li>- 1차 상위 멤버</li> <li>- 1차 상위 멤버(포함)</li> <li>- 상위 멤버</li> <li>- 상위 멤버(포함)</li> <li>- 상대</li> <li>- 레벨 0(기준)</li> <li>- 포함</li> </ul> <p><b>(Essbase 합산 저장영역 애플리케이션 사용자만 해당)</b> 탭에서 선택한 멤버나 함수와 연관된 다른 멤버나 함수를 추가하려면 다음 중 하나를 선택합니다.</p> <ul style="list-style-type: none"> <li>- 동위 멤버</li> <li>- 레벨 0 하위 멤버</li> </ul> <p><b>참고:</b> Planning의 [특별 추가] 선택에는 [레벨 0(기준)] 또는 [포함]이 없습니다.</p>
선택	선택 목록으로 멤버 또는 함수를 이동하려면 선택합니다.
선택 취소	선택 목록에서 멤버 또는 함수를 제거하려면 선택합니다.
모두 선택 취소	선택 목록에서 모든 멤버 또는 함수를 제거하려면 선택합니다.

- 함수에서 다음 태스크를 수행합니다.

**Planning 및 Essbase 블록 저장영역 애플리케이션 사용자만 해당:**

- a. **Ctrl + 누르기** 또는 **Shift + 누르기**를 사용하여 함수를 선택합니다.
- b. 아래 표에 따라 함수의 필수 값을 입력합니다.

표 8-5 함수 및 값

함수	입력할 값	설명
@ALLANCESTORS	멤버 이름	멤버 이름을 입력하거나 [멤버]를 눌러 멤버를 선택합니다.
@ANCEST	<ul style="list-style-type: none"> <li>- 차원 이름</li> <li>- 세대 레벨 번호</li> <li>- 멤버 이름</li> </ul>	<ol style="list-style-type: none"> <li>i. 차원에서 선택한 차원 이름을 입력합니다.</li> <li>ii. 상위 멤버 값이 반환되는 세대 또는 레벨 번호를 정의하는 정수 값을 입력합니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> <li>iii. 멤버 이름 또는 멤버 조합을 입력합니다.</li> </ol>

표 8-5 (계속) 함수 및 값

함수	입력할 값	설명
@ANCESTORS	<ul style="list-style-type: none"> <li>- 멤버 이름</li> <li>- 세대 레벨 번호</li> <li>- 세대 레벨 이름</li> </ul>	<ul style="list-style-type: none"> <li>i. 멤버 이름 또는 멤버 조합을 입력합니다.</li> <li>ii. 멤버를 포함할 절대 세대 또는 레벨 번호를 정의하는 정수 값을 입력합니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> <li>iii. 선택 영역에서 멤버를 포함할 레벨 이름 또는 세대 이름(해당 항목의 상위)을 입력합니다.</li> </ul>
@ATTRIBUTE	속성 멤버 이름	입력한 차원에 대해 선택 항목에 포함할 속성 멤버 이름 또는 멤버 조합을 입력합니다.
@CHILDREN	멤버 이름	멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수를 입력합니다.
@CURRMBR	차원 이름	차원 이름을 입력합니다.
@DESCENDANTS	<ul style="list-style-type: none"> <li>- 멤버 이름</li> <li>- 세대 레벨 번호</li> <li>- 세대 레벨 이름</li> </ul>	<ul style="list-style-type: none"> <li>i. 멤버 이름 또는 멤버 조합을 입력합니다.</li> <li>ii. 멤버를 포함할 절대 세대 또는 레벨 번호를 정의하는 정수 값을 입력합니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> <li>iii. 선택 영역에서 멤버를 포함할 레벨 이름 또는 세대 이름(해당 항목의 상위)을 입력합니다.</li> </ul>
@GENMBRS	<ul style="list-style-type: none"> <li>- 차원 이름</li> <li>- genName</li> </ul>	<ul style="list-style-type: none"> <li>i. 차원 이름을 입력합니다.</li> <li>ii. dimName의 세대 이름을 입력합니다. 양의 정수는 세대 번호를 정의합니다.</li> </ul>
@IALLANCESTORS	멤버 이름	멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수를 입력합니다.
@IANCESTORS	<ul style="list-style-type: none"> <li>- 멤버 이름</li> <li>- 세대 레벨 번호</li> </ul>	<ul style="list-style-type: none"> <li>i. 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수를 입력합니다.</li> <li>ii. 선택 영역에서 멤버를 포함할 절대 세대 또는 레벨 번호(해당 항목 위로 적용)를 정의하는 정수 값을 입력합니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> </ul>
@ICHILDREN	멤버 이름	멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수를 입력합니다.
@IDESCENDANTS	<ul style="list-style-type: none"> <li>- 멤버 이름</li> <li>- 세대 레벨 번호</li> </ul>	<ul style="list-style-type: none"> <li>i. 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수를 입력합니다.</li> <li>ii. 선택 영역에서 멤버를 포함할 절대 세대 또는 레벨 번호(해당 항목 위로 적용)를 정의하는 정수 값을 입력합니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> </ul>
@ILSIBLINGS	멤버 이름	멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수를 입력합니다.



표 8-5 (계속) 함수 및 값

함수	입력할 값	설명
@IRDESCENDANTS	<ul style="list-style-type: none"> <li>- 멤버 이름</li> <li>- 세대 레벨 번호</li> </ul>	<ul style="list-style-type: none"> <li>i. 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수를 입력합니다.</li> <li>ii. 선택 영역에서 멤버를 포함할 절대 세대 또는 레벨 번호(해당 항목 위로 적용)를 정의하는 정수 값을 입력합니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</li> </ul>
@IRSIBLINGS	<ul style="list-style-type: none"> <li>- 멤버 이름</li> </ul>	<ul style="list-style-type: none"> <li>- 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수를 입력합니다.</li> </ul>
@ISIBLINGS	<ul style="list-style-type: none"> <li>- 멤버 이름</li> </ul>	<ul style="list-style-type: none"> <li>- 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수를 입력합니다.</li> </ul>
@LEVMBRS	<ul style="list-style-type: none"> <li>- 차원 이름</li> <li>- 레벨 이름</li> </ul>	<ul style="list-style-type: none"> <li>i. 차원 이름을 입력합니다.</li> <li>ii. 레벨 이름 또는 레벨 번호를 정의하는 정수 값을 입력합니다. 정수 값은 0 또는 양의 정수여야 합니다.</li> </ul>
@LIST	<ul style="list-style-type: none"> <li>- 인수</li> </ul>	<ul style="list-style-type: none"> <li>- 상위 함수로 처리될 수 있도록 수집된 후 하나의 인수로 취급되는 인수 목록을 입력합니다. 인수는 멤버 이름, 멤버 조합, 멤버 세트 함수, 범위 함수 및 숫자 표현식일 수 있습니다.</li> </ul>
@LSIBLINGS	<ul style="list-style-type: none"> <li>- 멤버 이름</li> </ul>	<ul style="list-style-type: none"> <li>- 멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수를 입력합니다.</li> </ul>
@MATCH	<ul style="list-style-type: none"> <li>- 멤버, 세대</li> <li>- genName</li> <li>- 일치시킬 패턴</li> </ul>	<ul style="list-style-type: none"> <li>i. 선택 항목의 기준으로 사용할 멤버의 기본 또는 사용자정의 이름을 입력하십시오. 시스템에서 지정된 멤버 및 해당 하위 멤버의 멤버 이름 및 별칭 이름을 검색합니다.</li> <li>ii. 선택 항목의 기준으로 사용할 세대의 기본 또는 사용자정의 이름을 입력하십시오. 시스템에서 세대의 모든 멤버 이름 및 멤버 별칭 이름을 검색합니다.</li> <li>iii. 와일드카드(* 또는 ?)를 포함하여 검색할 문자 패턴을 입력합니다. ?는 모든 문자의 각 경우를 대신합니다. ?는 패턴의 어디에서나 사용할 수 있습니다. *는 임의의 문자 수를 대신합니다. *는 패턴 끝에만 사용할 수 있습니다. 문자 패턴에 공백을 포함하려면 패턴을 큰따옴표(" ")로 묶습니다.</li> </ul>
@MEMBER	<ul style="list-style-type: none"> <li>- 문자열</li> </ul>	<ul style="list-style-type: none"> <li>- 문자열(큰따옴표로 묶음) 또는 문자열을 반환하는 함수를 입력합니다.</li> </ul>
@MERGE	<ul style="list-style-type: none"> <li>- 목록 1</li> <li>- 목록 2</li> </ul>	<ul style="list-style-type: none"> <li>i. 병합할 멤버의 첫 번째 목록을 입력합니다.</li> <li>ii. 병합할 멤버의 두 번째 목록을 입력합니다.</li> </ul>
@PARENT	<ul style="list-style-type: none"> <li>- 차원 이름</li> <li>- 멤버 이름</li> </ul>	<ul style="list-style-type: none"> <li>i. 차원 이름을 입력합니다.</li> <li>ii. 반환된 상위 멤버와 조합될 멤버 이름이나 멤버 조합 또는 멤버나 멤버 조합을 반환하는 함수를 입력합니다.</li> </ul>
@RANGE	<ul style="list-style-type: none"> <li>- 멤버 이름</li> <li>- 범위 목록</li> </ul>	<ul style="list-style-type: none"> <li>i. 반환된 상위 멤버와 조합될 멤버 이름이나 멤버 조합 또는 멤버나 멤버 조합을 반환하는 함수를 입력합니다.</li> <li>ii. 멤버 이름, 심표로 구분된 멤버 이름 목록, 멤버 세트 함수 또는 범위 함수를 입력하십시오. rangeList가 지정되지 않은 경우 시스템에서 시간으로 태그가 지정된 차원의 레벨 0 멤버를 사용합니다.</li> </ul>

표 8-5 (계속) 함수 및 값

함수	입력할 값	설명
@RDESCENDANTS-	- 멤버 이름 - 세대 레벨 번호	<p>i. 반환된 상위 멤버와 조합될 멤버 이름이나 멤버 조합 또는 멤버나 멤버 조합을 반환하는 함수를 입력합니다.</p> <p>ii. 멤버를 선택할 절대 세대 또는 레벨 번호(해당 항목 아래로 적용)를 정의하는 정수 값을 입력합니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</p>
@RELATIVE	- 멤버 이름 - 세대 레벨 번호	<p>i. 반환된 상위 멤버와 조합될 멤버 이름이나 멤버 조합 또는 멤버나 멤버 조합을 반환하는 함수를 입력합니다.</p> <p>ii. 멤버를 선택할 절대 세대 또는 레벨 번호(해당 항목 아래로 적용)를 정의하는 정수 값을 입력합니다. 양의 정수는 세대 번호를 정의합니다. 0 값이나 음의 정수는 레벨 번호를 정의합니다.</p>
@REMOVE	- 목록 1 - 목록 2	<p>i. 병합할 멤버의 첫 번째 목록을 입력합니다.</p> <p>ii. 병합할 멤버의 두 번째 목록을 입력합니다.</p>
@RSIBLINGS	멤버 이름	멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수를 입력합니다.
@SHARE	범위 목록	멤버, 멤버를 반환하는 함수 또는 멤버의 범위를 심표로 구분된 목록으로 입력합니다. rangeList의 모든 멤버는 같은 차원에서 가져와야 합니다.
@SIBLINGS	멤버 이름	멤버 이름 또는 멤버 조합이나 멤버 또는 멤버 조합을 반환하는 함수를 입력합니다.
@UDA	- 차원 이름 - 사용자정의 속성	<p>i. 사용자정의 속성과 관련된 차원의 이름을 입력합니다.</p> <p>ii. 데이터베이스 아웃라인에 표시된 대로 사용자정의 속성의 이름을 입력합니다.</p>
@WITHATTR	- 차원 이름 - 연산자 - 값	<p>i. 속성 차원의 이름을 입력합니다.</p> <p>ii. 큰따옴표("")로 묶어 연산자를 입력합니다.</p> <p>iii. 연산자와 함께 사용되어 충족해야 하는 조건을 정의하는 값을 입력합니다. 값은 속성 멤버, 상수 또는 날짜 형식의 함수(즉, @TODATE)일 수 있습니다.</p>
@XRANGE	- 멤버 이름 1 - 멤버 이름 2	<p>i. 멤버 이름, 멤버 조합 또는 단일 멤버를 반환하는 함수를 입력합니다.</p> <p>ii. 멤버 이름, 멤버 조합 또는 단일 멤버를 반환하는 함수를 입력합니다. mbrName1이 차원 간 멤버(예: Actual-&gt;Jan)인 경우 mbrName2도 마찬가지로 하여 차원 또는 순서가 mbrName1에 사용된 순서와 일치해야 합니다.</p>

- c. 오른쪽 화살표를 눌러 멤버를 선택 항목 목록으로 이동합니다. 표 1의 옵션을 사용하여 선택 항목을 보다 구체적으로 정의할 수도 있습니다.
- 검색에서 다음 태스크를 수행하여 멤버를 검색합니다.
  - a. 차원에서 멤버를 검색할 차원을 선택합니다.
  - b. 찾기에서 검색할 멤버 유형, 멤버 이름 또는 설명을 선택합니다.

- c. 검색할 멤버 이름 또는 설명을 입력하거나, 차원에 있는 모든 멤버를 표시하려면 기본 와일드카드(\*)를 그대로 사용합니다.
  - d. 필드에 입력한 멤버를 검색하려면 **검색**을 선택합니다. **멤버 선택기 내에서 멤버 검색**을 참조하십시오.
  - e. 고급 검색 옵션에 액세스하려면 **고급 검색**을 선택합니다. **멤버 선택기 내에서 멤버 검색**을 참조하십시오.
  - f. 멤버를 선택하고 **오른쪽 화살표**를 눌러 멤버를 **선택 항목** 목록으로 이동합니다. **표 1**의 옵션을 사용하여 선택 항목을 보다 구체적으로 정의할 수도 있습니다.
- **Oracle Hyperion Financial Management 사용자만 해당:** 목록에서 다음 태스크를 수행합니다.
    - a. 선택한 차원에 대한 멤버 목록을 선택하고 이 표에 따라 목록의 매개변수를 입력합니다.

**표 8-6 Financial Management 멤버 목록 및 매개변수**

차원	멤버 목록 및 매개변수
계정	<ul style="list-style-type: none"> <li>- 계층, 하위 멤버, 상위 멤버, 상위, 1차 하위 및 기본에 대해 다음 매개변수의 값을 입력합니다.                             <ul style="list-style-type: none"> <li>* 최상위 멤버</li> <li>* 활성 엔티티</li> <li>* 시나리오</li> <li>* 연도</li> <li>* 기간</li> </ul> </li> <li>- 시스템에 대해 매개변수를 입력하지 않습니다.</li> </ul>
엔티티	<ul style="list-style-type: none"> <li>- 계층, 하위 멤버, 상위 멤버, 상위, 1차 하위 및 기본에 대해 다음 매개변수의 값을 입력합니다.                             <ul style="list-style-type: none"> <li>* 최상위 멤버</li> <li>* 활성 엔티티</li> <li>* 시나리오</li> <li>* 연도</li> <li>* 기간</li> </ul> </li> <li>- 조정 엔티티 및 상위 조정 엔티티에 대해 값을 입력하지 않습니다.</li> </ul>
ICP	<ul style="list-style-type: none"> <li>- 계층, 하위 멤버, 상위 멤버, 상위, 1차 하위 및 기본에 대해 다음 매개변수의 값을 입력합니다.                             <ul style="list-style-type: none"> <li>* 최상위 멤버</li> <li>* 활성 엔티티</li> <li>* 시나리오</li> <li>* 연도</li> <li>* 기간</li> </ul> </li> <li>- 시스템에 대해 값을 입력하지 않습니다.</li> </ul>

표 8-6 (계속) Financial Management 멤버 목록 및 매개변수

차원	멤버 목록 및 매개변수
기간	<ul style="list-style-type: none"> <li>- 계층, 하위 멤버, 상위 멤버, 상위 및 1차 하위에 대해 다음 매개변수의 값을 입력합니다. <ul style="list-style-type: none"> <li>* 최상위 멤버</li> <li>* 활성 엔티티</li> <li>* 시나리오</li> <li>* 연도</li> <li>* 기간</li> </ul> </li> <li>- 첫번째 세대, 두번째 세대, 세번째 세대, 네번째 세대, 다섯번째 세대 및 여섯번째 세대에 대해 값을 입력하지 않습니다.</li> </ul>
시나리오	<ul style="list-style-type: none"> <li>- 계층, 하위 멤버, 상위 멤버 및 상위에 대해 다음 매개변수의 값을 입력합니다. <ul style="list-style-type: none"> <li>* 최상위 멤버</li> <li>* 활성 엔티티</li> <li>* 시나리오</li> <li>* 연도</li> <li>* 기간</li> </ul> </li> <li>- 프로세스 관리 지원에 대해 값을 입력하지 않습니다.</li> </ul>
값	<ul style="list-style-type: none"> <li>- 계층 및 하위 멤버에 대해 다음 매개변수의 값을 입력합니다. <ul style="list-style-type: none"> <li>* 최상위 멤버</li> <li>* 활성 엔티티</li> <li>* 시나리오</li> <li>* 연도</li> <li>* 기간</li> </ul> </li> <li>- 입력, 조정, 합계 및 기본 통화에 대해 값을 입력하지 않습니다.</li> </ul>
뷰	<ul style="list-style-type: none"> <li>- 계층 및 하위 멤버에 대해 다음 매개변수의 값을 입력합니다. <ul style="list-style-type: none"> <li>- 최상위 멤버</li> <li>- 활성 엔티티</li> <li>- 시나리오</li> <li>- 연도</li> <li>- 기간</li> </ul> </li> </ul>
연도	<ul style="list-style-type: none"> <li>- 계층 및 하위 멤버에 대해 다음 매개변수의 값을 입력합니다. <ul style="list-style-type: none"> <li>- 최상위 멤버</li> <li>- 활성 엔티티</li> <li>- 시나리오</li> <li>- 연도</li> <li>- 기간</li> </ul> </li> </ul>

b. 오른쪽 화살표 버튼을 눌러 멤버 목록을 선택 항목 목록으로 이동합니다. 표 1의 옵션을 사용하여 선택 항목을 보다 구체적으로 정의할 수도 있습니다.

6. 다음 태스크 중 하나를 수행합니다.

- 멤버, 함수 또는 멤버 목록을 선택하여 유형이 멤버인 변수 값을 정의하는 경우 각 차원에 대한 멤버, 함수 또는 멤버 목록을 선택할 때까지 이전 두 단계를 반복한 후 다음 단계를 계속 진행합니다.
- 이러한 항목에 대해 멤버, 함수 또는 멤버 목록을 선택하는 경우 다음 단계를 계속 진행합니다.

- 글로벌 범위의 비즈니스 규칙
- 디자인 타임 프롬프트 유형이 멤버 교차인 템플릿
- 유형이 멤버인 변수 값
- 유형이 멤버인 변수 제한
- 루프 구성요소
- 매개변수가 멤버인 함수


7. 확인을 누릅니다.

## 구성요소에서 멤버 및 함수 제거

공식, 스크립트, 조건, 멤버 및 데이터 범위 구성요소에서 멤버를 제거할 수 있습니다. 공식, 스크립트 및 조건 구성요소에서 함수를 제거할 수 있습니다.

구성요소에서 멤버 및 함수를 제거할 경우 데이터베이스에서는 삭제되지 않습니다. 공유 구성요소에서 멤버 및 함수를 제거하려면 먼저 구성요소를 [공유 안 됨]으로 지정해야 합니다.

구성요소에서 멤버 또는 함수를 제거하려면 다음을 수행합니다.

1. **시스템 뷰**에서 템플릿 또는 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 비즈니스 규칙 또는 템플릿이 열리면 순서도에서 제거하려는 멤버나 함수가 들어 있는 구성요소를 선택합니다.
3. **작업, 멤버** 순으로 선택합니다.
4. **선택**에서 차원을 눌러 해당 차원의 모든 멤버를 제거하거나 멤버를 눌러 해당 멤버만 차원에서 제거합니다.
5. 왼쪽 화살표를 사용하여 **선택**에서 **멤버(둘 이상)** 또는 **함수**로 해당 멤버나 함수를 이동합니다. **표 1**을 참조하십시오.
6. 여러 차원에서 멤버를 제거하려면 다음을 누르고 3단계에서 5단계를 반복합니다.
7. **확인**을 누르고 를 누릅니다.

## 멤버 선택기 내에서 멤버 검색

멤버 선택기 내에서 멤버를 검색할 수 있습니다.

멤버를 검색하려면 다음을 수행합니다.

1. **시스템 뷰**에서 템플릿 또는 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 비즈니스 규칙이나 템플릿이 열리면 멤버를 검색하려는 구성요소를 선택합니다.
3. **작업**을 누르고 **멤버**를 선택합니다.
4. **멤버 선택기**의 **차원(둘 이상)**에서 검색하려는 멤버가 들어 있는 차원을 선택합니다. 차원, 별칭 및 개수가 **멤버** 탭에 표시됩니다.
5. 차원의 멤버를 표시하려면 다음을 선택합니다.

 주:

기본적으로 차원을 선택하면 아웃라인이 축소됩니다.

- **확장** - 차원의 한 레벨 아래 멤버를 표시합니다. **축소**를 선택하여 차원만 표시할 수 있습니다.
  - **모두 확장**을 선택하여 차원 아래의 모든 멤버를 표시합니다. **모두 축소**를 선택하여 차원만 표시할 수 있습니다.
6. **검색** 탭을 선택합니다.
    - a. **찾기**에서 검색할 멤버 유형, 멤버 이름 또는 설명을 선택합니다.
    - b. 검색할 멤버 이름 또는 설명을 입력하거나, 차원에 있는 모든 멤버를 표시하려면 기본 와일드카드(\*)를 그대로 사용합니다.
    - c. **검색**을 눌러 멤버나 설명을 검색합니다.  
멤버가 발견되면 **결과**에 표시됩니다. 아웃라인은 표시되지 않고 발견된 멤버만 표시됩니다.
    - d. **선택 사항**: **고급 검색**을 눌러 해당 이름, 별칭 또는 등록정보 중 하나로 멤버를 검색합니다. **멤버 선택기**에서 **이름**, **별칭** 또는 **등록정보**로 **멤버 검색**을 참조하십시오.
    - e. 멤버를 선택하고 **오른쪽 화살표**를 눌러 멤버를 **선택 항목** 목록으로 이동합니다. **표 1**의 옵션을 사용하여 선택 항목을 보다 구체적으로 정의할 수도 있습니다.
  7. **확인**을 누릅니다.

## 멤버 선택기에서 이름, 별칭 또는 등록정보로 멤버 검색

멤버 선택기 내의 고급 검색을 사용하여 해당 이름, 별칭 또는 등록정보 중 하나로 멤버를 검색할 수 있습니다.

이름, 별칭 또는 등록정보로 멤버를 검색하려면 다음을 수행합니다.

1. **멤버 선택기**에서 **검색** 탭을 선택하고 **고급 검색**을 누릅니다.
2. **멤버 찾기**의 **검색 기준**에서 다음 옵션 중 하나를 선택합니다.
  - **이름** - 이름으로 멤버를 검색하려면 선택합니다. 그런 다음 **4단계**로 이동합니다.
  - **별칭** - 별칭으로 멤버를 검색하려면 선택합니다. 그런 다음 **4단계**로 이동합니다.
  - **등록정보** - 등록정보 중 하나로 멤버를 검색하려면 선택합니다. 그런 다음 **3단계**로 이동합니다.
3. **등록정보**를 선택한 경우 **등록정보 이름**을 입력하거나 선택합니다.
4. 이름, 별칭 또는 등록정보 값을 입력합니다.
5. **확인**을 누릅니다.

별칭, 이름 또는 등록정보가 발견되면 **결과**에 표시됩니다. 아웃라인은 표시되지 않고 발견된 멤버만 표시됩니다.

 주:

별칭으로 멤버를 검색하는 경우 다른 언어로 된 별칭을 가진 멤버를 포함하여 검색 조건과 일치하는 별칭을 가진 모든 멤버가 결과에 표시됩니다. 아웃라인은 표시되지 않고 발견된 멤버만 표시됩니다. 그러나 사용되는 언어로 된 멤버 별칭만 멤버 선택기에 표시됩니다.

6. **선택 사항:** 둘 이상의 멤버가 검색 조건과 일치하는 경우 위쪽 및 아래쪽 화살표나 스크롤 막대(사용 가능한 경우)를 사용하여 위/아래로 이동하면서 검색 조건과 일치하는 모든 멤버를 찾습니다.

## 변수 작업

비즈니스 규칙 및 템플릿을 디자인할 때 구성요소에 변수를 사용합니다.



## 참조:

- [변수 정보](#)
- [변수 생성](#)
- [런타임 프롬프트 변수 입력](#)
- [변수 선택](#)
- [변수 편집](#)
- [변수 삭제](#)
- [변수 새로그침](#)
- [변수 복사](#)
- [변수 디자이너에서 텍스트 찾기 및 바꾸기](#)
- [변수 사용 표시](#)
- [사전 정의된 시스템 변수를 Financial Management 애플리케이션으로 로드\(Financial Management 사용자만 해당\)](#)

## 변수 정보

변수는 사용자가 정의하는 값을 갖습니다. 비즈니스 규칙 및 템플릿을 디자인할 때 구성요소에 변수를 사용할 수 있습니다.

다음과 같은 방법으로 변수를 생성할 수 있습니다.

-  을 눌러 변수 디자이너를 실행합니다.
- 규칙, 스크립트, 공식, 템플릿 등 변수 선택기 대화상자를 사용할 수 있는 임의 위치에서 변수를 생성합니다.  
예를 들면 다음과 같습니다.
  - 규칙을 연 다음 멤버 범위 구성요소를 끕니다.
  - 차원 옆에 있는  을 누른 다음 변수를 선택합니다.
  - **변수 선택** 대화상자에서 **생성**을 눌러 변수를 생성합니다.

변수를 생성하면 글로벌, 애플리케이션, 계획 유형, 비즈니스 규칙 등 선택한 레벨에 변수가 생성됩니다. 각 레벨에서 이름이 동일한 변수를 생성하면 규칙에서 최하위 레벨 변수를 사용합니다. 예를 들어 Month라는 글로벌 변수와 Month라는 계획 유형 변수를 생성하면 규칙에서 계획 유형의 변수를 사용합니다.

변수에는 다음 두 가지 유형이 있습니다.

- **실행** - 비즈니스 규칙이 실행되면 변수에 대해 정의된 계산이 수행됩니다. 스크립트 구성요소 또는 고정 루프 구성요소에서 실행 변수를 사용할 수 있습니다.

 주:

Oracle General Ledger에서 사용되는 애플리케이션을 포함하여 Oracle Essbase 집계 저장영역 애플리케이션에서는 실행 변수가 지원되지 않습니다.

- **대체** - 비즈니스 규칙을 디자인하거나 실행하는 경우 변수는 계산 내에서 대체됩니다. 모든 구성요소에서 대체 변수를 사용할 수 있습니다.

여러 유형의 실행 및 대체 변수를 생성할 수 있습니다. 생성할 수 있는 변수는 애플리케이션 유형과 실행 변수를 생성 중인지 또는 대체 변수를 생성 중인지에 따라 다릅니다.

 주:

Groovy 이외의 규칙인 경우, 하나 이상의 **멤버** 유형의 변수는 ASO(집계 저장영역 옵션) 유형의 Oracle Hyperion Planning 큐브에서만 지원되는 변수 유형입니다.

비즈니스 규칙을 실행할 때 정보 입력을 요청하는 메시지를 표시하는 변수를 생성할 수 있습니다. 이러한 런타임 프롬프트 변수는 멤버, 텍스트, 날짜 또는 숫자 같은 정보에 대해 사용자에게 변수 프롬프트를 표시합니다. 이 프롬프트는 어떤 유형의 데이터를 입력해야 하는지 알려줍니다.


예를 들면 다음과 같습니다.

- 월을 선택합니다.
- 분기당 예상 고객 방문 수를 입력합니다.
- 다음 달의 예상 수입 변동률을 지정합니다.

생성하려는 변수가 적용되는 애플리케이션 유형에 따라, 변수 한 개와 연결할 수 있는 데이터베이스 객체는 최대 4개입니다. 변수는 동시에 여러 객체에 있을 수 있고 각 객체에서 동일한 이름을 가질 수 있습니다.

## 변수 생성

변수를 생성하려면 다음을 수행합니다.

1. 시스템 뷰, 사용자정의 뷰, 필터 뷰 또는 배포 뷰에서  을 누릅니다.
2. 변수 네비게이터에서 애플리케이션 유형을 확장합니다.  
예를 들어 계획을 확장합니다.
3. 변수를 생성할 레벨을 선택합니다.



- **글로벌** - 동일한 애플리케이션 유형의 모든 애플리케이션에서 사용할 수 있는 변수를 생성하려면 <글로벌>을 마우스 오른쪽 버튼으로 누르고 **새로 작성**을 선택합니다.

Oracle Hyperion Financial Management 및 Oracle Hyperion Planning 사용자:

- **애플리케이션** - 해당 애플리케이션에서만 사용할 수 있는 변수를 생성하려면 애플리케이션을 마우스 오른쪽 버튼으로 누르고 **새로 작성**을 선택합니다.
- **계획 또는 데이터베이스** - 해당 계획 유형 또는 데이터베이스에서만 사용할 수 있는 변수를 생성하려면 계획 유형 또는 데이터베이스를 마우스 오른쪽 버튼으로 누르고 **새로 작성**을 선택합니다.

Planning, Oracle General Ledger 및 Oracle Essbase 사용자만 해당:

- **비즈니스 규칙** - 해당 규칙에서만 사용할 수 있는 변수를 생성하려면 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **새로 작성**을 선택합니다.

Planning, Oracle General Ledger 및 Essbase 사용자만 해당:

#### 4. 생성할 변수 유형을 선택합니다.

- **대체** - 비즈니스 규칙을 디자인하거나 실행할 때 변수는 계산으로 대체됩니다. 모든 구성요소에서 대체 변수를 사용할 수 있습니다.

대체 변수를 생성하려면 **대체** 탭에서 **작업, 새로 작성** 순으로 선택하고 다음 정보를 입력합니다.

- **이름** - 변수 이름
- **설명** - 변수에 대한 설명
- **그룹** - 이 변수를 그룹에 포함하려면 그룹 이름을 입력합니다. 변수를 저장한 후 새로고치면 **그룹** 열 아래에 그룹 이름이 표시됩니다.
- **유형** - 드롭다운을 누르고 유형을 선택합니다.
- **RTP** - 런타임 프롬프트 변수인 경우
  - \* **RTP**를 선택하고 변수를 사용할 때마다 표시할 텍스트를 입력합니다.
  - \* 필요한 경우 **기본값**을 입력합니다.
  - \* 프롬프트에 대해 마지막으로 입력된 값을 다음에 프롬프트를 표시할 때 기본값으로 표시하려면 **마지막 입력한 값 사용**을 선택합니다.

- **실행** - 비즈니스 규칙이 실행되면 변수에 대해 정의된 계산이 수행됩니다. 스크립트 구성요소 또는 고정 루프 구성요소에서 실행 변수를 사용할 수 있습니다.

실행 변수를 생성하려면 **실행** 탭에서 **작업, 새로 작성** 순으로 선택하고 다음 정보를 입력합니다.

- **이름** - 변수 이름
- **그룹** - 이 변수를 그룹에 포함하려면 그룹 이름을 입력합니다. 변수를 저장한 후 새로고치면 **그룹** 열 아래에 그룹 이름이 표시됩니다.
- **값** - 변수 값
  - \* 숫자 변수의 경우 **숫자 변수에 대한 변수 값 입력**을 참조하십시오.
  - \* 문자열 변수의 경우 **문자열 변수에 대한 변수 값 입력**을 참조하십시오.
  - \* 멤버 범위 변수의 경우 **멤버 또는 멤버 변수에 대한 변수 값 입력**을 참조하십시오.
  - \* 데이터 범위 변수의 경우 **데이터 범위 변수에 대한 값 입력(Financial Management 사용자만 해당)**을 참조하십시오.

- \* 로그 정보 변수의 경우 [로그 정보 변수에 대한 값 입력\(Financial Management 사용자만 해당\)](#)을 참조하십시오.

5.  을 누릅니다.

## 숫자 변수에 대한 변수 값 입력

숫자 변수는 Oracle Hyperion Planning, Oracle Essbase, Oracle General Ledger 또는 Oracle Hyperion Financial Management 대체 변수일 수 있습니다. 숫자 변수는 Financial Management 실행 변수일 수도 있습니다.

숫자 변수에 대한 값을 입력하려면 다음을 수행합니다.

1. 유형에서 숫자를 선택합니다.
2. 스마트 목록을 사용하려면 애플리케이션, 계획 유형 또는 규칙 레벨에서 변수를 생성합니다. 글로벌 레벨에서는 스마트 목록을 사용할 수 없습니다. 스마트 목록 상자를 누르고 드롭다운 목록을 누른 다음 스마트 목록을 선택합니다.

Planning 사용자만 해당

이 릴리스에 대한 *Administering Planning for Oracle Planning and Budgeting Cloud Service*를 참조하십시오. 예를 들어 보고 주기에 대해 연도별 (1), 분기별 (2), 월별 (3), 일별 (4) 및 시간별 (5)에 해당하는 값 1-5를 갖는 정수 스마트 목록을 설정할 수 있습니다. 사용자가 "월별"을 선택하면 숫자 3이 데이터베이스에 저장됩니다. 따라서 사용자가 숫자를 기억할 필요가 없습니다.

스마트 목록에 대한 값으로 텍스트 또는 날짜 문자열을 설정할 수도 있습니다.

 주:

스마트 목록 사용을 지원하는 Planning 애플리케이션을 선택해야 합니다.

3. 숫자 런타임 프롬프트를 사용하려면 **스마트 목록** 상자를 비워 두고 다음 단계로 이동합니다.
4. **(Planning 사용자만 해당) 제한**에서 스마트 목록에 대한 제한을 선택합니다.

 주:

런타임 프롬프트 값으로 Essbase 대체 변수를 사용하며 해당 대체 변수의 값이 변수 제한을 벗어날 경우에도 규칙 실행 시 오류가 발생하지 않으며 성공적으로 실행됩니다.

5. **선택 사항:** 변수 기본값을 입력합니다.
6. 기본적으로 **RTP**가 선택되어 있습니다. 이 변수에 대해 런타임 프롬프트를 생성하지 않으려면 **RTP**를 선택 취소합니다.

Planning 사용자만 해당)

 주:


RTP를 선택하지 않을 경우 변수의 기본값을 입력해야 합니다.

7. **RTP**를 선택한 경우 사용자에게 대한 기본값으로 표시할 런타임 프롬프트 텍스트를 입력합니다.

Planning 및 Essbase 사용자만 해당

8. 누락된 데이터 값의 허용 여부를 지정합니다.

Planning 사용자만 해당

9. 를 누릅니다.

Financial Management 대체 또는 실행 숫자 변수에 대한 값을 입력하려면 다음을 수행합니다.

1. 값을 입력하거나 **값** 필드를 눌러 **작업** 아이콘을 표시합니다.
2. 다음 중 한 가지 태스크를 수행합니다.
  - 멤버를 입력하려면 **멤버**를 선택합니다. [구성요소에 멤버 및 함수 추가](#)를 참조하십시오.
  - 함수를 입력하려면 **함수**를 선택합니다. [함수 작업](#)을 참조하십시오.

3. 를 누릅니다.

## 문자열 변수에 대한 변수 값 입력

문자열 변수는 Oracle Hyperion Financial Management, Oracle Hyperion Planning, Oracle General Ledger 또는 Oracle Essbase 대체 변수일 수 있습니다. 문자열 변수는 Financial Management 실행 변수일 수도 있습니다. 문자열 변수는 영숫자여야 하며 255자 이내여야 합니다. 이 변수에는 null 값이 포함될 수 있지만 값 맨 앞에 &(앰퍼샌드) 문자가 나오면 안 됩니다.

대체 문자열 변수에 대한 값을 입력하려면 다음을 수행합니다.

1. 유형에서 **문자열**을 선택합니다.
2. **값** 테이블에 변수 값을 입력합니다.
3. 기본적으로 **RTP**가 선택되어 있습니다. 이 변수에 대해 런타임 프롬프트를 생성하지 않으려면 **RTP**를 선택 취소합니다.

Planning 사용자만 해당



**주:**  
RTP를 선택하지 않을 경우 변수의 기본값을 입력해야 합니다.


4. **RTP**를 선택한 경우 표시하려는 런타임 프롬프트 텍스트를 입력합니다.

Planning 및 Essbase 사용자만 해당

5. 를 누릅니다.

Financial Management 대체 또는 실행 문자열 변수에 대한 값을 입력하려면 다음을 수행합니다.

1. 유형에서 **문자열**을 선택합니다.
2. 값을 입력하거나 **값** 필드를 눌러 **작업** 아이콘을 표시합니다.

3. 다음 중 한 가지 태스크를 수행합니다.
  - 멤버를 입력하려면 **멤버**를 선택합니다. **구성요소에 멤버 및 함수 추가**을 참조하십시오.
  - 함수를 입력하려면 **함수**를 선택합니다. **함수 작업**을 참조하십시오.
4. 를 누릅니다.


## 배열 변수 값 입력

배열 변수는 Oracle Hyperion Financial Management, Oracle Hyperion Planning, Oracle General Ledger 또는 Oracle Essbase 실행 변수일 수 있습니다. 배열에는 다차원일 수 있는 값 목록이 포함되어 있습니다.

일반적으로 배열은 변수를 멤버 공식의 일부로 저장하는 데 사용됩니다. 배열 변수의 크기는 해당 차원에 있는 멤버의 수에 의해 결정됩니다. 예를 들어 시나리오 차원에 4개의 멤버가 있을 경우 다음 명령은 항목이 4개인 Discount라는 배열을 생성합니다. 한 번에 둘 이상의 배열을 사용할 수 있습니다.

```
ARRAY Discount[Scenario];
```

배열 변수의 값을 입력하려면 다음을 수행합니다.

1. **배열 확인란**을 선택하여 이 실행 변수를 배열로 설정합니다. **배열**을 선택한 후 드롭다운 목록에서 차원을 선택해야 합니다. 그렇지 않으면 기본적으로 계획 유형의 차원으로 설정됩니다.
2. **값 필드**에 계획 레벨에서 변수를 생성하는 값을 입력합니다.
3. **선택 사항:** 변수 그룹을 입력합니다.
4. 를 누릅니다.

## 멤버 범위 변수 값 입력

멤버 범위 변수는 Oracle Hyperion Financial Management, Oracle Hyperion Planning, Oracle General Ledger 또는 Oracle Essbase 대체 변수일 수 있습니다. 멤버 범위 변수에는 멤버 범위가 포함되어야 합니다.

멤버 범위 변수 값을 입력하려면 다음을 수행합니다.

1. **유형**에서 **멤버 범위**를 선택합니다.
2. Financial Management, Planning, Oracle General Ledger 또는 Essbase에 대해 멤버 범위를 생성하는지에 따라 **가변 그리드**에서 다음 중 하나를 수행합니다.
  - a. Financial Management에 대해 멤버 범위 변수를 생성하는 경우
    - i. 멤버 선택을 제한하는 차원을 선택합니다. 차원을 선택할 경우 멤버 선택기에는 해당 차원에 대한 멤버만 표시됩니다. 차원을 선택하지 않으면 멤버 선택기에 모든 차원이 표시됩니다.
    - ii. **값**에 변수 값을 입력하거나 **작업** 아이콘을 선택하여 해당 변수 값에 대한 멤버, 변수 또는 함수를 입력합니다.
      - 변수를 입력하려면 **변수 작업**을 참조하십시오.
      - 멤버를 입력하려면 **구성요소에 멤버 및 함수 추가**을 참조하십시오.
      - 함수를 입력하려면 **함수 작업**을 참조하십시오.
    - iii. **변수**에서 **변수** 아이콘을 선택하여 멤버 범위에 대한 변수를 선택합니다.

- iv. 설명 아이콘을 누르고 변수에 대한 설명을 입력합니다.
  - v. 멤버 범위를 정의할 각 차원에 대해 이 단계를 반복합니다.
- b. Planning, Oracle General Ledger 또는 Essbase에 대해 멤버 범위 변수를 생성 중인 경우
- i. 멤버 범위 제한을 선택할 테이블의 각 차원에 대해 제한 필드를 누르고 제한을 입력합니다. 표시되는 차원은 변수를 생성 중인 애플리케이션에 속하는 차원입니다.

글로벌 레벨에서 [차원 유형] 옵션을 선택하면 표준 차원 유형만 표시됩니다. [차원 이름] 옵션을 선택하는 경우 아무 차원 이름이나 입력할 수 있습니다.

 주:

**(Planning 및 Essbase 사용자만 해당)** [한도] 필드에 텍스트를 입력하기 전에 RTP를 선택해야 합니다.


 주:

한도에 대해 함수를 사용하는 경우 Essbase 함수 대신 Planning 함수를 사용하는 것이 좋습니다. Essbase 함수 평가에 동적 멤버가 포함된 경우와 같이 Essbase 함수는 필요한 멤버를 반환하지 않는 경우가 있습니다. 멤버 범위 변수의 한도에 Essbase 함수 "@Relative("Mbr Name", 0)" 대신 "ILvl0Descendants("Mbr Name")"를 사용하십시오.

- ii. 멤버 범위에 대한 기본값을 입력하거나 멤버 선택기를 사용하여 선택합니다. 또는 변수가 런타임 프롬프트인 경우 기본값을 비워 둘 수 있습니다. 나열된 각 차원에 대해 멤버 및 함수를 여러 개 선택할 수 있습니다.
- iii. **(Planning 및 Essbase 사용자만 해당)** 변수가 실행될 때 정보를 묻는 메시지를 표시하게 하려면 나열된 각 차원에 대해 **RTP**를 선택합니다.

 주:

RTP를 선택하지 않을 경우 변수 값을 입력해야 합니다.

- iv. **(Planning 및 Essbase 사용자만 해당)** RTP를 선택한 각 차원에 대해 런타임 프롬프트 텍스트를 입력합니다.
  - v. 그리드 위쪽의 RTP 텍스트 상자에서 해당 차원에 대해 변수가 실행될 때마다 표시할 RTP 텍스트를 입력합니다.
3.  를 누릅니다.

## 횡차원 변수에 대한 변수 값 입력

횡차원 변수는 Oracle Hyperion Planning, Oracle General Ledger 또는 Oracle Essbase 대체 변수입니다. 이 변수에는 차원 간에 비즈니스 규칙을 실행할 수 있는 여러 차원의 멤버가 포함됩니다.

횡차원 변수에 대한 값을 입력하려면 다음을 수행합니다.

1. 유형에서 **횡차원**을 선택합니다.
2. **선택 사항:** 변수에 대한 제한을 선택할 테이블의 각 차원에 대해 제한 필드를 누르고 제한을 입력합니다. 표시되는 차원은 변수를 생성 중인 애플리케이션에 속하는 차원입니다.

### 주:

- [제한] 필드에 텍스트를 입력하려면 먼저 RTP를 선택해야 합니다.
- 한도에 대해 함수를 사용하는 경우 Essbase 함수 대신 Planning 함수를 사용하는 것이 좋습니다. Essbase 함수 평가에 동적 멤버가 포함된 경우와 같이 Essbase 함수는 필요한 멤버를 반환하지 않는 경우가 있습니다. 횡차원 변수의 한도에 Essbase 함수 "@Relative("Mbr Name", 0)" 대신 "lLvlODescendants("Mbr Name")"를 사용하십시오.
- 함수를 사용할 수 있지만 함수 선택이 적합하려면 차원의 단일 멤버를 반환해야 합니다.

3. 변수 값을 입력하거나 멤버 선택기에서 선택합니다. 멤버 또는 함수를 선택할 수 있습니다.
4. 변수에 런타임 프롬프트가 있는 경우 **RIP**를 선택합니다.

Planning 및 Essbase 사용자만 해당

### 주:

- RTP를 선택하지 않을 경우 변수 값을 입력해야 합니다.
- 함수를 사용할 수 있지만 함수가 적합한 선택이 되려면 차원의 단일 멤버를 반환해야 합니다.

5. 런타임 프롬프트 텍스트를 입력합니다.

Planning 및 Essbase 사용자만 해당


6. 를 누릅니다.

## 차원 변수에 대한 변수 값 입력

차원 변수는 Oracle Hyperion Planning, Oracle General Ledger 또는 Oracle Essbase 대체 변수입니다. 이 변수에는 선택한 차원이 포함되어 있습니다.

차원 변수에 대한 값을 입력하려면 다음을 수행합니다.

1. 유형에서 **차원**을 선택합니다.

2. **변수 그리드**에서 차원을 선택합니다. 표시되는 차원은 변수를 생성 중인 애플리케이션에 속하는 차원입니다.
3. 기본적으로 **RTP**가 선택되어 있습니다. 변수에 대해 런타임 프롬프트를 지정하지 않으려면 **RTP**를 선택 취소합니다.  
RTP를 선택하지 않을 경우 변수 값을 입력해야 합니다.
4. **(Planning 및 Essbase 사용자만 해당) RTP**를 선택한 경우 사용자에게 표시할 런타임 프롬프트 텍스트를 입력합니다.
5. 를 누릅니다.

## 멤버 또는 멤버 변수에 대한 변수 값 입력

멤버 및 멤버(둘 이상) 변수는 Oracle Hyperion Planning, Oracle General Ledger 및 Oracle Essbase 대체 변수입니다. 이러한 변수에는 선택한 차원의 단일 멤버 또는 여러 멤버가 포함되어 있습니다.

멤버 또는 멤버(둘 이상) 변수에 대한 값을 입력하려면 다음을 수행합니다.

1. 유형에서 **멤버** 또는 **멤버(둘 이상)**를 선택합니다.
2. **변수 그리드**에서 차원을 선택합니다. 표시되는 차원은 변수를 생성할 애플리케이션에 속하는 차원입니다.
3. 변수에 대한 제한을 입력하거나 멤버 선택기에서 선택합니다. 2단계에서 선택한 차원에서만 멤버를 선택할 수 있습니다. 함수를 선택할 수도 있습니다. [함수 작업을](#) 참조하십시오.

### 주:

한도에 대해 함수를 사용하는 경우 Essbase 함수 대신 Planning 함수를 사용하는 것이 좋습니다. Essbase 함수 평가에 동적 멤버가 포함된 경우와 같이 Essbase 함수는 필요한 멤버를 반환하지 않는 경우가 있습니다. 멤버 변수 또는 멤버(둘 이상) 변수의 한도에 Essbase 함수 "@Relative("Mbr Name", 0)" 대신 "lLv10Descendants("Mbr Name")"를 사용하십시오.

4. 변수 기본값을 입력하거나 멤버 선택기에서 선택합니다. *member* 변수에 대해 단일 멤버 또는 함수를 선택하고 *members* 변수에 대해 여러 멤버 및 함수를 선택할 수 있습니다.
5. 기본적으로 **RTP**가 선택되어 있습니다. 런타임 프롬프트 변수를 생성하지 않으려면 **RTP**를 선택 취소합니다.  
Planning 사용자만 해당

### 주:

RTP를 선택하지 않을 경우 변수 값을 입력해야 합니다.

6. **RTP**를 선택한 경우 표시하려는 런타임 프롬프트 텍스트를 입력합니다.  
Planning 및 Essbase 사용자만 해당

7. Workforce Planning, Capital Asset Planning, Public Sector Budgeting 및 Project Financial Planning 계획 유형의 **Planning 사용자: 동적 멤버 생성 및/또는 동적 멤버 삭제** 설정이 사용으로 설정된 비즈니스 규칙의 멤버 유형 변수를 사용할 경우 **동적 멤버 상위** 열에 기본 동적 상위 멤버를 입력합니다. 이러한 옵션을 둘 다 사용할 경우 Planning에서는 규칙이 실행되기 전에 이 상위에 대한 멤버를 동적으로 **생성하고/**하거나 규칙이 실행된 후 이 상위에 대한 멤버를 **삭제**합니다.

 **주:**

Planning 데이터베이스 아웃라인에서 동적 1차 하위 구성요소에 대한 동적 멤버 상위를 활성화해야 합니다. 이 설정을 활성화한 후 데이터베이스를 새로 고쳐야 합니다.

8. **파일, 저장** 순으로 선택합니다.

## 퍼센트 변수에 대한 변수 값 입력

퍼센트 변수는 Oracle Hyperion Planning, Oracle General Ledger 및 Oracle Essbase 대체 변수입니다. 퍼센트 변수는 Planning 및 Essbase 블록 저장영역 실행 변수이기도 합니다. 이 변수에는 지정한 비율이 포함되어 있습니다.

비율 변수에 대한 값을 입력하려면 다음을 수행합니다.

1. 유형에서 **퍼센트**를 선택합니다.
2. **변수 그리드**에서 **제한**을 눌러 변수의 최소값 및 최대값을 정의합니다.
3. 변수에 대한 숫자 값을 입력합니다.
4. 기본적으로 **RTP**가 선택되어 있습니다. 런타임 프롬프트를 생성하지 않으려면 **RTP**를 선택 취소합니다.

Planning 사용자만 해당)

 **주:**

RTP를 선택하지 않을 경우 변수 값을 입력해야 합니다.

5. **RTP**를 선택한 경우 표시하려는 런타임 프롬프트 텍스트를 입력합니다.  
(Planning 및 Essbase 사용자만 해당)
6. 누락된 데이터 값의 허용 여부를 선택합니다.
7. **파일, 저장** 순으로 선택합니다.

## 정수 변수에 대한 값 입력

정수 변수는 Oracle Hyperion Planning, Oracle General Ledger 및 Oracle Essbase 대체 변수입니다.

정수 변수의 값을 입력하려면 다음을 수행합니다.

1. 유형에서 **정수**를 선택합니다.
2. **선택 사항: 값** 테이블에서 **제한**을 눌러 변수의 최소 및 최대 정수 값을 정의합니다.




3. **선택 사항:** 기본값을 누르고 변수에 사용할 정수를 입력합니다.
4. 기본적으로 **RTP**가 선택되어 있습니다. 이 변수에 대해 런타임 프롬프트를 생성하지 않으려면 **RTP**를 선택 취소합니다.  
Planning 사용자만 해당  
RTP를 선택하지 않을 경우 변수의 기본값을 입력해야 합니다.
5. **RTP**를 선택한 경우 표시하려는 런타임 프롬프트 텍스트를 입력합니다.  
Planning 및 Essbase 사용자만 해당
6. #Missing 값의 허용 여부를 선택합니다.
7. **파일, 저장** 순으로 선택합니다.

## 문자열(숫자) 변수에 대한 변수 값 입력

문자열(숫자) 변수는 Oracle Hyperion Planning 대체 또는 실행 변수일 수 있습니다.

문자열(숫자) 변수에 대한 값을 입력하려면 다음을 수행합니다.

1. 유형에서 **문자열(숫자)**을 선택합니다.
2. **선택 사항: 값** 테이블에서 **제한**을 눌러 변수의 최소값 및 최대값을 정의합니다. 최소값과 최대값은 YYYYMMDD 형식의 숫자로 입력해야 합니다.
3. **선택 사항:** 변수에 대한 숫자 값을 입력합니다.
4. 기본적으로 **RTP**가 선택되어 있습니다. StringAsNumber 변수의 경우 **RTP**가 필요하므로 **RTP** 확인란의 선택을 취소할 수 없습니다.
5. #Missing 값의 허용 여부를 선택합니다.
6. **마지막 입력한 값 사용**을 선택하여 사용자가 마지막으로 입력한 값을 사용할 수 있게 합니다.
7. 를 누릅니다.

## 날짜(숫자) 변수에 대한 변수 값 입력

날짜(숫자) 변수는 Oracle Hyperion Planning 대체 또는 실행 변수일 수 있습니다.

날짜(숫자) 변수에 대한 값을 입력하려면 다음을 수행합니다.


1. 유형에서 **날짜(숫자)**를 선택합니다.
2. **선택 사항: 값** 테이블에서 **제한**을 눌러 변수의 최소값 및 최대값을 정의합니다. 최소값과 최대값은 YYYYMMDD 형식의 숫자로 입력해야 합니다.
3. **선택 사항:** 변수에 대한 숫자 값을 입력합니다. 날짜(숫자) 변수의 경우 YYYYMMDD 숫자 형식으로 날짜를 입력합니다.
4. 기본적으로 **RTP**가 선택되어 있습니다. 런타임 프롬프트를 생성하지 않으려면 **RTP**를 선택 취소합니다.  
RTP를 선택하지 않을 경우 변수 값을 입력해야 합니다.
5. **RTP**를 선택한 경우 표시하려는 런타임 프롬프트 텍스트를 입력합니다.
6. #Missing 값의 허용 여부를 선택합니다.
7. **마지막 입력한 값 사용**을 선택하여 사용자가 마지막으로 입력한 값을 사용할 수 있게 합니다.

8. 를 누릅니다.

## 데이터 범위 변수 값 입력(Financial Management 사용자만 해당)

데이터 범위 변수는 Oracle Hyperion Financial Management 대체 변수입니다. 여기에는 선택한 차원의 데이터 값 범위가 포함되어 있습니다.


데이터 범위 변수에 대한 값을 입력하려면 다음을 수행합니다.

1. 유형에서 **데이터 범위**를 선택합니다.
2. 값 테이블에서 나열된 각 차원에 대해 **작업** 아이콘을 선택하여 변수 값을 입력합니다.
  - 변수를 입력하려면 **변수 작업**을 참조하십시오.
  - 멤버를 입력하려면 **구성요소에 멤버 및 함수 추가**를 참조하십시오.
  - 함수를 입력하려면 **함수 작업**을 참조하십시오.
3. 변수에서 **변수** 아이콘을 눌러 데이터 범위에 대한 변수를 선택합니다.
4. **설명** 아이콘을 누르고 변수에 대한 설명을 입력합니다.
5. 데이터 범위를 정의할 각 차원에 대해 이 단계를 반복합니다.
6. 를 누릅니다.

## 부울 변수 값 입력(Financial Management 사용자만 해당)

부울 변수는 Oracle Hyperion Financial Management 실행 변수입니다. 부울 변수는 true 또는 false 값을 갖는 변수입니다.


부울 변수에 대한 값을 입력하려면 다음을 수행합니다.

1. 유형에서 **부울**을 선택합니다.
2. 값에 변수 값을 입력합니다.
3. 를 누릅니다.

## 로그 정보 변수 값 입력(Financial Management 사용자만 해당)

로그 정보 변수는 Oracle Hyperion Financial Management 대체 변수입니다.

로그 정보 변수에 대한 값을 입력하려면 다음을 수행합니다.

1. 유형에서 **로그 정보**를 선택합니다.
2. 값 테이블에 로그 텍스트 및 로그 표현식을 입력하거나 **조건 추가/편집**을 눌러 조건 빌더를 사용합니다. **조건 빌더를 사용하여 조건문 생성**을 참조하십시오.
3. 를 누릅니다.

## 런타임 프롬프트 변수 입력

### 주:

- Oracle Hyperion Calculation Manager의 Oracle Essbase 비즈니스 규칙 스크립트에 RUNTIMESUBVARS 섹션을 추가할 수 없습니다. 계산 관리자에 생성된 규칙 레벨 런타임 프롬프트 변수는 Essbase 애플리케이션을 배포할 때만 RUNTIMESUBVARS로 변환됩니다.
- 관리 서비스, MaxL 또는 계산 스크립트를 실행할 수 있는 모든 구성요소에서 런타임 프롬프트로 business rule을 실행하는 경우, 이 business rule의 런타임 프롬프트에 기본값이 있어야 합니다.
- Calculation Manager에서 규칙을 검증, 디버깅, 배포, 분석 또는 실행할 때 런타임 프롬프트 변수의 값을 입력하거나 편집할 수 있습니다. 비즈니스 규칙 세트를 검증하거나 배포할 때 런타임 프롬프트 변수를 입력하거나 편집할 수도 있습니다. 런타임 프롬프트에 멤버 한도가 있는 경우 검증 시 적합한 멤버 이름만 확인합니다(멤버가 한도 내에 있는지 여부는 검증하지 않음). 숫자 및 정수 유형 변수는 런타임 프롬프트 한도가 검증됩니다.
- 런타임 프롬프트가 포함된 Essbase 비즈니스 규칙을 Essbase에 배포하는 경우 Oracle Essbase Administration Services, MaxL 또는 계산 스크립트를 실행할 수 있는 모든 구성요소를 사용하여 비즈니스 규칙을 실행할 수 있습니다. 이러한 비즈니스 규칙은 배포할 때 Essbase 계산 스크립트로 변환되며, 런타임 프롬프트 변수는 Essbase 스크립트에서 RUNTIMESUBVARS로 변환됩니다.
- 멤버 승인 보안을 적용하도록 런타임 프롬프트를 디자인하는 방법에 대한 자세한 내용은 *Oracle Hyperion Planning 관리자 가이드*에서 [승인 보안의 런타임 프롬프트 정보](#)를 참조하십시오.

런타임 프롬프트 변수의 값을 입력하려면 다음을 수행합니다.

1. 비즈니스 규칙을 검증, 디버깅, 배포, 분석 또는 실행하거나 비즈니스 규칙 세트를 검증 또는 배포하는 경우 오류가 없으면 **RTP 값 입력** 대화상자가 표시됩니다.

비즈니스 규칙을 검증, 디버깅, 배포 또는 분석하거나 비즈니스 규칙 세트를 검증 또는 배포할 때 비즈니스 규칙이나 비즈니스 규칙 세트에서 사용 중인 런타임 프롬프트 변수 중 하나 이상의 값이 누락된 경우에만 [RTP 값 입력] 대화상자가 표시됩니다. 모든 런타임 프롬프트 변수에 값이 있으면 [RTP 값 입력] 대화상자가 표시되지 않습니다.

비즈니스 규칙을 실행하는 경우에는 런타임 프롬프트 변수에 값이 있는지 여부에 관계없이 매번 [RTP 값 입력] 대화상자가 표시됩니다. 값이 있는 경우 해당 값이 기본적으로 [RTP 값 입력] 대화상자가 표시됩니다.

2. 나열되는 각 런타임 프롬프트에 대해 값을 입력하거나 선택합니다.
3. **선택 사항:** 비즈니스 규칙으로 작업하는 경우 제공하는 값이 변수 값에서 동적으로 업데이트되고 규칙 디자이너 내 변수 탭의 값 열에 표시되도록 **규칙에 값 적용 확인란**을 선택합니다.

시스템 뷰에서 비즈니스 규칙을 검증하는 경우 이 확인란을 사용할 수 없습니다.

4. **확인**을 누릅니다.

5. 검증 오류가 있는 경우 해당 오류를 수정하고 런타임 프롬프트 변수 값을 입력할 태스크를 반복합니다.

## 변수 선택


여러 위치에서 변수를 선택할 수 있습니다. 구성요소 디자이너에서 구성요소를 생성할 때, 템플릿 디자이너에서 디자인 타임 프롬프트를 생성할 때 및 Oracle Hyperion Calculation Manager의 기타 위치에서 변수를 선택할 수 있습니다.

변수를 선택하려면 다음을 수행합니다.

1. 다음 태스크 중 하나를 수행합니다.
  - 변수를 추가하려는 구성요소가 들어 있는 템플릿을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
  - 변수를 추가하려는 구성요소가 들어 있는 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
2. 비즈니스 규칙이나 템플릿이 열리면 순서도에서 변수를 삽입하려는 구성요소를 선택합니다.
3. 순서도 아래의 탭에서 다음 태스크 중 하나를 수행합니다.
  - 멤버 범위 구성요소의 경우 차원 필드를 누르고 **작업** 아이콘을 선택한 다음 **변수**를 선택하여 멤버 범위 변수를 선택합니다. 또는 **변수 선택기**를 눌러 멤버 범위 변수를 선택합니다.
  - 데이터 범위 및 고정 루프 구성요소에 대해 **변수** 필드에서 **변수** 아이콘을 선택합니다.
  - 공식 구성요소의 경우 **작업** 아이콘을 누르고 **변수**를 선택합니다.
  - 스크립트 구성요소의 경우 **변수 삽입** 아이콘을 누릅니다.
  - 조건 구성요소의 경우 **조건 빌더**를 실행하고 **작업** 아이콘을 누른 후 **변수**를 선택합니다.
4. **변수 선택**에서 다음 태스크 중 하나를 수행합니다.
  - 새 변수를 생성하려면 **생성**을 눌러 **변수 디자이너**에 액세스합니다. **변수 생성**을 참조하십시오.
  - 기존 변수를 선택하려면 **범주**에서 사용할 변수가 포함된 레벨을 선택합니다. 다음을 선택할 수 있습니다.
    - **글로벌**: 변수는 글로벌 레벨에서 생성되며 이 애플리케이션 유형 내의 모든 애플리케이션에서 사용할 수 있습니다.  
Oracle Hyperion Financial Management 및 Oracle Hyperion Planning 사용자만 해당
    - **애플리케이션**: 변수는 애플리케이션 레벨에서 생성되며 이 애플리케이션에서만 표시됩니다.
    - **계획 유형 또는 데이터베이스**: 변수는 계획 유형 또는 데이터베이스 레벨에서 생성되며 이 계획 유형 또는 데이터베이스에서만 표시됩니다.  
Planning, Oracle Fusion General Ledger 및 Oracle Essbase 사용자만 해당
    - **규칙**: 변수는 규칙 레벨에서 생성되며 변수가 생성된 규칙에서만 표시됩니다.  
Planning, General Ledger 및 Essbase 사용자만 해당

 주:

Planning 및 Essbase 사용자만 해당: 멤버 범위, 멤버 블록, 공식, 고정 루프 및 조건 구성요소의 경우 구성요소를 사용하는 예상 변수 유형으로 표시 변수가 제한되므로 기본적으로 선택한 범위에서 사용 가능한 변수 중 일부만 표시됩니다. 선택된 범위의 모든 사용가능 변수를 보려면 [모든 변수 표시] 확인란을 선택합니다.

5. **대체** 또는 **실행**에서 구성요소에 삽입할 변수를 하나 이상 선택합니다.
6. **확인**을 누르고  를 누릅니다.

## 변수 편집

변수 디자이너에서 변수 등록정보를 편집할 수 있습니다. 변수를 변경할 때 해당 변수가 규칙에서 사용되고 있으면 규칙을 열어 저장하고 검증한 다음 재배포해야 합니다. [검증 및 배포](#)를 참조하십시오.

## 변수 삭제

구성요소나 멤버 공식에 사용되지 않는 변수는 변수 디자이너에서 삭제할 수 있습니다. 변수가 구성요소에서 사용되는 경우 먼저 구성요소에서 변수를 제거해야만 변수를 삭제할 수 있습니다.

변수를 삭제하려면 다음을 수행합니다.

1. **시스템 뷰** 또는 **필터 뷰**에서 **변수 디자이너** 아이콘을 누릅니다.
2. 변수 네비게이터에서 애플리케이션 유형 및 애플리케이션을 확장합니다.
3. 다음 태스크 중 하나를 수행합니다.
  - 변수가 글로벌 변수이면 **<글로벌>**을 선택합니다.  
Oracle Hyperion Financial Management 및 Oracle Hyperion Planning 사용자만 해당
  - 변수가 애플리케이션 변수이면 변수가 연결되어 있는 애플리케이션을 선택합니다.
  - 변수가 계획 유형 또는 데이터베이스 변수이면 해당 변수가 연결된 계획 유형 또는 데이터베이스를 선택합니다.  
Planning, Oracle Fusion General Ledger 및 Oracle Essbase 사용자만 해당
  - 변수가 비즈니스 규칙 변수이면 해당 변수가 연결된 비즈니스 규칙을 선택합니다.  
Planning, General Ledger 및 Essbase 사용자만 해당

애플리케이션 유형, 애플리케이션, 계산 유형, 계획 유형 또는 데이터베이스, 비즈니스 규칙에 연결된 모든 변수는 **대체** 또는 **실행**에 표시됩니다.
4. **대체** 또는 **실행**에서 삭제하려는 변수를 마우스 오른쪽 버튼으로 누르고 **삭제**를 선택합니다.
5. **삭제 확인**에서 **예**를 선택하여 변수 삭제를 확인합니다.

## 변수 새로고침

변수를 추가, 삭제 또는 변경한 후에 변수 네비게이터에서 변수 목록을 새로고쳐 최신 목록을 표시할 수 있습니다.

변수 네비게이터에서 변수 목록을 새로고치려면 다음을 수행합니다.

1. **시스템 뷰** 또는 **필터 뷰**에서 **변수 디자이너** 아이콘을 누릅니다.
2. 변수 네비게이터에서 변수를 생성, 편집 또는 삭제합니다.
3. **대체** 또는 **실행** 탭 위에서 **새로고침** 아이콘을 누릅니다.

## 변수 복사

복사 및 붙여넣기를 사용하여 변수를 동일하거나 다른 변수 범위(즉, 글로벌 레벨, 애플리케이션 레벨, 통합/계획/데이터베이스 레벨 또는 규칙 레벨)에 복사할 수 있습니다. 복사할 변수의 이름이 변수를 복사할 대상 위치의 변수 이름과 동일한 경우 복사할 변수에 새 이름을 지정하거나 변수 복사를 건너뛰거나 변수 콘텐츠를 덮어씁니다.

변수를 복사하고 붙여넣으려면 다음을 수행합니다.

1. **시스템 뷰** 또는 **필터 뷰**에서 **변수 디자이너** 아이콘을 누릅니다.
  2. 변수 네비게이터에서 다음 태스크 중 하나를 수행합니다.
    - **Oracle Hyperion Financial Management 사용자만 해당:** [통합]을 확장하고, 복사할 변수가 포함된 항목에 따라 애플리케이션 또는 글로벌을 선택합니다.
    - **Oracle Hyperion Planning 사용자만 해당:** Planning을 확장하고 글로벌이나 복사하려는 변수와 연계된 애플리케이션, 계획 유형 또는 비즈니스 규칙을 선택합니다.
    - **Oracle Essbase 및 Oracle Fusion General Ledger 사용자만 해당:** Essbase를 확장하고, 복사하려는 변수와 연계된 애플리케이션, 데이터베이스 또는 비즈니스 규칙을 선택합니다.
- 선택한 객체와 연계된 변수는 **대체** 탭 및/또는 **실행** 탭에 각각 나열됩니다.
3. 복사할 변수를 마우스 오른쪽 버튼으로 누르고 **복사**를 선택합니다.
  4. 복사된 변수를 붙여넣을 대상 위치 또는 변수를 마우스 오른쪽 버튼으로 누르고 **붙여넣기**를 선택합니다. 예를 들어 Planning 글로벌 변수를 복사할 경우 해당 변수를 다른 Planning 글로벌 변수 또는 계획 유형 변수로 복사할 수 있습니다.
    - 변수를 복사한 후 이름이 동일한 변수가 포함된 위치로 붙여넣으려고 하면 **충돌 해결** 대화상자가 표시됩니다. 다음 중 한 가지 태스크를 수행합니다.
      - 변수에 새 이름을 지정합니다. 동일한 위치에서 변수 두 개가 동일한 이름을 가질 수는 없습니다.
      - 변수 복사를 건너뛰도록 지정합니다. 복사된 변수의 콘텐츠가 새 위치로 삽입되지 않습니다.
      - 변수를 덮어쓰도록 지정합니다. 복사된 변수의 콘텐츠가 새 위치로 삽입되지 않고 복사 대상 변수의 콘텐츠를 덮어씁니다.
    - 변수를 복사한 후 이름이 동일한 변수가 포함되지 않은 위치로 붙여넣으려고 하면 변수가 새 위치에 삽입됩니다.

## 변수 디자이너에서 텍스트 찾기 및 바꾸기

변수 디자이너에서 변수의 텍스트를 검색하고 바꿀 수 있습니다. 모든 범위의 변수를 검색할 수 있습니다(글로벌, 애플리케이션, 계획이나 데이터베이스 또는 business rule).

변수 디자이너의 찾기 기능에 이름을 입력하여 변수를 검색할 수 있습니다. 또한 변수에서 텍스트 문자열을 검색할 수 있습니다. 예를 들어 변수에서 사용되는 기본값을 검색할 수 있습니다.

기본적으로 Oracle Hyperion Calculation Manager는 변수의 기본값, 정의된 한계 및 프롬프트 텍스트를 사용하여 검색합니다. 검색에 변수의 기본 등록정보를 포함하는 경우, 계산 관리자에서 변수 이름, 그룹 및 설명을 사용하여 검색합니다.

변수 또는 텍스트 문자열의 모든 인스턴스를 바꾸거나 선택한 인스턴스를 바꿀 수 있습니다. 텍스트를 바꾸고 한계 필드에 smartlist 또는 숫자 유형의 변수가 포함되어 있는 경우, 모두 바꾸기 작업에서 제외됩니다. 이러한 필드를 바꾸면 변수 정의가 부정확한 상태가 될 수 있습니다(예: 변수에 부적합한 스마트 목록 이름이나 지정된 한계에 없는 기본값이 포함될 수 있음).

변수 디자이너에서 텍스트를 찾으려면 다음을 수행합니다.

1. 모든 뷰에서 **변수 디자이너** 아이콘을 누릅니다.
2. **변수 네비게이터**에서 검색하려는 애플리케이션 유형, 글로벌(Oracle Hyperion Planning 및 Oracle Hyperion Financial Management 사용자만 해당), 계획 유형 또는 데이터베이스, 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **찾기**를 선택합니다.
3. **임의 텍스트**에서 다음을 선택합니다.
  - **시작** - 사용자가 지정하는 문자로 이름이 시작되는 변수만 표시합니다.
  - **끝** - 사용자가 지정하는 문자로 이름이 끝나는 변수만 표시합니다.
  - **포함** - 사용자가 지정하는 문자가 이름에 포함된 변수만 표시합니다.
  - **일치** - 사용자가 지정하는 문자와 이름이 일치하는 변수만 표시합니다.
4. **검색 필드**에 검색할 변수의 텍스트를 입력합니다.
5. 다음 옵션 중 하나 이상 선택합니다.
  - 검색할 텍스트의 대/소문자가 **검색 필드**에 입력하는 텍스트의 대/소문자와 일치할 필요가 없으면 **대/소문자 무시**를 선택합니다.
  - 변수 이름, 설명 및 그룹을 검색에 포함하려면 **기본 등록정보 포함**을 선택합니다. 변수 디자이너를 실행할 때 대소문자 무시 및 기본 등록정보 포함이 기본적으로 선택됩니다. 이 확인란을 선택 취소하면, 변수 디자이너를 닫고 다시 열 때까지 확인란이 선택 취소된 상태로 남아 있습니다.
  - 선택한 노드의 여러 레벨에서 변수를 검색하려면 **하위 범위의 변수 포함**을 선택합니다. *애플리케이션 유형* 레벨(예: Oracle Essbase 또는 Planning)에서 변수를 검색하는 경우 기본적으로 이 확인란이 선택되고 변경할 수 없습니다. 이 옵션이 선택되면, 애플리케이션의 변수, 계획 유형, 통합 유형 또는 데이터베이스 및 business rules가 표시됩니다. 변수 이름, 설명, 기본값, 그룹 및 소유자의 열 외에도 애플리케이션 열, 계획 유형 열 및 규칙 열이 변수 목록에 표시됩니다.
 

*애플리케이션* 레벨에서 변수를 검색하는 경우 이 확인란은 선택되거나 선택 취소될 수 있습니다. 이 옵션을 선택하면, 애플리케이션의 변수, 해당 계획 유형, 통합 유형 또는 데이터베이스 및 business rules도 표시됩니다. 변수 이름, 설명, 기본값, 그룹 및 소유자의 열 외에도, 계획 유형 열과 규칙 열이 표시됩니다.

이 옵션은 Planning의 글로벌 변수를 검색할 때는 사용할 수 없습니다. 글로벌 변수를 검색하면 글로벌 변수만 표시됩니다.

#### 6. 확인을 누릅니다.

검색하는 텍스트를 찾으면, 텍스트가 발견된 변수가 대체 또는 실행 탭에 나열됩니다.

텍스트를 찾은 다음 하나 이상의 텍스트 인스턴스를 바꿀 수 있습니다.

변수 디자이너에서 텍스트를 바꾸려면 다음 태스크 중 하나를 수행합니다.

- 텍스트 문자열의 선택한 인스턴스를 바꾸려면 다음을 수행합니다.
  1. 텍스트 문자열을 바꾸려는 변수를 선택합니다.
  2. **작업, 선택 내용 바꾸기** 순으로 선택합니다.
  3. **선택 내용 바꾸기** 대화상자의 **바꿀 내용**에 텍스트 문자열을 대체할 텍스트를 입력합니다.
  4. **선택 내용 바꾸기**를 누릅니다.
- 텍스트 문자열의 모든 인스턴스를 바꾸려면 다음을 수행합니다.
  1. **작업, 모두 바꾸기** 순으로 선택합니다.
  2. **모두 바꾸기** 대화상자의 **바꿀 내용**에 텍스트 문자열을 대체할 텍스트를 입력합니다.
  3. **모두 바꾸기**를 누릅니다.

#### 주:

텍스트 문자열을 검색하는 동안 선택한 옵션이 선택 내용 바꾸기 및 모두 바꾸기 대화상자의 찾기 영역에서 기본적으로 선택되며 변경할 수 없습니다. 예를 들어, 텍스트 문자열을 검색할 때 대소문자 무시 확인란을 선택 취소한 경우, 선택 내용 바꾸기/모두 바꾸기 대화상자에서 대소문자 무시 확인란이 선택 취소되며 변경할 수 없습니다.

## 변수 사용 표시

변수를 사용하는 비즈니스 규칙을 표시할 수 있습니다. 변수 사용을 표시하면 다음 정보가 나타납니다.

- 변수를 사용하는 비즈니스 규칙의 이름
- 변수를 사용하는 비즈니스 규칙의 애플리케이션 이름
- 변수를 사용하는 비즈니스 규칙의 계산 유형, 계획 유형 또는 데이터베이스
- 변수를 사용하는 비즈니스 규칙의 소유자
- 변수를 사용하는 비즈니스 규칙이 배포되었는지 여부
- 변수를 사용하는 비즈니스 규칙이 검증되었는지 여부
- 변수를 사용하는 비즈니스 규칙에 대한 설명

변수 사용을 표시하려면 다음을 수행합니다.

1. 시스템 뷰 또는 필터 뷰에서 **변수 디자이너** 아이콘을 누릅니다.
2. 변수 네비게이터에서 사용을 표시할 변수가 포함된 데이터베이스 객체를 선택합니다. 해당 객체에 대해 정의된 변수는 변수 디자이너의 **대체 및 실행** 탭에 표시됩니다.
3. 사용을 표시할 변수를 마우스 오른쪽 버튼으로 누르고 **사용 표시**를 선택합니다.



4. 정보를 검토한 후 **확인**을 누릅니다.

## 사전 정의된 시스템 변수를 Financial Management 애플리케이션으로 로드(Financial Management 사용자만 해당)

Oracle Hyperion Financial Management 시스템 템플릿으로 작업하려면 먼저 사전 정의된 시스템 변수를 시스템 템플릿을 사용할 애플리케이션으로 로드해야 합니다. 이러한 시스템 변수는 시스템 템플릿을 작동하게 하는 실행 변수와 대체 변수입니다.

### 주:

이러한 시스템 변수는 수정하거나 삭제하지 마십시오. 수정하거나 삭제할 경우 시스템 템플릿이 설계된 대로 작동하지 않습니다. 실수로 시스템 변수를 수정하거나 삭제하는 경우 시스템 변수를 애플리케이션으로 재로드할 수 있습니다.

사전 정의된 시스템 변수를 애플리케이션으로 로드한 후에 다른 애플리케이션으로 복사하고 애플리케이션 변수로 사용할 수 있습니다. 이러한 복사된 버전은 시스템 변수에 연결되어 있지 않기 때문에 수정하고 삭제할 수 있습니다.

사전 정의된 시스템 변수를 Financial Management 애플리케이션으로 로드하려면 다음을 수행합니다.

1. 시스템 뷰 또는 필터 뷰에서 **변수 디자이너** 아이콘을 누릅니다.
2. 변수 네비게이터에서 **Consolidation**을 확장하여 해당 애플리케이션을 표시합니다.
3. 시스템 템플릿을 사용하려는 애플리케이션을 마우스 오른쪽 버튼으로 누르고 **시스템 변수 로드**를 선택합니다.

시스템 변수가 애플리케이션에 성공적으로 로드되었다는 메시지가 표시됩니다. 사전 정의된 시스템 변수는 로드한 애플리케이션을 선택하면 [실행] 탭 및 [대체] 탭에 애플리케이션 변수와 함께 표시됩니다.

### 주:

시스템 템플릿을 사용하려는 각 애플리케이션에 대해 이 절차를 수행해야 합니다.

## 함수 작업

함수는 데이터 값이나 멤버를 반환하는 멤버 공식을 정의합니다.

예를 들어 함수(수치 연산자 및 논리 연산자)를 사용하여, 지정한 멤버의 동위 멤버, 상위 멤버 또는 1차 하위 구성요소 목록을 반환하거나, 지정한 값보다 크거나 작은 데이터 값 목록을 반환하거나, 지정한 멤버의 데이터 값을 할당할 수 있습니다. 함수를 선택하면 정확한 매개변수를 입력하라는 메시지가 표시됩니다.

Oracle Hyperion Financial Management, Oracle Hyperion Planning 및 Oracle Essbase 블록 저장영역 애플리케이션에서 작업하는 경우 공식, 스크립트, 조건 및 멤버 범위 구성요소에서 함수를 사용합니다. Oracle General Ledger에서 사용되는 항목을 포함하여

Essbase 집계 저장영역 애플리케이션에서 작업하는 경우에는 POV 구성요소에만 함수를 사용합니다.

계산에 사용할 수 있는 함수는 Financial Management, Planning, Oracle General Ledger 및 Essbase 애플리케이션에서 각각 다릅니다.

다음은 Financial Management 구성요소에서 사용할 수 있는 함수 유형 목록입니다. Financial Management 함수는 해당 함수를 사용할 수 있는 규칙 유형에 따라 분류됩니다. Financial Management 애플리케이션의 구성요소에서 사용할 수 있는 함수의 전체 목록 및 설명을 보려면 *Oracle Hyperion Financial Management Administrator's Guide*를 참조하십시오.

- 계산
- 환산
- 통합
- 할당
- 동적 계산
- 트랜잭션

다음은 Planning 및 Essbase 블록 저장영역 구성요소에서 사용할 수 있는 함수 유형 목록입니다. 함수의 전체 목록 및 설명을 보려면 [Oracle Essbase Technical Reference](#)를 참조하십시오.

- 부울
- 관계
- 계산 연산자
- 제어 플로우
- 데이터 선언
- 기능
- 수학
- 멤버 세트
- 범위(재무)
- 할당
- 예측
- 통계
- 날짜 및 시간
- 기타
- 사용자정의

#### 주:

함수는 멤버 선택기 및 함수 선택기에서 사용할 수 있습니다.

Oracle General Ledger에서 사용되는 항목을 포함하여 Essbase 합산 저장영역 애플리케이션 및 구성요소에서는 멤버 세트 함수만 사용합니다. 멤버 세트 함수의 전체 목록과 설명을 보려면 [Oracle Essbase Technical Reference](#)를 참조하십시오.

## Calculation Manager에서 지원되는 Essbase 함수

다음은 블록 저장영역 애플리케이션의 Oracle Hyperion Calculation Manager에서 지원되는 Oracle Essbase 함수입니다.

**표 8-7 Calculation Manager에서 지원되는 Essbase 함수**

@ABS	@ISANCEST	@MOVSUMX
@ACCUM	@ISATTRIBUTE	@NAME
@ALLANCESTORS	@ISCHILD	@NEXT
@ALIAS	@ISDESC	@NEXTS
@ALLOCATE	@ISGEN	@NEXTSIBLING
@ANCEST	@ISIANCEST	@NOTEQUAL
@ANCESTORS	@ISIBLINGS	@NPV
@ANCESTVAL	@ISICHILD	@PARENT
@ATTRIBUTE	@ISIDESC	@PARENTVAL
@ATTRIBUTEVAL	@ISIPARENT	@POWER
@ATTRIBUTESVAL	@ISISIBLING	@PREVSIBLING
@ATTRIBUTEVAL	@ISLEV	@PRIOR
@AVG	@ISMBR	@PRIORS
@AVGRANGE	@ISMBRUDA	@PTD
@BETWEEN	@ISMBRWITHATTR	@RANGE
@CALCMODE	@ISPARENT	@RANGEFIRSTVAL
@CHILDREN	@ISRANGENONEMPTY	@RANGELASTVAL
@COMPOUND	@ISSAMEGEN	@RANK
@COMPOUNDGROWTH	@ISSAMELEV	@RDESCENDANTS
@CONCATENATE	@ISSIBLING	@RELATIVE
@CORRELATION	@ISUDA	@RELXRANGE
@COUNT	@LANCESTORS	@REMAINDER
@CREATEBLOCK	@LDESCENDANTS	@REMOVE
@CURGEN	@LEV	@RETURN
@CURLEV	@LEVMBRS	@ROUND
@CURRMBR	@LIKE	@RSIBLINGS
@CURRMBRRANGE	@LIST	@SANCESTVAL
@DATEDIFF	@LN	@SHARE
@DATEPART	@LOG	@SHIFT
@DATEROLL	@LOG10	@SHIFTMINUS
@DECLINE	@LSIBLINGS	@SHIFTPLUS
@DESCENDANTS	@MATCH	@SHIFTSIBLING
@DISCOUNT	@MAX	@SIBLINGS
@ENUMVALUE	@MAXRANGE	@SLN
@EQUAL	@MAXS	@SPARENTVAL
@EXP	@MAXSRANGE	@SPLINE
@EXPAND	@MBRCOMPARE	@STDEV

표 8-7 (계속) Calculation Manager에서 지원되는 Essbase 함수

@FACTORIAL	@MBRPARENT	@STDEVP
@FORMATDATE	@MDALLOCATE	@STDEV RANGE
@GEN	@MDANCESTVAL	@SUBSTRING
@GENMBRS	@MDPARENTVAL	@SUM
@GROWTH	@MDSHIFT	@SUMRANGE
@IALLANCESTORS	@MEDIAN	@SYD
@IANCESTORS	@MEMBER	@TODATE
@ICHILDREN	@MEMBERAT	@TODATEEX
@IDESCENDANTS	@MERGE	@TODAY
@IALLANCESTORS	@MIN	@TREND
@ILDESCENDANTS	@MINRANGE	@TRUNCATE
@ILSIBLINGS	@MINS	@UDA
@INT	@MINSRANGE	@VAR
@INTEREST	@MOD	@VARPER
@INTERSECT	@MODE	@VARIANCE
@IRDESCENDANTS	@MOVAVG	@VARIANCEP
@IRR	@MOVMAX	@WITHATTR
@IRREX	@MOV MED	@XRANGE
@IRSIBLINGS	@MOV MIN	@XREF
@ISACCTYPE	@MOV SUM	@XWRITE

## 사용자정의 함수 작업

사용자정의 함수를 사용하여 데이터 복사 및 익스포트, 텍스트 문자열에 작은따옴표 또는 큰따옴표 추가 및 제거, 두 텍스트 문자열 비교, 다른 형식으로 날짜 변환과 같은 태스크를 수행합니다.

### 참조:

- 사용자정의 함수 정보
- 연도 매개변수가 포함된 사용자정의 함수 사용
- 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수
- 비트 단위 함수
- 카운터 함수
- 날짜/시간 함수
- 재무 함수
- 로그 함수
- 수리 함수
- MaxL 함수
- MDX 함수
- 통계 함수

- 문자열 함수

## 사용자정의 함수 정보

사용자정의 함수는 Oracle Hyperion Calculation Manager 조건, 스크립트 및 공식 구성요소의 함수 선택기에서 접근합니다.

사용자정의 함수를 사용하면 암호화된 MaxL 스크립트 및 파일 실행, 데이터 복사 및 익스포트, 텍스트 문자열에 작은따옴표 또는 큰따옴표 추가 및 제거, 두 문자열 비교, YYYYMMDD 형식으로 날짜 변환 등의 태스크를 수행할 수 있습니다.

Oracle Essbase 및 Oracle Hyperion Planning 비즈니스 규칙 구성요소에서 사용자정의 함수를 사용할 수 있습니다.

## 연도 매개변수가 포함된 사용자정의 함수 사용

일부 사용자정의 함수에서는 매개변수에 *year* 선택 항목이 포함된 드롭다운이 있을 수도 있습니다. 애플리케이션에 *year*라는 차원 또는 멤버가 있는 경우 *year* 선택 항목이 있는 사용자정의 함수 사용이 검증되지 않습니다. *year*뿐만 아니라 차원 또는 멤버 이름이 동일한 매개변수 선택 항목의 경우 이슈가 발생할 수 있습니다.

이 이슈를 해결하려면 매개변수 드롭다운에서 *year*를 선택한 후 규칙에서 @name(*year*)와 같이 표시되도록 @name( )를 추가합니다.

## 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 다음 함수를 사용하여 YYYYMMDD 형식(직렬 형식)의 날짜를 Excel 형식으로 변환할 수 있습니다.

- @CalcMgrExcelDATE는 특정 날짜의 일련 번호를 반환합니다.

다음 예는 20181214(YYYYMMDD 형식)를 Excel 날짜로 변환합니다.

```
@CalcMgrExcelDATE(20181214)
```

- @CalcMgrDateToExcel은 YYYYMMDD 형식의 단일 날짜를 Excel 날짜로 변환합니다.
- @CalcMgrDatesToExcel은 YYYYMMDD 형식의 여러 날짜를 Excel 날짜로 변환합니다.

다음 예를 참조하십시오.

```
@CalcMgrDatesToExcel(@LIST("Jan"->"Date_123", "Feb"->"Date_123"))
```

"Jan"->"Date\_123"은 01/31/19로 표시되고 "Feb"->"Date\_123"은 02/31/19로 표시됩니다.

## 비트 단위 함수

참조:

- @CalcMgrBitAnd
- @CalcMgrBitOR
- @CalcMgrBitExOR
- @CalcMgrBitExBoolOR
- @CalcMgrBitCompliment

- [@CalcMgrBitShiftLeft](#)
- [@CalcMgrBitShiftRight](#)
- [@CalcMgrBitUnsignedShiftRight](#)

## @CalcMgrBitAnd

### 목적:

첫 번째 피연산자의 각 비트를 두 번째 피연산자의 해당 비트와 비교하는 비트 단위 AND 연산을 수행합니다. 두 비트 모두 1인 경우 해당 결과 비트가 1로 설정되고 그렇지 않은 경우 해당 결과 비트가 0으로 설정됩니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.AND(double,double)
```

CDF Spec: @CalcMgrBitAnd(number1,number2)

## @CalcMgrBitOR

### 목적:

첫 번째 피연산자의 각 비트를 두 번째 피연산자의 해당 비트와 비교하는 비트 단위 OR 연산을 수행합니다. 두 비트 중 하나가 1인 경우 해당 결과 비트가 1로 설정되고 그렇지 않은 경우 해당 결과 비트가 0으로 설정됩니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.OR(double,double)
```

CDF 사양: @CalcMgrBitOR(number1,number2)

## @CalcMgrBitExOR

### 목적:

첫 번째 피연산자의 각 비트를 두 번째 피연산자의 해당 비트와 비교하는 배타적 비트 단위 OR 연산을 수행합니다. 두 비트 중 하나가 1인 경우 해당 결과 비트가 1로 설정되고 그렇지 않은 경우 해당 결과 비트가 0으로 설정됩니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.EXCLUSIVEOR(double,double)
```

CDF 사양: @CalcMgrBitExOR(number1,number2)

## @CalcMgrBitExBoolOR

### 목적:

첫 번째 피연산자의 각 비트를 두 번째 피연산자의 해당 비트와 비교하는 배타적 비트 단위 OR 연산을 수행합니다. 두 비트 중 하나가 1인 경우 해당 결과 비트가 1로 설정되고 그렇지 않은 경우 해당 결과 비트가 0으로 설정됩니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.EXCLUSIVEOR(double, double)`  
CDF 사양: `@CalcMgrBitExOR(number1, number2)`

## @CalcMgrBitCompliment

**목적:**

각 비트를 반대로 설정하는 단항 비트 단위 보수를 수행합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.UNARYCOMPLIMENT(double)`  
CDF 사양: `@CalcMgrBitCompliment(number1)`

## @CalcMgrBitShiftLeft

**목적:**

서명된 왼쪽 이동을 수행합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.SIGNEDSHIFTLLEFT(double, double)`  
CDF 사양: `@CalcMgrBitShiftLeft(number1, number2)`

## @CalcMgrBitShiftRight

**목적:**

서명된 오른쪽 이동을 수행합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.SIGNEDSHIFTRIGHT(double, double)`  
CDF 사양: `@CalcMgrBitShiftRight(number1, number2)`

## @CalcMgrBitUnsignedShiftRight

**목적:**

서명되지 않은 오른쪽 이동을 수행합니다.

**구문:**

`com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.UNSIGNEDSHIFTRIGHT(double, double)`  
`@CalcMgrBitUnsignedShiftRight(number1, number2)`

## 카운터 함수

### 참조:

- [@CalcMgrCounterAddNumber](#)
- [@CalcMgrCounterAddText](#)
- [@CalcMgrCounterClear](#)
- [@CalcMgrCounterClearAll](#)
- [@CalcMgrCounterClearKey](#)
- [@CalcMgrCounterDecrement](#)
- [@CalcMgrCounterDecrementKey](#)
- [@CalcMgrCounterGetKeyNumber](#)
- [@CalcMgrCounterGetKeyText](#)
- [@CalcMgrCounterGetNumber](#)
- [@CalcMgrCounterGetText](#)
- [@CalcMgrCounterIncrement](#)
- [@CalcMgrCounterIncrementKey](#)
- [@CalcMgrCounterUpdate](#)
- [@CalcMgrCounterUpdateNumber](#)
- [@CalcMgrCounterUpdateNumberText](#)
- [@CalcMgrCounterUpdateText](#)

### @CalcMgrCounterAddNumber

#### 목적:

카운터에 숫자를 추가하고 키를 반환합니다.

#### 구문:

Java 클래스: `com.hyperion.calcmgr.common.cdf.CounterFunctions.addNumber(double)`

CDF 사양: `@CalcMgrCounterAddNumber(number)`

### @CalcMgrCounterAddText

#### 목적:

카운터에 텍스트 문자열을 추가하고 키를 반환합니다.

#### 구문:

Java 클래스: `com.hyperion.calcmgr.common.cdf.CounterFunctions.addText(String)`

CDF 사양: `@CalcMgrCounterAddText(text)`



## @CalcMgrCounterClear

**목적:**

키를 통해 지정된 카운터를 지웁니다.

**구문:**

Java 클래스: `com.hyperion.calcmgr.common.cdf.CounterFunctions.clear(double)`

CDF 사양: `@CalcMgrCounterClear(key)`

## @CalcMgrCounterClearAll

**목적:**

카운터에서 모든 키와 값을 제거합니다.

**구문:**

Java Class: `com.hyperion.calcmgr.common.cdf.CounterFunctions.clearAll()`

CDF Spec: `@CalcMgrCounterClearAll()`

## @CalcMgrCounterClearKey

**목적:**

키와 연관된 카운터에서 값을 제거합니다.

**구문:**

Java 클래스:

`com.hyperion.calcmgr.common.cdf.CounterFunctions.clearKey(String)`

CDF 사양: `@CalcMgrCounterClearKey(key)`

## @CalcMgrCounterDecrement

**목적:**

키에 따라 카운터에서 값을 줄입니다. 키가 없는 경우 키에 대해 0 값이 설정됩니다.

**구문:**

Java 클래스:

`com.hyperion.calcmgr.common.cdf.CounterFunctions.decrement(double)`

CDF 사양: `@CalcMgrCounterDecrement(key)`

## @CalcMgrCounterDecrementKey

**목적:**

키에 따라 카운터에서 값을 줄입니다. 키가 없는 경우 키에 대해 0 값이 설정됩니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.CounterFunctions.decrementKey(String)`  
CDF 사양: `@CalcMgrCounterDecrementKey(key)`

## @CalcMgrCounterGetKeyNumber

**목적:**  
키에 따라 카운터에서 찾은 텍스트를 반환합니다. 키가 없는 경우 누락된 값이 반환됩니다.  
**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.CounterFunctions.getKeyNumber(String, double)`  
CDF 사양: `@CalcMgrCounterGetKeyNumber(key, missing_value)`

## @CalcMgrCounterGetKeyText

**목적:**  
키에 따라 카운터에서 찾은 텍스트를 반환합니다. 키가 없는 경우 누락된 값이 반환됩니다.  
**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.CounterFunctions.getKeyText(String, String)`  
CDF 사양: `@CalcMgrCounterGetKeyText(key, missing_value)`

## @CalcMgrCounterGetNumber

**목적:**  
키를 통해 지정된 카운터에서 숫자를 반환합니다. 키가 없거나 값이 숫자가 아닌 경우 누락된 값이 반환됩니다.  
**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.CounterFunctions.getNumber(double, double)`  
CDF 사양: `@CalcMgrCounterGetNumber(key, missingValue)`

## @CalcMgrCounterGetText

**목적:**  
키에 따라 카운터에서 찾은 텍스트를 반환합니다. 키가 없는 경우 누락된 값이 반환됩니다.  
**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.CounterFunctions.getText(double, String)`  
CDF 사양: `@CalcMgrCounterGetText(key, missing_value)`

## @CalcMgrCounterIncrement

**목적:**

키를 통해 지정된 카운터에서 값을 늘립니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CounterFunctions.increment(double)
```

CDF 사양: @CalcMgrCounterIncrement(key)

## @CalcMgrCounterIncrementKey

**목적:**

키에 따라 카운터에서 값을 늘립니다. 키가 없는 경우 키에 대해 0 값이 설정됩니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CounterFunctions.incrementKey(String)
```

CDF 사양: @CalcMgrCounterIncrementKey(key)

## @CalcMgrCounterUpdate

**목적:**

키가 지정된 카운터에서 숫자를 설정합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CounterFunctions.update(double, double)
```

CDF 사양: @CalcMgrCounterUpdate(key, number)

## @CalcMgrCounterUpdateNumber

**목적:**

키가 지정된 카운터에서 숫자를 업데이트합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CounterFunctions.updateNumber(String, double)
```

CDF 사양: @CalcMgrCounterUpdateNumber(key, number)

## @CalcMgrCounterUpdateNumberText

**목적:**

키가 지정된 카운터에서 숫자를 업데이트합니다.

**구문:**

Java 클래스:

`com.hyperion.calcmgr.common.cdf.CounterFunctions.updateNumberText(double, String)`CDF 사양: `@CalcMgrCounterUpdateNumberText(key, number)`

## @CalcMgrCounterUpdateText

**목적:**

키가 지정된 카운터에서 텍스트를 업데이트합니다.

**구문:**

Java 클래스:

`com.hyperion.calcmgr.common.cdf.CounterFunctions.updateText(String, String)`CDF 사양: `@CalcMgrCounterUpdateText(key, text)`

## 날짜/시간 함수

**참조:**

- [@CalcMgrAddDate](#)
- [@CalcMgrAddDatePart](#)
- [@CalcMgrAddDays](#)
- [@CalcMgrAddMonths](#)
- [@CalcMgrAddWeeks](#)
- [@CalcMgrAddYears](#)
- [@CalcMgrDateDiff](#)
- [@CalcMgrDateToExcel](#)
- [@CalcMgrDatesToExcel](#)
- [@CalcMgrDateTimeToExcel](#)
- [@CalcMgrDateTimesToExcel](#)
- [@CalcMgrDateToString](#)
- [@CalcMgrDaysBetween](#)
- [@CalcMgrDaysDiff](#)
- [@CalcMgrDiffDate](#)
- [@CalcMgrExcelADD](#)
- [@CalcMgrExcelDATE](#)
- [@CalcMgrExcelDATEDIF](#)
- [@CalcMgrExcelDAYOFYEAR](#)
- [@CalcMgrExcelDAYS360](#)
- [@CalcMgrExcelDAYSINMONTH](#)
- [@CalcMgrExcelEOMONTH](#)

- @CalcMgrExcelHOUR
- @CalcMgrExcelMINUTE
- @CalcMgrExcelMONTH
- @CalcMgrExcelNETWORKDAYS
- @CalcMgrExcelSECOND
- @CalcMgrExcelToDate
- @CalcMgrExcelToDateTime
- @CalcMgrExcelWEEKNUM
- @CalcMgrExcelWEEKDAY
- @CalcMgrExcelWORKDAY
- @CalcMgrExcelYEAR
- @CalcMgrExcelYEARFRAC
- @CalcMgrGetCurrentDate
- @CalcMgrGetCurrentDateTZ
- @CalcMgrGetCurrentDateTime
- @CalcMgrGetCurrentDateTimeTZ
- @CalcMgrGetCustomDate
- @CalcMgrGetCustomDateTime
- @CalcMgrGetDatePart
- @CalcMgrGetDateTimePart
- @CalcMgrGetDay
- @CalcMgrGetDayOfYear
- @CalcMgrGetFormattedDate
- @CalcMgrGetMaxDaysInMonth
- @CalcMgrGetMonth
- @CalcMgrGetStringFormattedDateTime
- @CalcMgrGetWeekOfMonth
- @CalcMgrGetWeekOfYear
- @CalcMgrGetYear
- @CalcMgrIsLeapYear
- @CalcMgrMonthsBetween
- @CalcMgrMonthsDiff
- @CalcMgrRollDate
- @CalcMgrRollDay
- @CalcMgrRollMonth
- @CalcMgrRollYear
- @CalcMgrWeeksBetween

- [@CalcMgrWeeksDiff](#)
- [@CalcMgrYearsBetween](#)
- [@CalcMgrYearsDiff](#)

## @CalcMgrAddDate

### 목적:

YYYYMMDD 형식인 날짜에 지정한 기간(년), 기간(월) 및 기간(일)을 더합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.addDate(int,int,int,int)
```

CDF 사양: @CalcMgrAddDate(date, years, months, days)

## @CalcMgrAddDatePart

### 목적:

YYYYMMDD 형식인 날짜에 지정한 기간(년)/기간(월)/기간(일)/기간(주)을 더합니다. "date\_part"는 "day", "month", "week", "year" 중 하나일 수 있습니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.addDatePart(int,String,int)
```

CDF 사양: @CalcMgrAddDatePart(date,date\_part, amountToAdd)

## @CalcMgrAddDays

### 목적:

YYYYMMDD 형식인 날짜에 지정한 기간(일)을 더합니다.

### 구문:

Java 클래스: com.hyperion.calcmgr.common.cdf.CalendarFunctions.addDays(int,int)

CDF 사양: @CalcMgrAddDays(date, daysToAdd)

## @CalcMgrAddMonths

### 목적:

YYYYMMDD 형식인 날짜에 지정한 기간(월)을 더합니다.

### 구문:

Java 클래스: com.hyperion.calcmgr.common.cdf.CalendarFunctions.addMonths(int,int)

CDF 사양: @CalcMgrAddMonths(date, monthsToAdd)

## @CalcMgrAddWeeks

**목적:**

YYYYMMDD 형식인 날짜에 지정한 기간(주)을 더합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.addWeeks(int,int)
```

CDF 사양: @CalcMgrAddWeeks (date, weeksToAdd)

## @CalcMgrAddYears

**목적:**

YYYYMMDD 형식인 날짜에 지정한 기간(년)을 더합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.addYears(int,int)
```

CDF 사양: @CalcMgrAddYears (date, yearsToAdd)

## @CalcMgrDateDiff

**목적:**

표준 양력 달력에 따라 YYYYMMDD 형식인 두 입력 날짜에서 지정한 날짜 부분의 차이(숫자)를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.dateDiff(int,int,String)
```

CDF 사양: @CalcMgrDateDiff (fromDate,toDate,datePart)

 **주:**

@CalcMgrDateDiff는 양수만 반환합니다. 음수를 반환하려면(적용되는 경우) @CalcMgrDiffDate를 사용합니다.

## @CalcMgrDateToExcel

**목적:**

YYYYMMDD 형식의 단일 날짜를 Excel 날짜로 변환합니다.

**구문:**

Java 클래스:  
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DATE\_TOEXCEL(double)  
CDF 사양: @CalcMgrDateToExcel(date)

## @CalcMgrDatesToExcel

### 목적:

YYYYMMDD 형식의 여러 날짜를 Excel 날짜로 변환합니다.

### 구문:

Java 클래스:  
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DATES\_TOEXCEL(double[])  
CDF 사양: @CalcMgrDatesToExcel(dates)

## @CalcMgrDateTimeToExcel

### 목적:

YYYYMMDDHHMMSS 형식의 단일 날짜를 Excel 날짜로 변환합니다.

### 구문:

Java 클래스:  
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DATETIME\_TOEXCEL(double)  
CDF 사양: @CalcMgrDateTimeToExcel(date)

## @CalcMgrDateTimesToExcel

### 목적:

YYYYMMDDHHMMSS 형식의 여러 날짜를 Excel 날짜로 변환합니다.

### 구문:

Java 클래스:  
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DATETIMES\_TOEXCEL(double[])  
CDF 사양: CalcMgrDateTimesToExcel(dates)

## @CalcMgrDateToString

### 목적:

YYYYMMDD 형식의 날짜를 지정된 형식을 사용하는 문자열로 반환합니다. 형식은 Java 설명서의 SimpleDateFormat을 참조하십시오.

### 구문:

Java 클래스:  
com.hyperion.calcmgr.common.cdf.CalendarFunctions.dateToString(int, String)  
CDF 사양: @CalcMgrDateToString(date, format)



## @CalcMgrDaysBetween

### 목적:

YYYYMMDD 형식인 두 날짜 사이의 일 수를 반환합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.daysBetween(int, int)
```

CDF 사양: @CalcMgrDaysBetween(fromDate, toDate)

### 주:

@CalcMgrDaysBetween은 양수만 반환합니다. 음수를 반환하려면(적용되는 경우) [@CalcMgrDaysDiff](#)를 사용합니다.

## @CalcMgrDaysDiff

### 목적:

YYYYMMDD 형식인 두 날짜 사이의 일 수를 반환합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.daysDiff(int, int)
```

CDF 사양: @CalcMgrDaysDiff(fromDate, toDate)

### 주:

@CalcMgrDaysDiff를 사용하는 경우 함수의 첫번째 날짜가 두번째 날짜보다 이후이면 음수가 반환됩니다. 함수의 첫번째 날짜가 두번째 날짜보다 이전이면 양수가 반환됩니다. 양수만 반환하려는 경우 [@CalcMgrDaysBetween](#)을 사용합니다.

## @CalcMgrDiffDate

### 목적:

표준 양력 달력에 따라 YYYYMMDD 형식인 두 입력 날짜에서 지정한 날짜 부분의 차이(숫자)를 반환합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.diffDate(int, int, String)
```

CDF 사양: @CalcMgrDiffDate(fromDate, toDate, datePart)

 주:

@CalcMgrDiffDate를 사용하는 경우 함수의 첫번째 날짜가 두번째 날짜보다 이후이면 음수가 반환됩니다. 함수의 첫번째 날짜가 두번째 날짜보다 이전이면 양수가 반환됩니다. 양수만 반환하려는 경우 @CalcMgrDateDiff를 사용합니다.

## @CalcMgrExcelADD

**목적:**

금액 누계를 추가합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.ADD(double, double, String)
```

CDF 사양: @CalcMgrExcelADD(date, amount, what)

## @CalcMgrExcelDATE

**목적:**

특정 날짜의 일련 번호를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DATE(double, double, double)
```

CDF 사양: @CalcMgrExcelDATE(year, month, day)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelDATEDIF

**목적:**

두 날짜 사이의 기간(일), 기간(월) 또는 기간(년)을 계산합니다.

기간을 계산해야 하는 공식에 유용합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DATEDIF(double, double, String)
```

CDF 사양: @CalcMgrExcelDATEDIF(start\_date, end\_date, unit)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelDAYOFYEAR

**목적:**

일련 번호를 연도의 일로 변환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DAYOFYEAR(double)
```

CDF 사양: @CalcMgrExcelDAYOFYEAR(Date)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelDAYS360

**목적:**

1년 360일 기준으로 두 날짜 사이의 기간(일)을 계산합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DAYS360(double, double, boolean)
```

CDF 사양: @CalcMgrExcelDAYS360(start\_date, end\_date, method)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelDAYSINMONTH

**목적:**

일련 번호를 월의 일로 변환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DAYSINMONTH(double)`

CDF 사양: `@CalcMgrExcelDAYSINMONTH(date)`

 주:

`@CalcMgrExcel` 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 `@CalcMgrExcel` 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelEOMONTH

**목적:**

지정된 기간(월) 이전 또는 이후 월 마지막 날의 일련 번호를 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.EOMONTH(double, double)`

CDF 사양: `@CalcMgrExcelEOMONTH(dateValue, adjustmentMonths)`

 주:

`@CalcMgrExcel` 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 `@CalcMgrExcel` 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelHOUR

**목적:**

일련 번호를 시로 변환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.HOUR(double)`

CDF 사양: `@CalcMgrExcelHOUR(date)`

## @CalcMgrExcelMINUTE

**목적:**

일련 번호를 분으로 변환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.MINUTE(double)`

CDF 사양: @CalcMgrExcelMINUTE (date)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelMONTH

**목적:**

일련 번호를 월로 변환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.MONTH(double)
```

CDF 사양: @CalcMgrExcelMONTH (Date)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelNETWORKDAYS

**목적:**

두 날짜 사이의 전체 근무일 수를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.NETWORKDAYS(double, double, double)
```

CDF 사양: @CalcMgrExcelNETWORKDAYS (startDate, endDate, holidays)

 주:

Oracle Hyperion Planning 날짜 형식으로 날짜를 전달하는 경우  
[@CalcMgrDateToExcel](#)을 사용하여 Planning 날짜를 Excel 날짜로 변환해야  
합니다.

**주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelSECOND

**목적:**

일련 번호를 초로 변환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.SECOND(double)
```

CDF 사양: @CalcMgrExcelSECOND(date)

**주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelToDate

**목적:**

Excel 날짜를 YYYYMMDD 형식으로 변환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.EXCEL_TODATE(double)
```

CDF 사양: @CalcMgrExcelToDate(excel\_date)

## @CalcMgrExcelToDateTime

**목적:**

Excel 날짜를 YYYYMMDDHHMMSS 형식으로 변환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.EXCEL_TODATETIME(double)
```

CDF 사양: @CalcMgrExcelToDateTime(excel\_date)

## @CalcMgrExcelWEEKNUM

**목적:**

특정 날짜의 주 수를 반환합니다. 예를 들어 1월 1일이 포함된 주가 해당 연도의 첫 번째 주이며 1주로 번호가 지정됩니다.

#### 구문:

Java 클래스:  
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.WEEKNUM(double, double)

CDF 사양: @CalcMgrExcelWEEKNUM(date, method)

#### 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelWEEKDAY

#### 목적:

날짜에 해당하는 요일을 반환합니다. 요일은 기본적으로 1(일요일)에서 7(토요일) 사이의 정수로 제공됩니다.

#### 구문:

Java 클래스:  
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.WEEKDAY(double, double)

CDF 사양: @CalcMgrExcelWEEKDAY(serial\_number, return\_type)

#### 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelWORKDAY

#### 목적:

지정된 근무일 수 이전 또는 이후 날짜의 일련 번호를 반환합니다.

#### 구문:

Java 클래스:  
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.WORKDAY(double, double, double[])

CDF 사양: @CalcMgrExcelWORKDAY(startDate, days, holidays)

**주:**

Oracle Hyperion Planning 날짜 형식으로 날짜를 전달하는 경우  
@CalcMgrDateToExcel을 사용하여 Planning 날짜를 Excel 날짜로 변환해야 합니다.

**주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜  
매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelYEAR

**목적:**

일련 번호를 연도로 변환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.YEAR(double)
```

CDF 사양: @CalcMgrExcelYEAR(date)

**주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜  
매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelYEARFRAC

**목적:**

start\_date 및 end\_date 사이의 전체 기간(일)을 나타내는 연도 분수를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.YEARFRAC(double, double, double)
```

CDF 사양: @CalcMgrExcelYEARFRAC(startDate, endDate, basis)

**주:**

Oracle Hyperion Planning 날짜 형식으로 날짜를 전달하는 경우  
@CalcMgrDateToExcel을 사용하여 Planning 날짜를 Excel 날짜로 변환해야 합니다.



**주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrGetCurrentDate

**목적:**

현재 날짜를 YYYYMMDD 형식으로 반환합니다(예: 20140101).

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getCurrentDate(int)
```

CDF Spec: @CalcMgrGetCurrentDate()

## @CalcMgrGetCurrentDateTZ

**목적:**

YYYYMMDD 형식으로 제공된 시간대의 현재 날짜를 반환합니다(예: 20140101).

**구문:**

Java Class:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getCurrentDate(String)
```

CDF Spec: @CalcMgrGetCurrentDateTZ(timeZone)

## @CalcMgrGetCurrentDateTime

**목적:**

현재 날짜와 시간을 YYYYMMDDHHMMSS 형식으로 반환합니다. 예:  
20140101143001(Year\_Month\_Day\_Hour\_Minute\_Second)

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getCurrentDateTime()
```

CDF Spec: @CalcMgrGetCurrentDateTime()

## @CalcMgrGetCurrentDateTimeTZ

**목적:**

YYYYMMDDHHMMSS 형식으로 제공된 시간대의 현재 날짜 및 시간을 반환합니다. 예:  
20140101143001(Year\_Month\_Day\_Hour\_Minute\_Second)

**구문:**

Java Class:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getCurrentDateTime(String)
```

CDF 사양: @CalcMgrGetCurrentDateTimeTZ(timeZone)

## @CalcMgrGetCustomDate

### 목적:

사용자정의 날짜를 YYYYMMDD 형식으로 반환합니다.

예: 20140101

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getDate(double, double, double)
```

CDF 사양: @CalcMgrGetCustomDate(year, month, day)

## @CalcMgrGetCustomDateTime

### 목적:

YYYYMMDDHHMMSS 형식의 사용자정의 일자와 시간을 반환합니다. 예: 20140101143001  
(Year\_Month\_Day\_Hour\_Minute\_Second)

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getDateTime(double, double, double, double, double, double)
```

CDF 사양: @CalcMgrGetCustomDateTime(year, month, day, hour, min, sec)

## @CalcMgrGetDatePart

### 목적:

Year/Month/DayOfMonth/WeekOfYear/WeekOfMonth/DayOfYear를 YYYYMMDD 형식인 날짜의 숫자로 반환합니다. "date\_part\_ex"는 "dayofmonth", "dayofyear", "month", "weekofmonth", "weekofyear" 또는 "year"일 수 있습니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.datePart(Double, String)
```

CDF 사양: @CalcMgrGetDatePart(date, date\_part\_ex)

## @CalcMgrGetDateTimePart

### 목적:

Year/Month/DayOfMonth/WeekOfYear/WeekOfMonth/DayOfYear/Hour/Minute/Seconds를 날짜의 숫자로 반환합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.dateTimePart(double, String)
```

CDF 사양: @CalcMgrGetDateTimePart(date, date\_part\_ex)

## @CalcMgrGetDay

**목적:**

YYYYMMDD 형식인 날짜의 일을 반환합니다.

**구문:**

Java 클래스: `com.hyperion.calcmgr.common.cdf.CalendarFunctions.getDay(int)`

CDF 사양: `@CalcMgrGetDay(date)`

## @CalcMgrGetDayOfYear

**목적:**

YYYYMMDD 형식인 날짜의 일(1-366)을 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.CalendarFunctions.getDayOfYear(int)`

CDF 사양: `@CalcMgrGetDayOfYear(date)`

## @CalcMgrGetFormattedDate

**목적:**

날짜를 YYYYMMDD 형식으로 변환합니다. 예를 들어  
`@CalcMgrGetFormattedDate(12302014, "mmdyyyyy")`는 20141230을 반환합니다

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.CalendarFunctions.getFormattedDate(int, String)`

CDF 사양: `@CalcMgrGetFormattedDate(date, format)`

## @CalcMgrGetMaxDaysInMonth

**목적:**

YYYYMMDD 형식인 날짜 월의 최대 일 수를 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.CalendarFunctions.getActualMaximumDays(int)`

CDF 사양: `@CalcMgrGetMaxDaysInMonth(date)`

## @CalcMgrGetMonth

**목적:**

YYYYMMDD 형식인 날짜의 월을 반환합니다.

**구문:**

Java 클래스: `com.hyperion.calcmgr.common.cdf.CalendarFunctions.getMonth(int)`

CDF 사양: `@CalcMgrGetMonth(date)`

## @CalcMgrGetStringFormattedDateTime

**목적:**

형식에 따라 정의된 날짜를 `YYYYMMddHHmmss` 형식의 날짜로 변환합니다.

예를 들어 `@CalcMgrGetFormattedDate(12302014, "MMddyyyyHHmmss")`는 `201412301430`을 반환합니다.

자세한 내용은 Java 설명서의 "SimpleDateFormat"을 참조하십시오.

가능한 형식 값: `mmddyyyyHHmmss`, `ddmmyyyyHHmmss`, `yyyymmddHHmmss`

**구문:**

Java 클래스:

`com.hyperion.calcmgr.common.cdf.CalendarFunctions.getStringFormattedDateTime(String, String)`

CDF 사양: `@CalcMgrGetStringFormattedDateTime(date, format)`

## @CalcMgrGetWeekOfMonth

`YYYYMMDD` 형식인 날짜의 주(월)를 반환합니다.

**구문:**

Java 클래스:

`com.hyperion.calcmgr.common.cdf.CalendarFunctions.getWeekOfMonth(int)`

CDF 사양: `@CalcMgrGetWeekOfMonth(date)`

## @CalcMgrGetWeekOfYear

**목적:**

`YYYYMMDD` 형식인 날짜의 연도의 주를 반환합니다.

**구문:**

Java 클래스: `com.hyperion.calcmgr.common.cdf.CalendarFunctions.getWeekOfYear(int)`

CDF 사양: `@CalcMgrGetWeekOfYear(date)`

## @CalcMgrGetYear

**목적:**

`YYYYMMDD` 형식인 날짜의 연도를 반환합니다. "date\_part\_ex"는 "dayofmonth", "dayofyear", "month", "weekofmonth", "weekofyear" 또는 "year"여야 합니다.

**구문:**

Java 클래스: `com.hyperion.calcmgr.common.cdf.CalendarFunctions.getYear(int)`

CDF 사양: @CalcMgrGetYear (date)

## @CalcMgrIsLeapYear

### 목적:

지정된 날짜가 윤년인지 여부를 확인합니다. 날짜는 YYYYMMDD 또는 YYYY 형식이어야 합니다 (예: 20140101 또는 2014).

### 구문:

Java 클래스:  
com.hyperion.calcmgr.common.cdf.CalendarFunctions.isLeapYear (int)  
CDF 사양: @CalcMgrIsLeapYear (date)

## @CalcMgrMonthsBetween

### 목적:

YYYYMMDD 형식인 두 날짜 사이의 월을 반환합니다.

### 구문:

Java 클래스:  
com.hyperion.calcmgr.common.cdf.CalendarFunctions.monthsBetween (int, int)  
CDF 사양: @CalcMgrMonthsBetween (fromDate, toDate)

### 주:

@CalcMgrMonthsBetween은 양수만 반환합니다. 음수를 반환하려면(적용되는 경우) @CalcMgrMonthsDiff를 사용합니다.

## @CalcMgrMonthsDiff

### 목적:

YYYYMMDD 형식인 두 날짜 사이의 월을 반환합니다.

### 구문:

Java 클래스:  
com.hyperion.calcmgr.common.cdf.CalendarFunctions.monthsDiff (int, int)  
CDF 사양: @CalcMgrMonthsDiff (fromDate, toDate)

### 주:

@CalcMonthsDiff를 사용하는 경우 함수의 첫번째 날짜가 두번째 날짜보다 이후이면 음수가 반환됩니다. 함수의 첫번째 날짜가 두번째 날짜보다 이전이면 양수가 반환됩니다. 양수만 반환하려는 경우 @CalcMgrMonthsBetween를 사용합니다.

## @CalcMgrRollDate

**목적:**

더 많은 필드를 변경하지 않고 지정된 날짜 필드에서 단일 시간 단위를 더하거나 뺍니다(위쪽 또는 아래쪽).

예를 들어 @CalcMgrRollDate(19960131,"month",@\_true)는 날짜 19960229를 생성합니다.  
@CalcMgrRollDate(19960131,"day",@\_true)는 날짜 19960101을 생성합니다.

date\_part에 사용 가능한 값은 day, month, week 및 year입니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.rollDate(int,String,boolean)
```

CDF 사양: @CalcMgrRollDate(date,date\_part,up)

## @CalcMgrRollDay

**목적:**

YYYYMMDD 형식의 날짜로 일을 롤업하거나 롤다운합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.rollDay(int,boolean)
```

CDF 사양: @CalcMgrRollDay(date,up)

## @CalcMgrRollMonth

**목적:**

YYYYMMDD 형식의 날짜로 월을 롤업하거나 롤다운합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.rollMonth(int,boolean)
```

CDF 사양: @CalcMgrRollMonth(date,up)

## @CalcMgrRollYear

**목적:**

YYYYMMDD 형식의 날짜로 연도를 롤업하거나 롤다운합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.rollYear(int,boolean)
```

CDF 사양: @CalcMgrRollYear(date,up)

## @CalcMgrWeeksBetween

### 목적:

YYYYMMDD 형식인 두 날짜 사이의 주를 반환합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.weeksBetween(int, int)
```

CDF 사양: @CalcMgrWeeksBetween(fromDate, toDate)

### 주:

@CalcMgrWeeksBetween은 양수만 반환합니다. 음수를 반환하려면(적용되는 경우) @CalcMgrWeeksDiff를 사용합니다.

## @CalcMgrWeeksDiff

### 목적:

YYYYMMDD 형식인 두 날짜 사이의 주를 반환합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.weeksDiff(int, int)
```

CDF 사양: @CalcMgrWeeksDiff(fromDate, toDate)

### 주:

@CalcMgrWeeksDiff를 사용하는 경우 함수의 첫번째 날짜가 두번째 날짜보다 이후이면 음수가 반환됩니다. 함수의 첫번째 날짜가 두번째 날짜보다 이전이면 양수가 반환됩니다. 양수만 반환하려는 경우 @CalcMgrWeeksBetween를 사용합니다.

## @CalcMgrYearsBetween

### 목적:

YYYYMMDD 형식인 두 날짜 사이의 연도를 반환합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.yearsBetween(int, int)
```

CDF 사양: @CalcMgrYearsBetween(fromDate, toDate)

**주:**

@CalcMgrYearsBetween은 양수만 반환합니다. 음수를 반환하려면(적용되는 경우) @CalcMgrYearsDiff를 사용합니다.

## @CalcMgrYearsDiff

### 목적:

YYYYMMDD 형식인 두 날짜 사이의 연도를 반환합니다.

### 구문:

Java 클래스: `com.hyperion.calcmgr.common.cdf.CalendarFunctions.yearsDiff(int,int)`

CDF 사양: @CalcMgrYearsDiff(fromDate,toDate)

**주:**

@CalcMgrYearsDiff를 사용하는 경우 함수의 첫번째 날짜가 두번째 날짜보다 이후이면 음수가 반환됩니다. 함수의 첫번째 날짜가 두번째 날짜보다 이전이면 양수가 반환됩니다. 양수만 반환하려는 경우 @CalcMgrYearsBetween를 사용합니다.

## 재무 함수

### 참조:

- [@CalcMgrExcelACCRINT](#)
- [@CalcMgrExcelACCRINTM](#)
- [@CalcMgrExcelAMORDEGRC](#)
- [@CalcMgrExcelAMORLINC](#)
- [@CalcMgrExcelCOUPDAYBS](#)
- [@CalcMgrExcelCOUPDAYS](#)
- [@CalcMgrExcelCOUPDAYSNC](#)
- [@CalcMgrExcelCOUPNCD](#)
- [@CalcMgrExcelCOUPNUM](#)
- [@CalcMgrExcelCOUPPCD](#)
- [@CalcMgrExcelCUMIPMT](#)
- [@CalcMgrExcelCUMPRINC](#)
- [@CalcMgrExcelDB](#)
- [@CalcMgrExcelDDB](#)
- [@CalcMgrExcelDISC](#)
- [@CalcMgrExcelDOLLARDE](#)



- @CalcMgrExcelDOLLARFR
- @CalcMgrExcelDURATION
- @CalcMgrExcelEFFECT
- @CalcMgrExcelFV
- @CalcMgrExcelFVSCCHEDULE
- @CalcMgrExcelMDURATION
- @CalcMgrExcelINTRATE
- @CalcMgrExcelPMT
- @CalcMgrExcelIRR
- @CalcMgrExcelISPMT
- @CalcMgrExcelMIRR
- @CalcMgrExcelNPER
- @CalcMgrExcelNPV
- @CalcMgrExcelPPMT
- @CalcMgrExcelPRICE
- @CalcMgrExcelPRICEDISC
- @CalcMgrExcelPRICEMAT
- @CalcMgrExcelPV
- @CalcMgrExcelRATE
- @CalcMgrExcelRECEIVED
- @CalcMgrExcelSLN
- @CalcMgrExcelSYD
- @CalcMgrExcelTBILLEQ
- @CalcMgrExcelTBILLPRICE
- @CalcMgrExcelTBILLYIELD
- @CalcMgrExcelXIRR
- @CalcMgrExcelXNPV
- @CalcMgrExcelYIELD
- @CalcMgrExcelYIELDDISC
- @CalcMgrExcelYIELDMAT

## @CalcMgrExcelACCRINT

### 목적:

정기적으로 이자가 납입되는 유가 증권의 경과 이자를 반환합니다.

### 구문:

Java 클래스:

`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.ACCRINT(double, double, double, d`

CDF 사양: @CalcMgrExcelACCRINT(issue, firstinterest, settlement, rate, par, frequency, basis, method)

## @CalcMgrExcelACCRINTM

### 목적:

만기에 이자가 납입되는 유가 증권의 경과 이자를 반환합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.ACCRINTM(issue, settlement, rate, par, basis)
```

CDF 사양:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.ACCRINTM(double, double, double,
```

## @CalcMgrExcelAMORDEGRC

### 목적:

감가상각 계수를 사용하여 각 회계 기간의 감가상각을 반환합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.AMORDEGRC(double, double, double,
```

CDF 사양: @CalcMgrExcelAMORDEGRC(cost, purchased, firstPeriod, salvage, period, rate, basis)

## @CalcMgrExcelAMORLINC

### 목적:

각 회계 기간의 감가상각을 반환합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.AMORLINC(double, double, double,
```

CDF 사양: @CalcMgrExcelAMORLINC(cost, date\_purchased, first\_period, salvage, period, rate, basis)

## @CalcMgrExcelCOUPDAYBS

### 목적:

쿠폰 기간 시작 날짜부터 정산 날짜까지의 기간(일)을 반환합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.COUPDAYBS(double, double, double,
```

CDF 사양: @CalcMgrExcelCOUPDAYBS(settlement, maturity, frequency, basis)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelCOUPDAYS

**목적:**

정산 날짜가 포함된 쿠폰 기간의 기간(일)을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.COUPDAYS(double, double, double,
```

```
CDF 사양: @CalcMgrExcelCOUPDAYS(settlement, maturity, frequency, basis)
```

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelCOUPDAYSNC

**목적:**

정산 날짜부터 다음 쿠폰 날짜까지의 기간(일)을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.COUPDAYSNC(double, double, double,
```

```
CDF 사양: @CalcMgrExcelCOUPDAYSNC(settlement, maturity, frequency, basis)
```

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelCOUPNCD

**목적:**

정산 날짜 이후 다음 쿠폰 날짜를 나타내는 수를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.COUPNCD(double, double, double, d
```

CDF 사양: @CalcMgrExcelCOUPNCD(settlement, maturity, frequency, basis)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. [날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수](#)를 참조하십시오.

## @CalcMgrExcelCOUPNUM

목적:

정산 날짜 및 만기일 사이에 지급 가능한 쿠폰 수를 가장 가까운 정수로 반올림하여 반환합니다.

구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.COUPNUM(double, double, double, d
```

CDF 사양: @CalcMgrExcelCOUPNUM(settlement, maturity, frequency, basis)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. [날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수](#)를 참조하십시오.

## @CalcMgrExcelCOUPPCD

목적:

정산 날짜 이전의 이전 쿠폰 날짜를 나타내는 수를 반환합니다.

구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.COUPPCD(double, double, double, d
```

CDF 사양: @CalcMgrExcelCOUPPCD(settlement, maturity, frequency, basis)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. [날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수](#)를 참조하십시오.

## @CalcMgrExcelCUMIPMT

**목적:**

start\_period에서 end\_period 사이에 대출에 대해 지불된 누적 이자를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.CUMIPMT(double, double, double, double, double, double, double)
```

CDF 사양: @CalcMgrExcelCUMIPMT(rate, nper, pv, start\_period, end\_period, type)

## @CalcMgrExcelCUMPRINC

**목적:**

시작 기간에서 종료 기간 사이에 대출에 대해 지불된 누적 원금을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.CUMPRINC(double, double, double, double, double, double, double)
```

CDF 사양: @CalcMgrExcelCUMPRINC(rate, per, nper, pv, fv, type)

## @CalcMgrExcelDB

**목적:**

고정 채감잔액법을 사용하여 지정된 기간에 대한 자산 감가상각을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.DB(double, double, double, double, double, double)
```

CDF 사양: @CalcMgrExcelDB((cost, salvage, life, period, month)

## @CalcMgrExcelDDB

**목적:**

이중 채감잔액법 또는 지정된 다른 방법을 사용하여 지정된 기간에 대한 자산 감가상각을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.DDB(double, double, double, double, double, double)
```

CDF 사양: @CalcMgrExcelDDB(cost, salvage, life, period, factor)

## @CalcMgrExcelDISC

**목적:**

유가 증권에 대한 할인을 반환합니다.

#### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.DISC(double, double, double, double, double)
```

CDF 사양: @CalcMgrExcelDISC(settlement, maturity, pr, redemption, basis)

#### 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelDOLLARDE

#### 목적:

정수 부분과 분수 부분으로 표시된 달러 가격을 1.02와 같이 소수로 표시된 달러 가격으로 변환합니다. 분수로 표시된 달러 금액이 유가 증권 가격에 사용되는 경우가 있습니다.

#### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.DOLLARDE(double, double)
```

CDF 사양: @CalcMgrExcelDOLLARDE(fractional\_dollar, fraction)

## @CalcMgrExcelDOLLARFR

#### 목적:

소수로 표시된 달러 가격을 분수로 표시된 달러 가격으로 변환합니다.

#### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.DOLLARFR(double, double)
```

CDF 사양: @CalcMgrExcelDOLLARFR(decimal\_dollar, fraction)

## @CalcMgrExcelDURATION

#### 목적:

정기적으로 이자가 납입되는 유가 증권의 연간 듀레이션을 반환합니다. 주:

@CalcMgrExcelMDURATION 함수를 사용하는 경우 Oracle Hyperion Calculation Manager와 Excel 간에 계산이 일치하지 않을 수도 있습니다. 수치가 일치하도록 하려면 소수를 7로 변경하고 Open Office를 사용합니다.

#### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.DURATION(double, double, double, double, double, double, double)
```

CDF 사양: @CalcMgrExcelDURATION(settlement, maturity, coupon, yld, frequency, basis)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelEFFECT

**목적:**

유효한 연간 이자율을 반환합니다.

**구문:**

Java 클래스:

com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.EFFECT(double, double)

CDF 사양: @CalcMgrExcelEFFECT(nominal\_rate, npery)

## @CalcMgrExcelFV

**목적:**

투자의 미래 가치를 반환합니다.

**구문:**

Java 클래스:

com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.FV(double, double, double, double)

CDF 사양: @CalcMgrExcelFV(rate, nper, pmt, pv, type)

## @CalcMgrExcelFVSCCHEDULE

**목적:**

일련의 복리 이자율을 적용한 후 초기 원금의 미래 가치를 반환합니다.

**구문:**

Java 클래스:

com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.FVSCCHEDULE(double, double[])

CDF 사양: @CalcMgrExcelFVSCCHEDULE(principal, schedule)

## @CalcMgrExcelMDURATION

**목적:**

액면가를 \$100로 가정하여 유가 증권의 Macauley 수정 듀레이션을 반환합니다. 주: @CalcMgrExcelDURATION 함수를 사용하는 경우 Oracle Hyperion Calculation Manager와 Excel 간에 계산이 일치하지 않을 수도 있습니다. 수치가 일치하도록 하려면 소수를 7로 변경하고 Open Office를 사용합니다.

**구문:**

Java 클래스:

`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.MDURATION(double, double, double,`CDF 사양: `@CalcMgrExcelMDURATION(settlement, maturity, coupon, yld, frequency, basis)` **주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelINTRATE

**목적:**

전액 투자 유가 증권에 대한 이자율을 반환합니다.

**구문:**

Java 클래스:

`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.INTRATE(double, double, double, d`CDF 사양: `@CalcMgrExcelINTRATE(settlement, maturity, investment, redemption, basis)` **주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelPMT

**목적:**

연금의 정기 지불을 반환합니다.

**구문:**

Java 클래스:

`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.PMT(double, double, double, doubl`CDF 사양: `@CalcMgrExcelPMT(rate, nper, pv, fv, type)`

## @CalcMgrExcelIRR

**목적:**

일련의 현금 흐름에 대한 내부 반환율을 반환합니다.

**구문:**



Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.IRR(double[], double)`

CDF 사양: `@CalcMgrExcelIRR(values, guess)`

## @CalcMgrExcelISPMT

**목적:**

특정 투자 기간 동안 지불된 이자를 계산합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.ISPMT(double, double, double, double)`

CDF 사양: `@CalcMgrExcelISPMT(rate, per, nper, pv)`

## @CalcMgrExcelMIRR

**목적:**

긍정적 현금 흐름과 부정적 현금 흐름이 서로 다른 비율로 재정 지원되는 환경의 내부 반환율을 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.MIRR(double[], double, double)`

CDF 사양: `@CalcMgrExcelMIRR(values, finance_rate, reinvest_rate)`

## @CalcMgrExcelNPER

**목적:**

투자 기간 수를 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.NPER(double, double, double, double, double)`

CDF 사양: `@CalcMgrExcelNPER(rate, pmt, pv, fv, type)`

## @CalcMgrExcelNPV

**목적:**

일련의 정기적 현금 흐름 및 할인율을 기준으로 투자의 순 현재 가치를 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.NPV(double, double[])`

CDF 사양: `@CalcMgrExcelNPV(rate, values)`

## @CalcMgrExcelPPMT

**목적:**

정기적 고정 지불 및 고정 이자율을 기준으로 투자에 대한 지정된 기간의 원금 지불을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.PPMT(double, double, double, double, double)
```

CDF 사양: @CalcMgrExcelPPMT(rate, per, nper, pv, fv, type)

## @CalcMgrExcelPRICE

**목적:**

정기적으로 이자가 납입되는 유가 증권의 액면가 \$100당 가격을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.PRICE(double, double, double, double, double, double, double)
```

CDF 사양: @CalcMgrExcelPRICE(settlement, maturity, rate, yld, redemption, frequency, basis)

**주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelPRICEDISC

**목적:**

할인된 유가 증권의 액면가 \$100당 가격을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.PRICEDISC(double, double, double, double, double, double)
```

CDF 사양: @CalcMgrExcelPRICEDISC(settlement, maturity, discount, redemption, basis)

**주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelPRICEMAT

**목적:**

만기에 이자가 납입되는 유가 증권의 액면가 \$100당 가격을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.PRICEMAT(double, double, double,
```

CDF 사양: @CalcMgrExcelPRICEMAT((settlement, maturity, issue, rate, yld, basis)

**주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelPV

**목적:**

투자의 현재 가치를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.PV(double, double, double, double,
```

CDF 사양: @CalcMgrExcelPV(rate, nper, pmt, fv, type)

## @CalcMgrExcelRATE

**목적:**

연금의 기간 당 이자율을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.RATE(double, double, double, double,
```

CDF 사양: @CalcMgrExcelRATE(nper, pmt, pv, fv, type, guess)

## @CalcMgrExcelRECEIVED

**목적:**

전액 투자 유가 증권의 만기 시 수령 금액을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.RECEIVED(double, double, double,
```

CDF 사양: @CalcMgrExcelRECEIVED(settlement, maturity, investment, discount, basis)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelSLN

**목적:**

특정 기간의 자산에 대한 정액 감가상각을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.SLN(double, double, double)
```

CDF 사양: @CalcMgrExcelSLN(cost, salvage, life)

## @CalcMgrExcelSYD

**목적:**

지정된 기간에 대한 연도 합계 자산 감가상각을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.SYD(double, double, double, double)
```

CDF 사양: @CalcMgrExcelSYD(cost, salvage, life, per)

## @CalcMgrExcelTBILLEQ

**목적:**

단기재정증권의 채권 등가 수익률을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.TBILLEQ(double, double, double)
```

CDF 사양: @CalcMgrExcelTBILLEQ(settlement, maturity, discount)

 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelTBILLPRICE

**목적:**

단기재정증권의 액면가 \$100당 가격을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.TBILLPRICE(double, double, double)
```

CDF 사양: @CalcMgrExcelTBILLPRICE(settlement, maturity, discount)

**주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelTBILLYIELD

**목적:**

단기재정증권 수익률을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.TBILLYIELD(double, double, double)
```

CDF 사양: @CalcMgrExcelTBILLYIELD(settlement, maturity, pr)

**주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelXIRR

**목적:**

정기적인 필요 없는 현금 흐름 스케줄 동안의 내부 반환율을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.XIRR(double[], double[], double)
```

CDF 사양: @CalcMgrExcelXIRR(values, dates, guess)

## @CalcMgrExcelXNPV

**목적:**

정기적인 필요 없는 현금 흐름 스케줄 동안의 순 현재 가치를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.XNPV(double, double[], double[])
```

CDF 사양: @CalcMgrExcelXNPV(rate, values, dates)

## @CalcMgrExcelYIELD

**목적:**

정기적으로 이자가 납입되는 유가 증권의 수익률을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.YIELD(double, double, double, double, double, double, double)
```

CDF 사양: @CalcMgrExcelYIELD(settlement, maturity, rate, pr, redemption, frequency, basis)

**주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelYIELDDISC

**목적:**

할인 유가 증권(예: 단기재정증권)의 연간 수익률을 반환합니다. 주: @CalcMgrExcelYIELDDISC 함수를 사용하는 경우 Oracle Hyperion Calculation Manager와 Excel 간에 계산이 일치하지 않을 수도 있습니다. 수치가 일치하도록 하려면 소수를 7로 변경하고 Open Office를 사용합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.YIELDDISC(double, double, double, double, double, double)
```

CDF 사양: @CalcMgrExcelYIELDDISC(settlement, maturity, pr, redemption, basis)

**주:**

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다. 날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## @CalcMgrExcelYIELDMAT

### 목적:

만기에 이자가 납입되는 유가 증권의 연간 수익률을 반환합니다. 주:  
@CalcMgrExcelYIELDMAT 함수를 사용하는 경우 Oracle Hyperion Calculation Manager와 Excel 간에 계산이 일치하지 않을 수도 있습니다. 수치가 일치하도록 하려면 소수를 7로 변경하고 Open Office를 사용합니다.

### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.YIELDMAT(double,double,double,
```

```
CDF 사양: @CalcMgrExcelYIELDMAT(settlement, maturity, issue, rate, pr, basis)
```

### 주:

@CalcMgrExcel 함수에 사용되는 날짜 매개변수는 Excel 형식이어야 합니다.  
날짜 매개변수가 포함된 @CalcMgrExcel 사용자정의 함수를 참조하십시오.

## 로그 함수

### 참조:

- [@CalcMgrLog](#)
- [@CalcMgrLogText](#)

## @CalcMgrLog

### 목적:

계산 스크립트에서 진단을 로그합니다. 파일 이름이 비어 있으면 메시지가 콘솔에 기록됩니다

### 구문:

Java Class:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.log(String,String,String[],double[],boolean)
```

```
CDF Spec: CalcMgrLog(fileName,msg,strings,values,printDate)
```

### 예 1:

```
FIX ("Sales",@Children("100"),"Florida")
    "Budget" (
        @CalcMgrLog("c:/temp/cal.log", "Log: ",
            @LIST(@NAME(@CURRMBR("Year")),
                @NAME(@CURRMBR("Measures")),
                @NAME(@CURRMBR("Product")),
                @NAME(@CURRMBR("Market")),
                @NAME(@CURRMBR("Scenario"))
```

```

        ),
        @LIST(@CURRMBR("Year"),
              @CURRMBR("Measures"),
              @CURRMBR("Product"),
              @CURRMBR("Market"),
              @CURRMBR("Scenario")
            ),
        @_True
    );
)
ENDFIX
The log file entry might look like this:

```

이 예에서 로그 파일 항목은 다음과 같습니다.

```
2014-12-20 15:38:13 Log: Apr Sales 100-10 Florida Budget 210.0 210.0 210.0
210.0 210.0
```

## 예 2:

```

FIX ("Sales",@Children("100"),"Florida")
  "Budget" (
    @CalcMgrLog(
      "c:/temp/cal.log",
      "Log: ",
      @LIST(
        @CalcMgrStringsToString(
          @LIST( @NAME( @CURRMBR( "Year")),
                @NAME( @CURRMBR( "Measures")), @NAME( @CURRMBR( "Product")),
                @NAME( @CURRMBR( "Market")), @NAME( @CURRMBR( "Scenario"))),
          ",",
          @CalcMgrFormatDouble( "%7.2f", @CURRMBR( "Year")-
>@CURRMBR( "Measures")->@CURRMBR( "Product")->@CURRMBR( "Market")-
>@CURRMBR( "Scenario"))
        ),
        @LIST( 0.0), @_True);
    )
  )
ENDFIX

```

이 예에서 로그 파일 항목은 다음과 같습니다.

```
2014-12-20 15:40:10 Log: Apr,Sales,100-10,Floridshoula,Budget 210.00 0.0
```

## @CalcMgrLogText

### 목적:

로그 파일에 메시지를 씁니다. 파일 이름이 비어 있으면 메시지가 콘솔에 기록됩니다.

### 구문:

Java Class:  
com.hyperion.calcmgr.common.cdf.StringFunctions.logText (String,String,boolean)



```
CDF Spec: @CalcMgrLogText(fileName,msg,printDate)
```

**예:**

```
FIX ("Sales",@Children("100"),"Florida")
    "Budget" (
        @CalcMgrLogText (
            "c:/temp/cal.log",
            "text of message to display" @_False);
        )
ENDFIX
```

## 수리 함수

**참조:**

- [@CalcMgrExcelCEILING](#)
- [@CalcMgrExcelCOMBIN](#)
- [@CalcMgrExcelEVEN](#)
- [@CalcMgrExcelFACT](#)
- [@CalcMgrExcelFLOOR](#)
- [@CalcMgrExcelGCD](#)
- [@CalcMgrExcelLCM](#)
- [@CalcMgrExcelMROUND](#)
- [@CalcMgrExcelMULTINOMIAL](#)
- [@CalcMgrExcelODD](#)
- [@CalcMgrExcelPOWER](#)
- [@CalcMgrExcelPRODUCT](#)
- [@CalcMgrExcelROUNDDOWN](#)
- [@CalcMgrExcelROUNDUP](#)
- [@CalcMgrExcelSQRT](#)
- [@CalcMgrExcelSQRTPI](#)
- [@CalcMgrExcelSUMPRODUCT](#)
- [@CalcMgrExcelSUMSQ](#)

### @CalcMgrExcelCEILING

**목적:**

significance에 가장 가까운 정수 또는 가장 가까운 배수로 숫자를 반올림합니다(0에서 멀어지도록).

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.CEILING(double, double)
```

CDF 사양: @CalcMgrExcelCEILING(number, significance)

## @CalcMgrExcelCOMBIN

**목적:**

지정된 수의 객체에 대한 조합 수를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.COMBIN(double, double)
```

CDF 사양: @CalcMgrExcelCOMBIN(number, number\_chosen)

## @CalcMgrExcelEVEN

**목적:**

숫자를 가장 가까운 짝수로 반올림합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.EVEN(double)
```

CDF 사양: @CalcMgrExcelEVEN(number)

## @CalcMgrExcelFACT

**목적:**

숫자의 계승을 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.FACT(double)
```

CDF 사양: @CalcMgrExcelFACT(number)

## @CalcMgrExcelFLOOR

**목적:**

숫자를 0에 더 가깝게 반내림합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.FLOOR(double, double)
```

CDF 사양: @CalcMgrExcelFLOOR(number, significance)

## @CalcMgrExcelGCD

**목적:**

최대공약수를 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.GCD(double[])`  
CDF 사양: `@CalcMgrExcelGCD(numbers)`

## @CalcMgrExcelLCM

**목적:**

최소공배수를 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.LCM(double[])`  
CDF 사양: `@CalcMgrExcelLCM(numbers)`

## @CalcMgrExcelMROUND

**목적:**

숫자를 지정된 자릿수로 반올림합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.MROUND(double, double)`  
CDF 사양: `@CalcMgrExcelMROUND(number, num_digits)`

## @CalcMgrExcelMULTINOMIAL

**목적:**

다항 숫자 세트를 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.MULTINOMIAL(double[])`  
CDF 사양: `@CalcMgrExcelMULTINOMIAL(numbers)`

## @CalcMgrExcelODD

**목적:**

숫자를 가장 가까운 홀수로 반올림합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.ODD(double)`  
CDF 사양: `@CalcMgrExcelODD(number)`

## @CalcMgrExcelPOWER

**목적:**

숫자를 거듭제곱한 결과를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.POWER(double, double)
```

CDF 사양: @CalcMgrExcelPOWER(number, power)

## @CalcMgrExcelPRODUCT

**목적:**

해당 인수를 곱합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.PRODUCT(double[])
```

CDF 사양: @CalcMgrExcelPRODUCT(numbers)

## @CalcMgrExcelROUNDDOWN

**목적:**

숫자를 0에 더 가깝게 반내림합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.ROUNDDOWN(double, double)
```

CDF 사양: @CalcMgrExcelROUNDDOWN(number, num\_digits)

## @CalcMgrExcelROUNDUP

**목적:**

숫자를 0에서 멀어지도록 반올림합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.ROUNDUP(double, double)
```

CDF 사양: @CalcMgrExcelROUNDUP(number, num\_digits)

## @CalcMgrExcelSQRT

**목적:**

양의 제곱근을 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.SQRT(double)`  
CDF 사양: `@CalcMgrExcelSQRT(number)`

## @CalcMgrExcelSQRTPI

**목적:**

(`number * pi`)의 제곱근을 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.SQRTPI(double)`  
CDF 사양: `@CalcMgrExcelSQRTPI(number)`

## @CalcMgrExcelSUMPRODUCT

**목적:**

해당 배열 구성요소의 곱의 합을 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.SUMPRODUCT(double[],double[])`  
CDF 사양: `@CalcMgrExcelSUMPRODUCT(values1, values2)`

## @CalcMgrExcelSUMSQ

**목적:**

인수의 제곱 합을 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.SUMSQ(double[])`  
CDF 사양: `@CalcMgrExcelSUMSQ(numbers)`

## MaxL 함수

 **주:**

FIX 명령문 내에서 함수를 호출하지 않고 RUNJAVA 명령을 통해 MAXL 함수를 사용하는 것이 좋습니다. 스크립트 구성요소에서는 RUNJAVA 명령만 사용할 수 있습니다.

## @CalcMgrExecuteEncryptMaxLFile

**목적:**

Oracle Essbase에서 액세스할 수 있는 MaxL 스크립트 파일을 사용하여 MaxL 스크립트를 실행합니다.

**구문:**

Java Class: com.hyperion.calcmgr.common.cdf.MaxLFunctions

CDF Spec: @CalcMgrExecuteEnMaxLFile(privateKey, maxlFileName, arguments, asynchronous)

## @CalcMgrExecuteMaxLScript

**목적:**

MaxL 스크립트를 계산 스크립트 자체에 포함시킵니다.

**구문:**

Java Class: com.hyperion.calcmgr.common.cdf.MaxLFunctions

CDF Spec: @CalcMgrExecuteMaxLEnScript(privateKey, maxlScripts, arguments, asynchronous)

## MaxL 암호화 사용

MaxL 암호화를 사용하려면 다음을 수행합니다.

1. `essmsh`를 사용하여 공개 키와 개인 키를 생성합니다. `essmsh -gk` 명령을 실행하면 콘솔에 개인 및 공개 키가 생성됩니다. 암호화를 위한 공개 키: 25159,850102129; 해독을 위한 개인 키: 240764119,850102129.
2. 공개 키 `essmsh -ep admin 25159,850102129`를 사용하여 암호화된 토큰을 생성합니다. 906388712099924604712352658511과 비슷한 토큰이 생성될 수 있습니다.
3. `localhost`의 MaxL 스크립트 로그인 `$key $1 $key $2`에 토큰을 사용합니다.
4. 개인 키와 함께 `-D` 옵션으로 `essmsh`를 호출하여 `essmsh -D` 스크립트를 실행합니다.  
예: `essmsh C:/Temp/maxlcmd1.xmls 240764119, 850102129 906388712099924604712352658511 0893542980829559883146306518502837293210.`

## @CalcMgrExecuteEncryptMaxLFile을 사용하여 MaxL 스크립트 파일 실행

@CalcMgrExecuteEncryptMaxLFile의 구문은 다음과 같습니다.

```
FIX ("Sales", "100-10", "New York", "Actual")
    "Jan" (
        @CalcMgrExecuteEncryptMaxLFile("2115028571,2505324337", "C:/Temp/
exportdata.xmls",
    @List("906388712099924604712352658511", "0893542980829559883146306518502837293210"),
    "true");
    ENDFIX
```

첫 번째 인수는 개인 키이고, 두 번째 인수는 MaxL 스크립트 파일이며, 세 번째 인수는 스크립트의 인수이고, 마지막 인수는 비동기적으로 실행하는지 여부를 표시합니다("true"는 비동기를 표시하고, "false"는 동기를 표시).

이 방법에는 몇 가지 이슈가 있습니다. 첫 번째 이슈는 동일한 스크립트를 여러 번 실행해야 한다는 점입니다. FIX 문의 모든 멤버 조합에 대해 실행됩니다. 두 번째 이슈는 Oracle Essbase에서 FIX 문의 멤버 조합에 블록이 없다고 확인하면 스크립트가 전혀 실행되지 못한다는 점입니다.

일반적으로 MaxL 스크립트를 한 번만 실행하려 하며, 다음과 같이 RUNJAVA 명령을 실행합니다.

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLFunctions
  <LOG FILE> OPTIONAL ex., "logfile=c:/Temp/maxlrule.log" The path
  should be accessible from the ESSBASE process.
  <ASYNCH FLAG> OPTIONAL ex., "true" This means calc engine does
  not wait for the completion of the MaxL script. The default is "false"
  <ALL OTHER PARAMETERS YOU PASS TO ESSMSH>
```

예:

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLFunctions
"logfile=c:/Temp/maxlrule.log"
"-D"
"c:\\Temp\\maxl.msh"
"28567271,1186612387"
"893848844082678214004255849650"
"4647275840619320283077900267208176084380"
"{varAppName}";
```

## @CalcMgrExecuteMaxLScript를 사용하여 MaxL 스크립트 실행

@CalcMgrExecuteMaxLScript의 구문은 다음과 같습니다.

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLScriptFunction
  <LOG FILE> OPTIONAL ex., "logfile=c:/Temp/maxlrule.log" The path
  should be accessible from the ESSBASE process
  <ASYNCH FLAG> OPTIONAL ex., "true" This means calc engine will
  not wait for the completion of the maxl script.
  "mdx=true"OPTIONAL The variables are replaced with MDX syntax.
  This applies only to variables within the Crossjoin sections.
  "-D" if using encryption
  "<PRIVATE KEY>" if using encryption
  <SCRIPT LINE>
  <SCRIPT LINE>
  <SCRIPT LINE>
  ...
  <ALL OTHER PARAMETERS YOU PASS TO ESSMSH>
```

### 예 1: @CalcMgrExecuteMaxLScript 사용

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLScriptFunction
"-D"
```

```

"28567271,1186612387"
"spool on to 'C:\\ALC\\out.log'"
"login $key 893848844082678214004255849650 $key
4647275840619320283077900267208176084380 on localhost"
"alter database 'SampleASO'. 'Basic' clear data in region
'Crossjoin(Descendants ([TotalPV], Province.Levels (0)), Crossjoin (Descendants ([TotalLOB], LOB.
Crossjoin(Descendants ([TotalActivity], Activity.Levels (0)), Crossjoin (Descendants ([TotalBranch], Branch.Levels (0)), Crossjoin (Descendants ([BS000], Account.Levels (0)), Crossjoin ({ [FY14] }, Crossjoin ({ [Working] }, Crossjoin ({ [Actual] })))))))))' physical"
"logout"
"spool off"
"exit"

```

### 예 2: 런타임 프롬프트 변수 및 MaxL 인수 표시자 사용

스크립트가 고정되어 있지 않으나, 런타임 프롬프트 또는 MaxL에 대한 인수를 기반으로 한다고 가정합니다. 따옴표로 묶은 텍스트 내에 런타임 프롬프트를 사용하거나 MaxL에서 따옴표로 묶은 텍스트 내에 \$1을 사용하는 경우 Oracle Hyperion Planning에서 항목을 대체하지 않습니다. 아래 예에서는 "Working"으로 하드 코딩하지 않고 런타임 프롬프트를 사용합니다. 런타임 프롬프트 및 MaxL 인수 표시자를 사용하여 스크립트를 재작성할 수 있습니다.

```

RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLScriptFunction
"-D"
"28567271,1186612387"
"spool on to 'C:\\ALC\\out.log'"
"login $key 893848844082678214004255849650 $key
4647275840619320283077900267208176084380 on localhost"
"alter database 'SampleASO'. 'Basic' clear data in region
'Crossjoin(Descendants ([TotalPV], Province.Levels (0)), Crossjoin (Descendants ([TotalLOB], LOB.
Crossjoin(Descendants ([TotalActivity], Activity.Levels (0)), Crossjoin (Descendants ([TotalBranch], Branch.Levels (0)), Crossjoin (Descendants ([BS000], Account.Levels (0)), Crossjoin ({ [FY14] }, Crossjoin ({ [$1] }, Crossjoin ({ [Actual] })))))))))' physical"
"logout"
"spool off"
"exit"
{WhatVersion}

```

이 예에서 {WhatVersion}은 런타임 프롬프트 변수입니다. 이 예에서 "exit" 뒤에 오는 항목은 스크립트에 대한 인수로 간주됩니다. 사용자 정의 함수가 인수 수를 기반으로 스크립트의 \$1, \$2..\$n을 바꿉니다.

### 예 3: MaxL=True 사용

```

RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLScriptFunction
....."mdx=true"
....."-D"
....."28567271,1186612387"
....."spool on to '/tmp/Essbase_ClearAndLoadStandardFX.log'"
....."login $key 893848844082678214004255849650 $key
4647275840619320283077900267208176084380 on localhost"

```



```

....."alter database glblaso.aso clear data in region '
CrossJoin( {$1} , { ( $2 , [Actual] , $3 , [USD] )} )' physical "
....."import database 'GlblASO'. 'ASO' data from server text data_file
'../../StdFxExport.txt' using server rules_file 'LdASO' on
.....error write to './tmp/EssbaseLoadASOStdFX_errors.log' "
"logout"
"spool off"
"exit"
"LE_83901,LE_83911"
"Jan"
"2014"

```

이 예에 나오는 구문의 결과 스크립트는 다음과 같습니다.

```

spool on to './tmp/Essbase_ClearAndLoadStandardFX.log';
login $key 893848844082678214004255849650 $key
4647275840619320283077900267208176084380 on localhost;
alter database glblaso.aso clear data in region '
CrossJoin([LE_83901], [LE_83911] , { ( [Jan] , [Actual] , [2014] ,
[USD] )} )' physical ;
import database 'GlblASO'. 'ASO' data from server text data_file
'../../StdFxExport.txt' using server rules_file 'LdASO' on
error write to './tmp/EssbaseLoadASOStdFX_errors.log' ;
logout;
spool off;
exit;

```

**참고:** MDX=true는 선택사항입니다. MDX=true를 사용하면 변수를 Crossjoin 섹션의 MDX 구문으로 바꿉니다.

**예 4: 큰따옴표와 세미콜론이 있는 함수 목록 사용**  
CDF를 통해 다음 계산을 실행한다고 가정하겠습니다.

```

RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLScriptFunction
"spool on to 'c:/tmp/calc.log'"
"login admin password on localhost"
"execute calculation "
SET CREATEBLOCKONEQ ON;
SET AGGMISSG ON;
FIX (@IDESCENDANTS("BS"), "Plan")
    DATACOPY "Working" to "Target";
ENDFIX

FIX ("Plan", "Target")
    CALC DIM ("Account", "Period", "Entity", "Product");
ENDFIX'
on Vision.Plan1"
"logout"
"spool off"
"exit"
;

```

```

OK/INFO - 1012553 - Copying data from [Working] to [Target] with
fixed members [Account(1110, 1150, 1100, 1210, 1220, 1260, 1200, 1310,

```

```
1332, 1340, 1300,
1410, 1460, 1 471, 1400, 1010, 1510, 1520, 1530, 1540, 1550, 1560,
1565, 1570, 1580, 1500, 1610, 1620, 1630, 1640, 1650, ].
OK/INFO - 1012675 - Commit Blocks Interval for the calculation is
[3000].
OK/INFO - 1012675 - Commit Blocks Interval for the calculation is
[3000].
OK/INFO - 1012574 - Datacopy command copied [0] source data blocks to
[0] target data blocks.
OK/INFO - 1012675 - Commit Blocks Interval for the calculation is
[3000].
OK/INFO - 1012714 - Regular member [Sales Calls] depends on dynamic-
calc member [Rolling]..
OK/INFO - 1012714 - Regular member [Sales Calls] depends on dynamic-
calc member [YearTotal]..
OK/INFO - 1012714 - Regular member [Period] depends on dynamic-calc
member [YearTotal]..
OK/INFO - 1012684 - Multiple bitmap mode calculator cache memory usage
has a limit of [5] bitmaps..
OK/INFO - 1012668 - Calculating [ Account(All members) Period(All
members) Entity(All members) Product(All members)
with fixed members [Scenario(Plan); Version(Target)].
OK/INFO - 1012677 - Calculating in serial.
OK/INFO - 1012550 - Total Calc Elapsed Time : [0.002] seconds.
OK/INFO - 1013274 - Calculation executed.

MAXL> logout;
User admin is logged out
MaxL Shell completed
```

이 스크립트에서는 다음을 생성합니다.

```
essmsh C:\Users\SRMENON\AppData\Local\Temp\2\calc8153268024796650753.mxl
```

```
Essbase MaxL Shell 64-bit - Release 11.1.2 (ESB11.1.2.4.000B154)
Copyright (c) 2000, 2014, Oracle and/or its affiliates.
All rights reserved.
```

```
MAXL> login admin password on localhost;
""
OK/INFO - 1051034 - Logging in user [admin@Native Directory].
OK/INFO - 1241001 - Logged in to Essbase.
```

```
MAXL> execute calculation '
2> SET CREATEBLOCKONEQ ON;
3> SET AGGMISSG ON;
4> FIX (@IDESCENDANTS("BS"), "Plan")
5> DATACOPY "Working" to "Target";
6> ENDFIX
7>
8> FIX ("Plan", "Target")
9> CALC DIM ("Account", "Period", "Entity", "Product");
10> ENDFIX'
11> on Vision.Plan1;
```



@CalcMgrMDXDataCopy는 복사할 영역을 정의하는 MDX 표현식에 따라 달라집니다. 메모리에 그리드를 생성하고, MDX 표현식을 사용하여 소스 데이터베이스에서 그리드를 채운 다음, 타겟 데이터베이스로 데이터를 업데이트합니다. @CalcMgrMDXDataCopy는 calcmgrCmdLine.jar을 통해 암호화된 사용자 이름 및 비밀번호도 사용합니다.

예를 들어 사용자 이름과 비밀번호를 암호화하려면 `java -jar calcmgrCmdLine.jar -gk`를 사용하여 키를 생성해야 합니다. 사용자 이름 및 비밀번호를 암호화하는 데 사용할 수 있는 `83qosW4LbxVGp6uBwqtQWb`와 같은 키가 반환됩니다. 사용자 이름을 암호화하는 경우 비밀번호도 암호화해야 합니다. CalcMgrCmdLine.jar 사용에 대한 자세한 내용은 [명령행 실행 프로그램으로 Essbase Business Rules 실행](#)을 참조하십시오.



#### 주:

열과 행 축만 있는 MDX 표현식이 지원됩니다.

#### 구문:

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MDXDataCopy
  <Key used to encrypt user name and password>
  <Encrypted user name>
  <Encrypted password>
  <Source Application>
  <Source Database>
  <Target Application>
  <Target Database>
  <MDX expression that defines column members>
  <MDX expression that defines row members>
  <Source Member Mappings>
  <Target Member Mappings>
  <Target POV columns>
  <Rows Per Page>
  <Log File>
  <Source Server>
  <Target Server>
  <Where MDX>
  <"true" or "false">
```

- <Source Member Mappings> 및 <Target Member Mappings>는 소스의 멤버 이름이 타겟과 다른 경우에 사용됩니다. 쉼표로 구분되어야 하며 함수를 사용할 수 없습니다.
- <Target POV columns>는 소스보다 타겟에 차원이 더 많은 경우 사용됩니다.
- <ROWS PER PAGE>는 타겟 업데이트를 트리거하는 데 사용됩니다. 기본값은 -1이며, 전체 그리드가 채워지면 업데이트가 수행됩니다. 30으로 설정하는 경우 30개 행이 채워지는 즉시 업데이트가 트리거됩니다. 그런 다음, 다음 30개 행으로 계속 진행합니다.
- <Source Server> 및 <Target Server>는 소스 및 타겟 서버가 서로 다른 경우에 사용됩니다. 서로 다르지 않은 경우 ""를 자리 표시자로 사용해야 합니다.
- <Where MDX>는 스크립트에 인수를 지정하려는 경우 사용됩니다.
- <"true" 또는 "false"> 매개변수의 경우 "true"는 로그 파일을 추가하고 "false"는 로그 파일을 지웁니다. 이 매개변수를 비워 두는 경우 기본값은 "True"입니다.

사용자정의 함수가 실행되는 JVM의 메모리를 늘리려면 opmn.xml 파일을 Oracle Essbase에 대해 편집하고 ESS\_JVM\_OPTION1의 값 등록정보에 있는 숫자를 변경하십시오. 이 항목은 JVM <variable id="ESS\_JVM\_OPTION1" value="-Xmx256M"/>에 대해 256MB의 메모리를 설정합니다.

예를 들면 다음과 같습니다.

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MDXDataCopy
      "vta2a536uC/wyX8jM2G1HA==" /* key */

"qza9xZxUX+srS1Gr1R2Qboeq77InwkNSQSVBHL02siidSEglMndoIpneZVPK0tWt" /*
user */
      "eldizBRiyqdPdXl9Bpu/
HIE+YIgvRBa59aTjTDWpWM0T+ZEPjF+zyINfE0KHX0qP" /* password */
      "Sample" /* from application */
      "Basic" /* from database */
      "Samp2" /* to application */
      "Basic" /* to database */
      "crossjoin([[Jan],[Feb],[March]],{{Budget}})" /* MDX that defines
the column members */
      "crossjoin(crossjoin([[Sales]],{{100}.CHILDREN}),
[[Connecticut]])" /* MDX that defines the row members */
      "Jan,Feb" /* source member mappings */
      "January,February" /* target member mappings */
      "VP, IT" /* Target POV columns, members from dimensions that do
not exist on the source*/
      "-1" /* rows per page */
      "c:\\Temp\\cpdata3.log"; /* log file could be empty */
      ""source server
      ""destination server
      ""where MDX
```

## @CalcMgrMDXDataCopyRTP

### 목적:

런타임 프롬프트를 사용하여 한 데이터베이스의 데이터를 다른 데이터베이스로 복사하거나 MDX 스크립트를 사용하여 한 서버의 데이터를 다른 서버로 복사합니다.

RTP를 다음 인수에 사용할 수 있습니다.

- 열 MDX
- 행 MDX
- Where MDX
- 소스 매핑
- 타겟 매핑
- POV 열

이러한 인수의 경우 변수를 사용하지 않고 \$1, \$2, \$3 등을 사용합니다. 그런 다음 인수에서 @LIST 함수를 사용하여 이러한 변수를 전달합니다. 첫번째 인수의 입력은 \$1를 대체하고, 두번째 인수는 \$2를 대체하는 방식으로 계속됩니다.

### 구문:

```
Java
Class: .hyperion.calcmgr.common.cdf.MDXDataCopy.dataCopy(String,String,String,String,String,
CDF Spec:
@CalcMgrMDXDataCopyRTP(key,user,password,sourcApplication,sourrcDatabase,dtnApplication,dtnD
```

### **@CalcMgrMDXDataCopyRPT를 사용하여 한 데이터베이스의 데이터를 다른 데이터베이스로 복사**

이 예에서는 런타임 프롬프트 인수에 변수를 사용하지 않고 \$1, \$2, \$3 등을 사용합니다. 인수에 @LIST 함수를 사용하여 이러한 변수를 전달했습니다. 첫번째 인수의 입력은 \$1를 대체하고, 두번째 인수의 입력은 \$2를 대체하는 방식으로 계속됩니다.

#### **예:**

```
FIX ("FY14", {Scenario}, {Version}, {Department})
    "Jan" (
        @CalcMgrMDXDataCopyRTP(
            "fht1qmevLWMqKSHF2yusKS",
            "py9FTBsRCwp8afii9ntCAwSvVUXr3oZbpxFMqobx5Jwkt34Cfxvhs21z4WmG9KT",
            "KvwlKgs56CoilHQsrklUwiVi9zyoE+Llel4BrYBHiJz9PF6ZDVQ/
rF5taip707CL",
            "Vision",
            "Plan1",
            "",
            "ASO_401",
            "ASO_401",
            "",
            "[{$1}]",
            "NON EMPTY (Crossjoin({Descendants([&QRFPer1],
[Period].Levels(0)),Descendants([&QRFPer2],
[Period].Levels(0)),Descendants([&QRFPer3],
[Period].Levels(0)),Descendants([&QRFPer4],[Period].Levels(0))},
            Crossjoin([&QRFYr1],[&QRFYr2],[&QRFYr3],[&QRFYr4],Crossjoin([{$3}],
Crossjoin(Descendants([P_TP],[Product].Levels(0)),{[Units],[5800],
[9000]}))))))",
            "([{$2})",
            "",
            "",
            "",
            "",
            "-1",
            @LIST(@name({Scenario}),@name({Department}),@name({Version})),
            "c:/Temp/CM_DCopyVW1.log"
        )
    )
ENDFIX
```

## **@CalcMgrMDXDataCopyServer**

### **목적:**

MDX 스크립트를 사용하는 where 명령문 사용 옵션으로 한 데이터베이스에서 다른 데이터베이스로 또는 한 서버에서 다른 서버로 데이터를 복사합니다.

### **구문:**

Java Class:

```
CDF Spec: @CalcMgrMDXDataCopyServer(Encryption Key, user,
password,sourceApplicaionName,sourceDatabaseName,
sourceServer,targetApplicationName, targetDatabaseName,targetServer,
columnAxisMDX, rowAxisMDX,whereMDX,sourceMemberMappings,
targetMemberMappings,targetPOVCols,maxRowsPerPage, loggingFileName)
```

## @CalcMgrMDXExport

### 목적:

MDX 표현식을 사용하여 데이터를 익스포트합니다. 이 함수는 블록 저장영역 애플리케이션 계산 스크립트를 사용하여 집계 저장영역 데이터베이스에서 데이터를 익스포트하는 데 사용됩니다. @CalcMgrMDXExport에서는 calcmgrCmdLine.jar을 사용하여 암호화된 사용자 이름 및 비밀번호로 암호화된 토큰을 생성합니다. 예를 들어, 사용자 관리를 암호화하려면 `java -jar calcmgrCmdLine.jar -encrypt -key vta2a536uC/wyX8jm2G1HA== admin`을 사용합니다.

### 주:

열과 행 축만 있는 MDX 표현식이 지원됩니다.

### 구문:

Java Class:

```
com.hyperion.calcmgr.common.cdf.MDXExport.exportData(String,String,String,String,String,Stri
```

CDF Spec:

```
@CalcMgrMDXExport(key,user,pwd,fileName,application,database,server,columnAxisMdx,rowAxisMdx
```

**@CalcMgrMDXExport를 사용하여 데이터베이스에서 데이터 익스포트**

### 구문:

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MDXExport
<KEY used to encrypt user name and password>
<Encrypted user name>
<Encrypted password>
<Export file>
<Application>
<Database>
<Server>
  <MDX expression that defines column members>
  <MDX expression that defines row members>
  <Separator>
  <#MISSING Value>
  <Use Unique Name> "true" or "false"
  <Use Alias Name> "true" or "false"
  <Supress Zeros> "true" or "false"
  <Show Row Headers> "true" or "false"
  /* where mdx */
  /* source member mappings */
```

```
/* target member mappings */
```

**예:**

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MDXExport
    "fht1qmevLWMqKSHF2yusKS" /* key */
    "py9FTBsRCwp8afii9ntCAwSvVUXr3oZbxpxFMqobx5Jwkt34Cfxvhs21z4WmG9KT" /*
user */
    "KvwlKgs56CoilHQsrklUwiVi9zyoE+Llel4BrYBHiJz9PF6ZDVQ/rF5taip707CL" /*
password */
    "c:\\Temp\\export_W1.txt" /* export file */
    "Vision" /* source application name */
    "Plan1" /* source Database name */
    "server1" /* server */
    "[{Actual}]" /* columnAxisMDX */
    "NON EMPTY (Crossjoin({Descendants([&QRFPer1],
[Period].Levels(0)),Descendants([&QRFPer2],
[Period].Levels(0)),Descendants([&QRFPer3],
[Period].Levels(0)),Descendants([&QRFPer4],[Period].Levels(0))},
    Crossjoin({[Department]}},
    Crossjoin([&QRFPer1],[&QRFPer2],[&QRFPer3],[&QRFPer4]},
    Crossjoin({[Version]}},CrossJoin([Basedata]},
    {[Units],[5800],[4110]}))))))" /* row AxisMdx */
    "," /* separator */
    "" /* #MISSING */
    "false" /* use unique name */
    "false" /* use alias name */
    "true" /* suppress zero's */
    "true" /* show row headers */
    "[{P_000}]" /* where mdx */
    "Jan, Feb, Mar" /* source member mappings */
    "January, February, March" /* target member mappings */
```

## @CalcMgrMDXExportRTP

**목적:**

MDX 스크립트를 사용하여 큐브에서 런타임 프롬프트로 데이터를 익스포트합니다.

**구문:**

Java

```
Class:.hyperion.calcmgr.common.cdf.MDXExport.dataExport (String,String,String,String,String,S
```

CDF Spec:

```
@CalcMgrMDXExportRTP (key,user,pwd,fileName,application,database,server,columnAxisMdx,rowAxis
```

**@CalcMgrMDXExportRTP를 사용하여 데이터베이스에서 데이터 익스포트**  
다음 인수에 런타임 프롬프트를 사용할 수 있습니다.

- 열 MDX
- 행 MDX
- Where MDX



- 소스 매핑
- 타겟 매핑
- POV 열

이러한 인수에 경우 변수를 사용하지 않고 \$1, \$2, \$3 등을 사용합니다. 인수에 @LIST 함수를 사용하여 이러한 런타임 프롬프트를 전달합니다. 첫번째 인수의 입력은 \$1를 대체하고, 두번째 인수는 \$2를 대체하는 방식으로 계속됩니다.

## 통계 함수

### 참조:

- @CalcMgrExcelAVEDEV
- @CalcMgrExcelDEVSQ
- @CalcMgrExcelLARGE
- @CalcMgrExcelMEDIAN
- @CalcMgrExcelSMALL
- @CalcMgrExcelSTDEV
- @CalcMgrExcelVAR
- @CalcMgrExcelVARP

### @CalcMgrExcelAVEDEV

#### 목적:

평균에서 데이터 포인트의 절대 편차 평균을 반환합니다.

#### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.AVEDEV(double[])
```

CDF 사양: @CalcMgrExcelAVEDEV(numbers)

### @CalcMgrExcelDEVSQ

#### 목적:

편차의 제곱 합을 반환합니다.

#### 구문:

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.DEVSQ(double[])
```

CDF 사양: @CalcMgrExcelDEVSQ(numbers)

### @CalcMgrExcelLARGE

#### 목적:

n번째 가장 높은 수를 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.NTHLARGEST(double[], double)`  
CDF 사양: `@CalcMgrExcelLARGE(values, rank)`

## @CalcMgrExcelMEDIAN

**목적:**

지정된 수의 중앙값을 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.MEDIAN(double[])`  
CDF 사양: `@CalcMgrExcelMEDIAN(values)`

## @CalcMgrExcelSMALL

**목적:**

n번째 가장 낮은 수를 반환합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.NTHSMALLEST(double[], double)`  
CDF 사양: `@CalcMgrExcelSMALL(values, rank)`

## @CalcMgrExcelSTDEV

**목적:**

샘플을 기준으로 표준 편차를 계산합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.STDEV(double[])`  
CDF 사양: `@CalcMgrExcelSTDEV(values)`

## @CalcMgrExcelVAR

**목적:**

샘플을 기준으로 차이를 계산합니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.VAR(double[])`  
CDF 사양: `@CalcMgrExcelVAR(values)`

## @CalcMgrExcelVARP

**목적:**

전체 모집단을 기준으로 차이를 계산합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.VARP(double[])
```

CDF 사양: @CalcMgrExcelVARP(values)

## 문자열 함수

**참조:**

- [@CalcMgrCompare](#)
- [@CalcMgrConcat](#)
- [@CalcMgrDecimalFormat](#)
- [@CalcMgrDoubleFromString](#)
- [@CalcMgrDoubleToString](#)
- [@CalcMgrDQuote](#)
- [@CalcMgrEndsWith](#)
- [@CalcMgrFindFirst](#)
- [@CalcMgrFindLast](#)
- [@CalcMgrFormatDouble](#)
- [@CalcMgrGetListCount](#)
- [@CalcMgrGetListItem](#)
- [@CalcMgrIndexOf](#)
- [@CalcMgrIntegerToString](#)
- [@CalcMgrLastIndexOf](#)
- [@CalcMgrLowercase](#)
- [@CalcMgrMatches](#)
- [@CalcMgrMessageFormat](#)
- [@CalcMgrPadText](#)
- [@CalcMgrUppercase](#)
- [@CalcMgrRemoveQuotes](#)
- [@CalcMgrRemovedQuotes](#)
- [@CalcMgrRemoveSQuotes](#)
- [@CalcMgrReplaceAll](#)
- [@CalcMgrReplaceFirst](#)

- [@CalcMgrSortAndReturn](#)
- [@CalcMgrSortList](#)
- [@CalcMgrSortValues](#)
- [@CalcMgrSplit](#)
- [@CalcMgrSQuote](#)
- [@CalcMgrStartsWith](#)
- [@CalcMgrStringsToString](#)
- [@CalcMgrSubstring](#)
- [@CalcMgrTextLength](#)
- [@CalcMgrTrim](#)

## @CalcMgrCompare

**목적:**

두 문자열을 비교합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.compare (String, String, boolean)
```

CDF 사양: @CalcMgrCompare(text1, text2, ignoreCase)

## @CalcMgrConcat

**목적:**

지정된 문자열을 이 문자열의 끝에 연결

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.concat (String, String)
```

CDF 사양: @CalcMgrConcat(text1, text2)

## @CalcMgrDecimalFormat

**목적:**

지정한 형식 문자열을 사용하여 형식이 지정된 문자열을 반환합니다.

형식 지정은 Java 설명서의 소수 형식을 참조하십시오.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.decimalFormat (String, double)
```

CDF 사양: @CalcMgrDecimalFormat(formatString, value)

## @CalcMgrDoubleFromString

**목적:**

문자열에서 `double`을 변환합니다.

**구문:**

Java

클래스: `com.hyperion.calcmgr.common.cdf.StringFunctions.getDoubleFromString(String)`

CDF 사양: `@CalcMgrDoubleFromString(text)`

## @CalcMgrDoubleToString

**목적:**

`double`을 문자열로 변환합니다.

**구문:**

Java 클래스:

`com.hyperion.calcmgr.common.cdf.MaxLFunctions.doubleToString(double)`

CDF 사양: `@CalcMgrDoubleToString(doubleNumber)`

## @CalcMgrDQuote

**목적:**

텍스트가 큰따옴표로 묶여 있지 않은 경우 텍스트에 큰따옴표를 추가합니다.

**구문:**

Java 클래스: `com.hyperion.calcmgr.common.cdf.StringFunctions.dQuote(String)`

CDF 사양: `@CalcMgrDQuote(text)`

## @CalcMgrEndsWith

**목적:**

이 문자열이 지정된 접미 기호로 끝나는지 테스트

**구문:**

Java 클래스:

`com.hyperion.calcmgr.common.cdf.StringFunctions.endsWith(String, String)`

CDF 사양: `@CalcMgrEndsWith(text, suffix)`

## @CalcMgrFindFirst

**목적:**

지정된 정규 표현식과 일치하는 이 문자열의 첫 번째 하위 문자열을 찾습니다.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.StringFunctions.findFirst(String, String, boolean)`  
CDF 사양: `@CalcMgrFindFirst(text, regExpr, ignoreCase)`

## @CalcMgrFindLast

### 목적:

지정된 정규 표현식과 일치하는 이 문자열의 마지막 하위 문자열을 찾습니다.

### 구문:

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.StringFunctions.findLast(String, String, boolean)`  
CDF 사양: `@CalcMgrFindLast(text, regExpr, ignoreCase)`

## @CalcMgrFormatDouble

### 목적:

지정한 형식 문자열을 사용하여 형식이 지정된 문자열을 반환합니다.

형식 지정은 Java 설명서의 인쇄 형식을 참조하십시오.

### 구문:

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.StringFunctions.formatDouble(String, double)`  
CDF 사양: `@CalcMgrFormatDouble(formatString, value)`

## @CalcMgrGetListCount

### 목적:

목록의 항목 수를 반환합니다.

### 구문:

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.StringFunctions.getListCount(String[])`  
CDF 사양: `@CalcMgrGetListCount(list)`

## @CalcMgrGetListItem

### 목적:

목록의 인덱스 항목을 반환합니다.

### 구문:

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.StringFunctions.getListItem(String[], int)`  
CDF 사양: `@CalcMgrGetListCount(list, index)`

## @CalcMgrIndexOf

**목적:**

지정한 인덱스부터 시작하여 이 문자열 내에서 지정한 하위 문자열이 처음 발생한 위치의 인덱스를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.indexOf(String, String, int)
```

CDF 사양: @CalcMgrIndexOf(text, searchText, begIndex)

**주:**

@CalcMgrIndexOf 또는 @CalcMgrLastIndexOf(아래)에서 인덱스로 -1을 사용하는 경우 전체 문자열이 검색됩니다.

## @CalcMgrIntegerToString

**목적:**

정수를 문자열로 변환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.MaxLFunctions.integerToString(int)
```

CDF 사양: @CalcMgrIntegerToString(integerNumber)

## @CalcMgrLastIndexOf

**목적:**

지정한 인덱스부터 시작하여 뒤로 검색하면서 이 문자열 내에서 지정한 하위 문자열이 마지막 발생한 위치의 인덱스를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.lastIndexOf(String, String, int)
```

CDF 사양: @CalcMgrLastIndexOf(text, searchText, begIndex)

## @CalcMgrLowercase

**목적:**

텍스트를 소문자로 변환합니다.

**구문:**

Java 클래스: `com.hyperion.calcmgr.common.cdf.StringFunctions.toLowerCase(String)`  
CDF 사양: `@CalcMgrLowercase(text)`

## @CalcMgrMatches

### 목적:

이 문자열의 첫 번째 하위 문자열이 지정된 정규 표현식과 일치하는 경우 `true`를 반환합니다.

정규 표현식은 Java 설명서에서 "`java.util.regex.Pattern`"을 참조하십시오.

### 구문:

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.StringFunctions.matches(String,String,boolean)`  
CDF 사양: `@CalcMgrMatches(text, regExpr, ignoreCase)`

## @CalcMgrMessageFormat

### 목적:

지정된 패턴으로 문자열을 생성하고 이 문자열을 사용하여 지정된 인수의 형식을 지정합니다.

### 구문:

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.StringFunctions.messageFormat(String,String[])`  
CDF 사양: `@CalcMgrMessageFormat(text, parameters)`

## @CalcMgrPadText

### 목적:

길이를 채우기 위해 텍스트 앞뒤에 패딩 텍스트로 텍스트를 채웁니다.

예를 들어 `@CalcMgrPadText("01",5,"0",@_true)`는 `01000`을 반환합니다.

`@CalcMgrPadText("01",5,"0",@_false)`는 `00001`을 반환합니다.

### 구문:

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.StringFunctions.padText(String,int,String,boolean)`  
CDF 사양: `@CalcMgrPadText(text,length,padText,append)`

## @CalcMgrUppercase

### 목적:

텍스트를 대문자로 변환합니다.

### 구문:

Java 클래스: `com.hyperion.calcmgr.common.cdf.StringFunctions.toUpper(String)`  
CDF 사양: `@CalcMgrUppercase(text)`



## @CalcMgrRemoveQuotes

**목적:**

텍스트 문자열에서 작은따옴표 또는 큰따옴표를 제거합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.removeQuotes (String)
```

CDF 사양: @CalcMgrRemoveQuotes (text)

## @CalcMgrRemoveDQuotes

**목적:**

텍스트 문자열에서 큰따옴표를 제거합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.removeDQuotes (String)
```

CDF 사양: CDF 사양: @CalcMgrRemoveDQuotes (text)

## @CalcMgrRemoveSQuotes

**목적:**

텍스트 문자열에서 작은따옴표를 제거합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.removeSQuotes (String)
```

CDF 사양: @CalcMgrRemoveSQuotes (text)

## @CalcMgrReplaceAll

**목적:**

지정된 정규 표현식과 일치하는 이 문자열의 각 하위 문자열을 지정된 대체 항목으로 대체.

정규 표현식은 `java.util.regex.Pattern`에 대한 Java 설명서를 참조하십시오.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.replaceAll (String,String,String)
```

CDF 사양: @CalcMgrReplaceAll (text, regExpr, replacement)

## @CalcMgrReplaceFirst

**목적:**

지정된 정규 표현식과 일치하는 이 문자열의 첫 번째 하위 문자열을 지정된 대체 항목으로 대체.

정규 표현식은 `java.util.regex.Pattern`에 대한 Java 설명서를 참조하십시오.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.replaceFirst(String,String,String)
```

CDF 사양: `@CalcMgrReplaceFirst(text, regExpr, replacement)`

## @CalcMgrSortAndReturn

**목적:**

값에 따라 목록에 있는 항목을 정렬하고 최상위 n개 요소를 반환합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.sortAndReturn(String[],double[],int,boolean)
```

CDF 사양: `@CalcMgrSortAndReturn(list, values, topN, sortAscending)`

## @CalcMgrSortList

**목적:**

목록에 있는 항목을 정렬합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.sortList(String[],boolean,boolean)
```

CDF 사양: `@CalcMgrSortList(list, caseSensitive, sortAscending)`

## @CalcMgrSortValues

**목적:**

값에 따라 목록에 있는 항목을 정렬합니다.

**구문:**

Java 클래스:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.sortValues(double[],boolean)
```

CDF 사양: `@CalcMgrSortValues(values, sortAscending)`

## @CalcMgrSplit

**목적:**

`regex`에 따라 텍스트를 분할합니다.

정규 표현식은 `java.util.regex.Pattern`에 대한 Java 설명서를 참조하십시오.

**구문:**

Java 클래스:  
`com.hyperion.calcmgr.common.cdf.StringFunctions.split(String, String)`  
CDF 사양: `@CalcMgrSplit(text, regex)`

## @CalcMgrSQuote

**목적:**  
텍스트가 작은따옴표로 묶여 있지 않은 경우 텍스트에 작은따옴표를 추가합니다.

**구문:**  
Java 클래스: `com.hyperion.calcmgr.common.cdf.StringFunctions.sQuote(String)`  
CDF 사양: `@CalcMgrSQuote(text)`

## @CalcMgrStartsWith

**목적:**  
이 문자열이 지정된 접두 기호로 시작하는지 테스트

**구문:**  
Java 클래스:  
`com.hyperion.calcmgr.common.cdf.StringFunctions.startsWith(String, String)`  
CDF 사양: `@CalcMgrStartsWith(text, prefix)`

## @CalcMgrStringsToString

**목적:**  
문자열 배열을 구분자를 사용하는 문자열로 변환합니다.

**구문:**  
Java 클래스:  
`com.hyperion.calcmgr.common.cdf.MaxLFunctions.stringsToString(String[], String)`  
CDF 사양: `@CalcMgrStringsToString(strings, seperator)`

## @CalcMgrSubstring

**목적:**  
이 문자열의 하위 문자열인 새 문자열을 반환합니다.  
하위 문자열은 `startIndex`에서 시작하고 `index endIndex - 1`의 문자까지 확장됩니다.  
따라서 하위 문자열의 길이는 `endIndex-startIndex`입니다.

`endIndex`가 0 미만인 경우 `endIndex`가 마지막 문자의 인덱스가 됩니다.

**구문:**  
Java 클래스:  
`com.hyperion.calcmgr.common.cdf.StringFunctions.substring(String, int, int)`

CDF 사양: @CalcMgrSubstring(text, startIndex, endIndex)

## @CalcMgrTextLength

### 목적:

텍스트의 길이를 반환합니다.

### 구문:

Java 클래스: com.hyperion.calcmgr.common.cdf.StringFunctions.length(String)

CDF 사양: @CalcMgrTextLength(text)

## @CalcMgrTrim

### 목적:

텍스트 문자열에서 선행 및 후행 공백을 제거합니다.

### 구문:

Java 클래스: com.hyperion.calcmgr.common.cdf.StringFunctions.trim(String)

CDF 사양: @CalcMgrTrim(text)

# 구성요소에 함수 삽입

선택할 수 있는 함수 유형은 작업 중인 애플리케이션 유형 및 구성요소 유형에 따라 다릅니다.

Oracle Hyperion Financial Management, Oracle Hyperion Planning 또는 Oracle Essbase 블록 저장영역 애플리케이션으로 작업하는 경우 공식, 스크립트, 조건 및 멤버 범위 구성요소에 함수를 삽입할 수 있습니다.

Oracle General Ledger 및 Planning에서 사용된 Essbase 집계 저장영역 애플리케이션을 포함하여 Essbase 집계 저장영역 애플리케이션으로 작업하는 경우 POV 구성요소에만 함수를 삽입할 수 있습니다. Essbase 합산 저장영역 애플리케이션에서는 한 가지 함수 유형, 즉 멤버 세트 함수 유형만 선택할 수 있습니다. 이 항목의 두 번째 절차를 참조하십시오.

공식, 스크립트, 조건 및 멤버 범위 구성요소에 함수를 삽입하려면 다음을 수행합니다.

- 다음 태스크 중 하나를 수행합니다.
  - 함수를 삽입하려는 구성요소가 들어 있는 비즈니스 규칙을 엽니다. 그런 후 비즈니스 규칙의 순서도에서 함수를 삽입하려는 구성요소를 선택합니다.
  - 함수를 삽입하려는 구성요소가 들어 있는 템플릿을 엽니다. 그런 후 템플릿의 순서도에서 함수를 삽입하려는 구성요소를 선택합니다.
  - 함수를 삽입할 공식 구성요소를 엽니다.
  - 함수를 삽입할 스크립트 구성요소를 엽니다.
- 다음 태스크 중 하나를 수행합니다.
  - 공식 구성요소에 함수를 삽입하려면 공식 탭에서 공식 행을 누르고 작업 아이콘, 함수 순으로 선택합니다.
  - 함수를 스크립트 구성요소에 삽입하려면 스크립트 탭에서 함수와 해당 매개변수 삽입 아이콘을 누릅니다.

- 멤버 범위 구성요소에 함수를 삽입하려면 **멤버 범위** 탭에서 차원의 **값** 열을 누르고 **작업** 아이콘, **함수** 순으로 선택합니다.
  - 함수를 조건 구성요소에 삽입하려면 **조건** 탭에서 **조건 빌더** 아이콘을 누릅니다. 조건 빌더의 **공식**, **함수** 또는 **값**에서 **작업** 아이콘, **함수** 순으로 선택합니다.
3. **함수 선택기**에서 다음 태스크 중 하나를 수행합니다.

 **주:**

함수 선택기를 연 컨텍스트에 따라, 사용 가능한 함수 유형이 **함수 작업**에 설명된 함수 유형보다 더 제한될 수 있습니다.

- **범주**에서 함수 유형을 선택할 수 있는 경우 함수 유형을 하나 선택하거나 **모든 함수**를 선택하여 모든 함수 유형의 함수를 표시합니다. 해당 범주에 대한 함수 또는 모든 함수가 범주 선택 항목 아래 목록에 표시됩니다.
  - **범주**의 함수 유형 중에서 선택할 수 없으면 4에서 계속합니다.
4. 함수 목록에서 함수를 선택합니다.
5. 함수에 대한 매개변수를 입력합니다. Oracle Fusion General Ledger 애플리케이션에서 사용된 항목을 포함하여 Planning 및 Essbase 구성요소에 사용할 수 있는 함수 및 매개변수에 대한 설명은 *Oracle Essbase Technical Reference*를 참조하십시오. Financial Management 구성요소에 사용할 수 있는 함수 및 매개변수에 대한 설명을 보려면 *Oracle Hyperion Financial Management 관리자 가이드*를 참조하십시오.
6. **확인**을 누릅니다.

POV 구성요소에 함수를 삽입하려면 다음을 수행합니다.

1. 시스템 뷰에서 Essbase 애플리케이션 유형, 애플리케이션, 데이터베이스, **규칙** 순으로 확장합니다.
2. 함수를 삽입하려는 POV 구성요소가 들어 있는 비즈니스 규칙을 엽니다. 그런 후 비즈니스 규칙의 순서도에서 함수를 삽입하려는 POV 구성요소를 선택합니다.
3. 함수를 POV 구성요소에 삽입하려면 **POV** 탭에서 차원에 대한 **값** 열을 누르고 **작업** 아이콘, **함수** 순으로 선택합니다.
4. **함수 선택기**의 함수 목록에서 함수를 선택합니다.
  - @Attribute
  - @Level0Descendants
  - @Siblings
  - @UDA

 **주:**

Oracle General Ledger에서 사용되는 항목을 포함하여 Essbase 합산 저장영역 애플리케이션에서 작업 중인 경우 멤버 세트 함수만 사용할 수 있습니다.

5. 함수에 대한 매개변수를 입력합니다. 함수 및 매개변수에 대한 설명을 보려면 *Oracle Essbase Technical Reference*를 참조하십시오.
6. 확인을 누릅니다.

## 스마트 목록 작업

스마트 목록은 사용자가 Oracle Hyperion Planning 애플리케이션의 Planning 데이터 양식 셀에서 액세스하는 사용자정의 드롭다운 목록입니다.

데이터 양식 셀을 눌러 데이터를 입력할 때 셀에 직접 입력하는 대신 드롭다운 목록에서 항목을 선택할 수 있습니다. 스마트 목록이 포함된 셀에는 입력할 수 없습니다.

Oracle Hyperion Calculation Manager에서 변수 또는 공식, 스크립트, 조건 및 멤버 범위 구성요소에 스마트 목록을 삽입할 수 있습니다.

## 스마트 목록 삽입

Oracle Hyperion Planning 비즈니스 규칙, 공식 구성요소 또는 스크립트 구성요소에서 스마트 목록을 사용합니다.

스마트 목록은 Planning 데이터 양식에서 Planning 관리자가 지정하는 특정 데이터 셀 내에 사용할 수 있습니다. 스마트 목록은 사용자가 선택할 수 있는 옵션을 포함하는 사용자정의 드롭다운 목록입니다.

스마트 목록을 삽입하려면 다음을 수행합니다.

1. 다음 태스크 중 하나를 수행합니다.
  - 스마트 목록을 삽입하려는 구성요소가 있는 비즈니스 규칙을 엽니다. 그런 후 비즈니스 규칙의 순서도에서 스마트 목록을 삽입하려는 공식 또는 스크립트 구성요소를 선택합니다.
  - 스마트 목록을 삽입할 공식 구성요소를 엽니다.
  - 스마트 목록을 삽입할 스크립트 구성요소를 엽니다.
2. 다음 태스크 중 하나를 수행합니다.
  - 스마트 목록을 비즈니스 규칙에 삽입하려면 스크립트 탭에서 **스마트 목록 삽입** 아이콘을 누릅니다.
  - 공식 구성요소에 스마트 목록을 삽입하려면 공식 탭에서 **공식 행**을 누르고 **작업** 아이콘, **스마트 목록** 순으로 선택합니다.
  - 스마트 목록을 스크립트 구성요소에 삽입하려면 스크립트 탭에서 **스마트 목록 삽입** 아이콘을 누릅니다.
3. **저장**을 누릅니다.

## Planning 공식 표현식 작업

Oracle Hyperion Calculation Manager 그래픽 또는 스크립트 규칙에서 Oracle Hyperion Planning 공식 표현식을 사용합니다.

다음 유형의 공식 표현식을 사용할 수 있습니다.

- [스마트 목록](#)
- [차원\(둘 이상\)](#)

- Planning 사용자 변수
- 기간
- 시나리오
- 상호 참조
- Workforce 큐브 연 누계
- 문자열의 ID 가져오기

## 스마트 목록

예를 들어 "Product Channel"=[[Channel.Retail]]과 같이 공식 표현식에 스마트 목록을 변수로 포함할 수 있습니다.

"Product Channel"은 "스마트 목록" 유형인 계정이고, Channel은 스마트 목록 이름이고, Retail은 스마트 목록 항목입니다. Retail의 스마트 목록 ID가 2인 경우 멤버 공식에서 Channel.Retail이 2로 바뀝니다(애플리케이션은 스마트 목록을 숫자로 처리함). Retail의 스마트 목록 ID가 2인 경우 2를 넣어 계산하고 2를 데이터베이스에 저장합니다.

### Calculation Manager 구문:

```
[[SLName.entryname]]
```

#### 예:

다음 구문의 경우

```
FIX (Mar, Actual, Working, FY15, P_000, "111")
  "Product Channel" =[[Channel.Retail]] ;
ENDFIX
```

위 구문은 다음 스크립트를 반환합니다.

```
FIX (Mar, Actual, Working, FY15, P_000, "111")
  "Product Channel" =2 ;
ENDFIX
```

## 차원(둘 이상)

Dimension(dimTag)은 사전 정의된 차원의 이름을 반환합니다.

dimtag는 다음과 같습니다.

- DIM\_NAME\_PERIOD
- DIM\_NAME\_YEAR
- DIM\_NAME\_ACCOUNT
- DIM\_NAME\_ENTITY
- DIM\_NAME\_SCENARIO
- DIM\_NAME\_VERSION
- DIM\_NAME\_CURRENCY

**Calculation Manager 구문:**

```
[[Dimension("DIM_NAME_ENTITY")]]
```

**예:**

```
CALC DIM([[Dimension("DIM_NAME_ENTITY")]]);
```

이 애플리케이션에서는 Entity의 이름이 *엔티티*로 지정되므로 위 스크립트는 다음을 반환합니다.

```
CALC DIM ("Entity");
```

엔티티 차원의 이름이 *비용 센터*로 지정된 경우에는 다음이 반환됩니다.

```
CALC DIM ("Cost Center");
```

## Planning 사용자 변수

Planning 사용자 변수는 사용자 변수의 멤버를 반환합니다.

**Calculation Manager 구문:**

```
[[PlanningFunctions.getUserVarValue("xyz")]]
```

**예:**

```
FIX (Feb, Actual, Working, P_000,
[[PlanningFunctions.getUserVarValue("Entity View")]] )
"5800" = 40;
ENDFIX
```

이 애플리케이션에는 *엔티티 뷰*라는 Planning 사용자 변수가 있습니다. 이 사용자의 경우 112로 설정되어 있습니다. 따라서 위 예에서 스크립트는 다음을 반환합니다.

```
FIX (Feb, Actual, Working, P_000, "112")
"5800" = 40;
ENDFIX
```

## 기간

**참조:**

- [Period\(periodName\)](#)
- [NumberofPeriodsinYear](#) 및 [NumberofYears](#)

### Period(periodName)

Period(periodName)는 지정된 기간을 반환합니다.

기간 이름 옵션은 다음과 같습니다.

- FIRST\_QTR\_PERIOD
- SECOND\_QTR\_PERIOD



- THIRD\_QTR\_PERIOD
- FOURTH\_QTR\_PERIOD
- FIRST\_PERIOD
- LAST\_PERIOD

**Calculation Manager 구문:**

```
[[Period("FIRST_QTR_PERIOD")]]
```

**예:**

다음 구문의 경우

```
FIX ( Mar, Actual, Working, P_000, "6100", FY15 )
    "120" = [[Period("FIRST_QTR_PERIOD")]];
ENDFIX
```

위 구문은 다음 스크립트를 반환합니다.

```
FIX (Mar, Actual, Working, P_000, "6100", FY15)
    "120" = "Mar";
ENDFIX
```

## NumberOfPeriodsinYear 및 NumberOfYears

NumberOfPeriodsinYear는 연도의 기간 수를 반환하고, NumberOfYears는 애플리케이션의 연수를 반환합니다.

**Calculation Manager 구문:**

```
[[NumberOfPeriodsinYear]]
```

```
[[NumberOfYears]]
```

**예:**

다음 구문의 경우

```
FIX (Mar, Actual, Working, P_000, "6100", FY15)
    "120"=[[NumberOfPeriodsinYear]];
    "120"=[[NumberOfYears]];
ENDFIX
```

위 구문은 다음 스크립트를 반환합니다.

```
FIX (Mar, Actual, Working, P_000, "6100", FY15)
    "120"=12;
    "120"=9;
ENDFIX
```

## 시나리오

SPR(Scenario Planning Range) 시간 기간과 모듈 계획 및 예측 준비 구성 태스크 정보를 다음과 같이 지정할 수 있는 표현식으로 표시합니다.

- **시작 연도:** 지정된 시나리오의 시작 연도를 문자열 형식으로 반환합니다.
- **종료 연도:** 지정된 시나리오의 종료 연도를 문자열 형식으로 반환합니다.
- **시작 월:** 지정된 시나리오의 시작 월을 문자열 형식으로 반환합니다.
- **종료 월:** 지정된 시나리오의 종료 월을 문자열 형식으로 반환합니다.
- **모듈 시작 연도:** 지정된 모듈 및 시나리오의 시작 연도를 문자열 형식으로 반환합니다.
- **모듈 종료 연도:** 지정된 모듈 및 시나리오의 종료 연도를 문자열 형식으로 반환합니다.
- **모듈 시작 기간:** 지정된 모듈 및 시나리오의 시작 기간을 문자열로 반환합니다.
- **모듈 종료 기간:** 지정된 모듈 및 시나리오의 종료 기간을 문자열 형식으로 반환합니다.
- **모듈 계획 시작 연도:** 현재 회계 연도인 경우 정수 값 "1", 다음 회계 연도인 경우 "0"을 반환합니다.

### 주:

모듈 시작 연도, 모듈 종료 연도, 모듈 시작 기간, 모듈 종료 기간 및 모듈 계획 시작 연도는 Planning 모듈 및 Oracle Strategic Workforce Planning Cloud 애플리케이션에만 적합하며 "PlanningFunctions" 구문이 필요합니다.

### Calculation Manager 구문:

```
[[getStartYear("ScenarioName")]]
[[getEndYear("ScenarioName")]]
[[getStartMonth("ScenarioName")]]
[[getEndMonth("ScenarioName")]]
[[PlanningFunctions.getModuleStartYear("ModuleName","ScenarioName")]]
[[PlanningFunctions.getModuleEndYear("ModuleName","ScenarioName")]]
[[PlanningFunctions.getModuleStartPeriod("ModuleName","ScenarioName")]]
[[PlanningFunctions.getModuleEndPeriod("ModuleName","ScenarioName")]]
[[PlanningFunctions.isPlanStartYearSameAsCurrentFiscalYear("ModuleName")]]
```

### 시나리오 이름

ScenarioName은 입력된 시나리오 멤버 또는 Oracle Hyperion Calculation Manager 런타임 프롬프트 멤버 형식 변수일 수 있습니다. 멤버는 큰 따옴표로 묶어야 합니다. 예:  
[[getStartYear("Actual")]].

ScenarioName은 다음 형식의 대체 변수일 수도 있습니다.

- `getSubVarValue("CubeName", "SubstitutionVariableName")`는 CubeName을 통해 정의된 큐브 레벨의 지정된 대체 변수에 대한 대체 변수 값을 반환합니다.
- `getSubVarValue("SubstitutionVariableName")`는 모든 큐브의 애플리케이션 레벨에서 지정된 대체 변수에 대한 대체 변수 값을 반환합니다.

CubeName 및 SubstitutionVariableName은 큰따옴표로 묶어야 하며 & 또는 {}를 사용하지 마십시오. 아래 예 3 및 예 4를 참조하십시오.

### 모듈 이름

ModuleName은 Oracle Hyperion Planning 및 Strategic Workforce Planning의 현재 모듈에 대한 사전 정의된 이름이어야 합니다. ModuleName을 매개변수로 사용하는 이러한 표현식은 ScenarioName의 계획 및 예측 준비 구성 태스크 시간 대상 기간 정보를 사용합니다. 해당 시간 대상 기간은 계획 및 예측 시나리오에 대한 하나의 애플리케이션 내에서 모듈별로 다를 수 있습니다.

계획 및 예측 이외의 다른 시나리오도 이러한 표현식에서 참조될 수 있지만, 차원 편집기에서 SPR(Scenario Planning Range)에 대해 설정된 시간 기간만 반환합니다. 이 시간 기간은 해당 애플리케이션 내 모든 모듈에 대해 동일합니다. SPR(Scenario Planning Range)에 대한 자세한 내용은 [시나리오 정보](#)를 참조하십시오.

계획 시작 연도는 모듈에 대한 계획 시나리오와 관련된 다른 계획 및 예측 준비 구성 태스크입니다. 이 표현식은 해당 설정에 상대적인 값을 반환하므로 *현재 회계 연도*로 설정된 경우 정수 값 "1"을 반환하고 *다음 회계 연도*로 설정된 경우 "0"을 반환합니다. 이 표현식은 차원 멤버를 값으로 반환하지 않으므로 조건부 명령문 컨텍스트 내에서만 사용할 수 있습니다. 아래 예 5를 참조하십시오.

ModuleName에 적합한 값은 다음과 같으며 특정 비즈니스 프로세스에 적용할 수 있습니다.

**표 8-8 적합한 ModuleName 값**

비즈니스 프로세스	적합한 ModuleName 값
Planning 모듈	<ul style="list-style-type: none"> <li>• "자본"</li> <li>• "재무"</li> <li>• "프로젝트"</li> <li>• "인력"</li> </ul>
Strategic Workforce Planning	<ul style="list-style-type: none"> <li>• "전략적 인력"</li> <li>• "인력"</li> </ul>



#### 주:

ModuleName 값은 큰따옴표로 묶어야 하며 대소문자를 구분하지 않습니다.

### 예 1

다음은 {rtpScenario}가 *멤버* 유형의 실행 시간 프롬프트 변수이며 기본값은 *actual*인 구문입니다.

```
FIX({rtpScenario}, [[getStartYear({rtpScenario})]]:
[[getEndYear({rtpScenario})]],
[[getStartMonth({rtpScenario})]]:[[getEndMonth({rtpScenario})]]):
    FIX ( Working, P_000, "111")
        "5800" = 5500;
    ENDFIX
ENDFIX
```

위 구문은 다음 스크립트를 반환합니다.

```
FIX ("Actual", "FY10" : "FY18", "Jan" : "Dec")
  FIX (Working, P_000, "111")
    "5800" = 5500;
  ENDFIX
ENDFIX
```

## 예 2

다음 구문의 경우

```
FIX({rtpScenario}, [[PlanningFunctions.getModuleStartYear("CAPITAL",
{rtpScenario})]]:
[[PlanningFunctions.getModuleEndYear("CAPITAL",{rtpScenario})]] , "Jan" :
"Dec")
  FIX(OEP_Working, P_000, "111")
    "5800" = 5500;
  ENDFIX
ENDFIX
```

다음 스크립트를 반환합니다. 여기서 자본 모듈 계획 및 예측 준비 구성 태스크는 계획 시나리오에 대해 FY18 및 FY22 각각의 시작 및 종료 연도를 설정합니다.

```
FIX("OEP_Plan", "FY18" : "FY22", "Jan" : "Dec")
  FIX(OEP_Working, P_000, "111")
    "5800" = 5500;
  ENDFIX
ENDFIX
```

## 예 3

다음 구문에서는 큐브 레벨의 대체 변수를 사용합니다. 이 예에서 Plan1은 큐브 이름이고 CurrentMonth는 대체 변수 이름입니다.

```
FIX ("OEP_Plan", [[getStartMonth(getSubVarValue("Plan1", "CurrentMonth"))]],
"FY15",
"BU Version_1", "No Currency", "No Entity", "No Grades")
"Current" = 15;
ENDFIX
```

이 구문은 다음 스크립트를 생성합니다.

```
FIX ("OEP_Plan", "Jan", "FY15", "BU Version_1", "No Currency", "No Entity",
"No
Grades")
"Current" = 15;
ENDFIX
```

**예 4**

다음 구문에서는 애플리케이션 레벨의 대체 변수를 사용합니다. 이 예에서 CurrentMonth는 대체 변수 이름입니다.

```
FIX ("OEP_Plan", [[getStartMonth(getSubVarValue("CurrentMonth"))]],
"FY15", "BU
Version_1", "No Currency", "No Entity", "No Grades")
"Current" = 15;
ENDFIX
```

이 구문은 다음 스크립트를 생성합니다.

```
FIX ("OEP_Plan", "Jan", "FY15", "BU Version_1", "No Currency", "No
Entity", "No
Grades")
"Current" = 15;
ENDFIX
```

**예 5**

다음 구문의 경우

```
FIX("OEP_Plan","OEP_Working",FY20:FY24)
"Bonus"
(
  IF([[PlanningFunctions.isPlanStartYearSameAsCurrentFiscalYear
("Workspace")]] == 1)
    "Bonus" = "Salary" * 0.2;
  ELSEIF([[PlanningFunctions.isPlanStartYearSameAsCurrentFiscalYear
("Workforce")]] == 0)
    "Bonus" = "Salary" * 0.3;
)
ENDFIX
```

다음 스크립트를 생성합니다. 여기서 인력 모듈 계획 및 예측 준비 구성 태스크에 계획 시작 연도가 *다음 회계 연도*로 설정되어 있어 isPlanStartYearSameAsCurrentFiscalYear가 *다음 회계 연도*로 설정된 경우 정수 "0"을 반환하므로 첫번째 조건 테스트(IF)는 실패하여 실행되지 않고 두번째 조건 테스트(ELSEIF)는 통과하여 실행됩니다.

```
FIX("OEP_Plan","OEP_Working", FY20:FY24)
"Bonus"
(
  IF(0 == 1)
    "Bonus" = "Salary" * 0.2;
  ELSEIF(0 == 0)
    "Bonus" = "Salary" * 0.3;
)
ENDFIX
```

## 상호 참조

### 참조:

- [CrossRef\(accountName\)](#)
- [CrossRef\(accountName, prefix\)](#)
- [CrossRef\(accountName, prefix, true\)](#)

## CrossRef(accountName)

`CrossRef(accountName)`는 각 차원 이름(통화, 기간 및 연도 제외)에 기본 접두어 No와 지정된 계정을 추가하여 상호 참조를 생성합니다.

### Calculation Manager 구문:

```
[[CrossRef(accountName)]]
```

### 예:

애플리케이션에 계정, 기간, HSP\_View, 연도, 시나리오, 버전, 엔티티 및 제품 차원이 있다고 가정합니다. 이 예에서

```
FIX (Aug, Actual, Working, FY15, P_000, "112")
    "120" = [[CrossRef("5800")]];
ENDFIX
```

위 구문은 다음 스크립트를 반환합니다.

```
FIX (Aug, Actual, Working, FY15, P_000, "112")
    "120" = "BegBalance"->"No HSP_View"->"No Scenario"->"No Version"->"No
Entity"->"No Product"->"5800";
ENDFIX
```

## CrossRef(accountName, prefix)

`CrossRef(accountName, prefix)`는 각 차원 이름(통화, 기간 및 연도 제외)에 지정된 접두어와 지정된 계정을 추가하여 상호 참조를 생성합니다. 접두어는 큰 따옴표로 묶어야 합니다.

### Calculation Manager 구문:

```
[[CrossRef(accountName, "prefix")]]
```

### 예:

애플리케이션에 계정, 기간, HSP\_View, 연도, 시나리오, 버전, 엔티티 및 제품 차원이 있다고 가정합니다. 이 예에서

```
FIX (Aug, Actual, Working, FY15, P_000, "112")
    "110" = [[CrossRef("5800", "No")]];
ENDFIX
```

위 구문은 다음 스크립트를 반환합니다.

```
FIX (Aug, Actual, Working, FY15, P_000, "112")
  "110" = "BegBalance"->"No HSP_View"->"No Scenario"->"No Version"-
>"No Entity"->"No Product"->"5800";
ENDFIX
```

## CrossRef(accountName, prefix, true)

CrossRef(accountName, prefix, true)는 각 차원 이름에 지정된 접두어와 지정된 계정을 추가하여 상호 참조를 생성합니다. (여기에는 연도가 포함되지만 통화 및 기간은 포함되지 않습니다.) 접두어는 큰 따옴표로 묶어야 합니다.

### Calculation Manager 구문:

```
[[CrossRef(accountName, "prefix", true)]]
```

### 예:

애플리케이션에 계정, 기간, HSP\_View, 연도, 시나리오, 버전, 엔티티 및 제품 차원이 있다고 가정합니다. 이 예에서

```
FIX (Aug, Actual, Working, FY15, P_000, "112")
  "111" = [[CrossRef("5800", "NoX", true)]];
ENDFIX
```

위 구문은 다음 스크립트를 반환합니다.

```
FIX (Aug, Actual, Working, FY15, P_000, "112")
  "111" = "BegBalance"->"NoXHSP_View"->"NoXYear"->"NoXScenario"-
>"NoXVersion"->"NoXEntity"->"NoXProduct"->"5800";
ENDFIX
```

## Workforce 큐브 연 누계

### 참조:

- [CYTD\(memberName\)](#)
- [CYTD\(memberName, calTpIndexName, fiscalTpIndexName\)](#)

## CYTD(memberName)

### 주:

CYTD(memberName)는 **workforce** 큐브에만 해당합니다.

CYTD(memberName)는 멤버에 대한 달력 연 누계 공식을 생성합니다.

**Calculation Manager 구문:**

```
[[CYTD(memberName)]]
```

**예:**

```
Fix (NOV, Actual, Working, FY15, P_000, "112")
    "5800" = [[CYTD("6100")]];
ENDFIX
```

## CYTD(memberName, calTpIndexName, fiscalTPIndexName)

**주:**

CYTD(memberName, calTpIndexName, fiscalTPIndexName)는 **workforce** 큐브에만 해당합니다.

CYTD(memberName, calTpIndexName, fiscalTPIndexName)는 멤버에 대한 달력 연 누계 공식을 생성하고, 달력의 연도와 회계 연도를 기준으로 기간 인덱스를 생성합니다. 멤버의 이름이 바뀔 때 사용합니다. 기본 멤버 이름은 "Cal TP-Index" 및 "Fiscal TPIndex"입니다.

**Calculation Manager 구문:**

```
[[CYTD(accountName, "Cal TP-Index", "Fiscal TPIndex")]]
```

**예:**

```
Fix (Dec, Actual, Working, FY15, P_000, "112")
    "5800" = [[CYTD("6100", "Cal TP-Index", "Fiscal TPIndex")]];
ENDFIX
```

## 문자열의 ID 가져오기

Oracle Hyperion Planning에서 계정 유형이 **텍스트**인 경우 Oracle Hyperion Calculation Manager에서 공식을 작성하여 텍스트 값을 지정할 수 있습니다.

**Calculation Manager 구문:**

```
[[PlanningFunctions.getIdForString("text")]]
```

**예:**

Planning에 **텍스트** 유형인 "acct1 text" 계정이 있습니다. FY16 Dec의 값을 FY17 Mar에 복사하고 텍스트 계정을 "예산 편성되지 않음"으로 변경하려고 합니다.

```
FIX (Actual, Working, P_000, "210")
    DATACOPY FY16->Dec TO FY17->Mar;
    Mar("acct1 text"->FY17 = [[PlanningFunctions.getIdForString("Not
    Budgeted")]]);
ENDFIX
```



# 9

## 검증 및 배포

### 참조:

- [시스템 뷰에서 비즈니스 규칙, 비즈니스 규칙 세트, 공식 및 스크립트 구성요소 검증](#)  
애플리케이션에 배포하기 전에 비즈니스 규칙, 규칙 세트, 공식 및 스크립트 구성요소를 검증하여 구문이 올바른지 확인합니다.
- [규칙 디자이너에서 비즈니스 규칙 검증](#)  
비즈니스 규칙을 생성하거나 디버그할 때 규칙 디자이너에서 비즈니스 규칙을 검증할 수 있습니다.
- [비즈니스 규칙 및 비즈니스 규칙 세트 배포](#)  
비즈니스 규칙 및 비즈니스 규칙 세트를 배포합니다.

## 시스템 뷰에서 비즈니스 규칙, 비즈니스 규칙 세트, 공식 및 스크립트 구성요소 검증

애플리케이션에 배포하기 전에 비즈니스 규칙, 규칙 세트, 공식 및 스크립트 구성요소를 검증하여 구문이 올바른지 확인합니다.

검증 프로세스는 다음을 확인합니다.

- 애플리케이션에 있는 차원의 모든 차원 멤버가 적합합니다.
- 모든 함수가 존재하며 정확한 수의 매개변수를 가지며 해당 애플리케이션 유형에 적합합니다.
- 비즈니스 규칙의 모든 변수 참조가 적합합니다. 대체 변수의 경우, 변수는 먼저 정확한 문자열로 바뀐 후 검증됩니다. 실행 변수의 경우 검증 프로세스는 해당 애플리케이션, 애플리케이션 유형 내의 애플리케이션, 계산 유형(Oracle Hyperion Financial Management 사용자의 경우), 계획 유형(Oracle Hyperion Planning 사용자) 또는 데이터베이스(Oracle Essbase 및 Oracle General Ledger 사용자) 및/또는 비즈니스 규칙(Planning 및 Essbase 사용자만 해당)에 대해 변수가 정의되었는지 확인합니다.
- 스크립트 생성에 구문 오류가 없습니다.

**Planning 사용자만 해당:** 기본값을 갖는 런타임 프롬프트를 포함하는 비즈니스 규칙을 검증하는 경우 검증 프로세스는 런타임 프롬프트 내의 모든 멤버가 선택된 계획 유형 및 애플리케이션에 대해 적합하며, 구문 또는 의미적 오류가 없는지 확인합니다. 기본값을 갖지 않는 런타임 프롬프트를 포함하는 비즈니스 규칙을 검증하는 경우 검증이 수행되지 않습니다.

### 주:

배포하기 전에 규칙 및 규칙 세트를 검증하지 않으면 성공적으로 배포할 수는 있지만 규칙 및 규칙 세트를 실행하지 못할 수 있습니다.

비즈니스 규칙, 규칙 세트, 공식 또는 스크립트 구성요소를 검증하려면 다음을 수행합니다.

1. 다음 태스크 중 하나를 수행합니다.

- **Financial Management 사용자:** 규칙 세트를 검증하려면 계산 유형 및 규칙 세트를 확장합니다.
- 규칙 세트를 검증하려면 **규칙 세트**를 확장합니다.  
Planning 및 Essbase 사용자

 **주:**

Financial Management 애플리케이션의 경우 애플리케이션 내의 각 계산 유형에 대해 [규칙 세트] 노드가 있지만, Planning 애플리케이션의 경우 계획 유형 및 데이터베이스와 동일한 레벨에 있는 각 애플리케이션에 대해 하나의 규칙 세트 노드만 있습니다.

- 규칙, 공식, 스크립트 또는 템플릿을 검증하려면 계산 유형, 계획 유형 또는 데이터베이스를 확장한 다음 검증할 객체에 따라 **규칙, 공식, 스크립트 또는 템플릿**를 확장합니다.
2. 다음 태스크 중 하나를 수행합니다.
    - 검증하려는 객체를 마우스 오른쪽 버튼으로 누르고 **검증**을 선택합니다.
    - 검증하려는 객체를 선택하고 **작업, 검증**을 선택합니다.
  3. **Planning 및 Essbase 사용자만 해당:** 검증 옵션에서 Planning(Planning 규칙의 경우)에 대해 검증할 것인지 Essbase(Essbase 규칙의 경우)에 대해 검증할 것인지 둘 다에 대해 검증할 것인지 선택합니다. 그런 다음 **확인**을 누릅니다.
  4. 다음 태스크 중 하나를 수행합니다.
    - a. 객체가 성공적으로 검증되면 **확인**을 누릅니다.
    - b. 오류가 있으면 표시됩니다. 해당 오류를 수정하고 객체를 다시 검증합니다.

## 규칙 디자이너에서 비즈니스 규칙 검증

비즈니스 규칙을 생성하거나 디버그할 때 규칙 디자이너에서 비즈니스 규칙을 검증할 수 있습니다.

- Financial Management 애플리케이션 관리에서 생성된 Oracle Hyperion Financial Management 비즈니스 규칙은 Financial Management에 대해서만 검증합니다.
- Planning 애플리케이션 관리에서 생성된 Oracle Hyperion Planning 비즈니스 규칙은 Planning에 대해서만 검증합니다.
- Essbase 애플리케이션의 Oracle Essbase 규칙은 Essbase에 대해서만 검증합니다.
- Oracle General Ledger 애플리케이션의 Oracle General Ledger 규칙은 Essbase에 대해서만 검증합니다.

규칙 디자이너에서 비즈니스 규칙을 검증하려면 다음을 수행합니다.

1. 시스템 뷰에서 애플리케이션 유형, 애플리케이션, 계산 유형, 계획 유형 또는 데이터베이스, **규칙** 순으로 확장합니다.
2. 검증할 규칙을 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
3. 규칙 디자이너에서 **작업, 검증** 순으로 선택하고 다음 태스크 중 하나를 수행합니다.
  - Financial Management 규칙으로 작업하는 경우 **Consolidation**을 선택합니다.

- Planning 규칙으로 작업하는 경우 **Planning**을 선택합니다.

 **주:**

Financial Management 또는 Planning 애플리케이션 관리로 생성된 애플리케이션에서 비즈니스 규칙으로 작업 중인 경우 규칙이 생성된 애플리케이션에 대해서만 검증할 수 있습니다.

- Essbase 애플리케이션과 규칙(Oracle General Ledger 애플리케이션에 대해 생성된 항목 포함)으로 작업 중인 경우 **Essbase**를 선택합니다.
- Planning 애플리케이션 관리에서 생성된 애플리케이션 및 Planning 규칙으로 작업하는 경우 **Planning**을 선택합니다.

규칙을 성공적으로 검증한 경우 검증에 성공했음을 알려 주는 메시지가 표시됩니다. 규칙이 검증되지 않으면 오류가 표시됩니다.

## 비즈니스 규칙 및 비즈니스 규칙 세트 배포

비즈니스 규칙 및 비즈니스 규칙 세트를 배포합니다.

**참조:**

- [비즈니스 규칙 및 비즈니스 규칙 세트 배포 정보](#)
- [비즈니스 규칙 및 비즈니스 규칙 세트를 배포 가능 및 배포 불가능하게 만들기](#)
- [배포 뷰에서 비즈니스 규칙 및 비즈니스 규칙 세트 배포](#)
- [규칙 또는 규칙 세트 디자이너에서 비즈니스 규칙 또는 비즈니스 규칙 세트 배포](#)
- [바로가기 있는 비즈니스 규칙 배포](#)
- [Planning에 표시되는 배포된 비즈니스 규칙 지정](#)

## 비즈니스 규칙 및 비즈니스 규칙 세트 배포 정보

비즈니스 규칙은 Oracle Hyperion Planning, Oracle General Ledger 및 Oracle Essbase에 배포할 수 있습니다. 비즈니스 규칙 세트는 Oracle Hyperion Financial Management, Planning 및 Oracle General Ledger에 배포할 수 있습니다. 하나 이상의 비즈니스 규칙이나 비즈니스 규칙 세트를 배포(부분 배포)하거나 애플리케이션의 모든 비즈니스 규칙 및 규칙 세트를 배포(전체 배포)할 수 있습니다.

 **주:**

여러 레벨에 있는 변수(글로벌, 애플리케이션, 계획 유형 또는 규칙 레벨 중 2개 이상의 레벨에 있는 변수)가 포함된 Planning 비즈니스 규칙에서 최하위 레벨 변수를 삭제하는 경우에는 이 삭제가 Planning에서 이 변수를 사용하는 모든 규칙에 적용되도록 Planning 애플리케이션의 전체 재배포를 수행해야 합니다. 부분 재배포만 수행하면 변수가 삭제되지 않고 Planning에서 계속 사용될 수 있습니다.

Financial Management에는 비즈니스 규칙 세트만 배포할 수 있습니다. 애플리케이션의 각 계산 유형에서 하나의 규칙 세트를 배포할 수 있으며 개별 규칙을 배포할 수는 없습니다.

비즈니스 규칙 세트를 Financial Management에 배포한 후에 데이터 양식 또는 데이터 그리드에서 계산 또는 통합을 실행할 수 있습니다. 비즈니스 규칙 및 비즈니스 규칙 세트를 Planning에 배포한 후에는 데이터 양식에서 실행하거나 실행 메뉴와는 별도로 실행할 수 있습니다. 비즈니스 규칙과 비즈니스 규칙 세트를 Essbase에 배포한 후에 Oracle Essbase Administration Services에서 실행하거나 Oracle Hyperion Calculation Manager에서 직접 실행할 수 있습니다. 비즈니스 규칙 및 비즈니스 규칙 세트를 Oracle General Ledger에 배포한 후에 할당 생성 링크를 사용하여 Oracle General Ledger에서 실행할 수 있습니다.

비즈니스 규칙 및 비즈니스 규칙 세트 실행에 대한 자세한 내용은 다음 리소스를 참조하십시오.

- *Oracle Hyperion Financial Management 사용자 가이드*
- *Oracle Hyperion Planning 사용자 가이드*
- *Oracle Essbase Administration Services 온라인 도움말*
- 이 가이드의 [비즈니스 규칙 실행](#)

## 비즈니스 규칙 및 비즈니스 규칙 세트를 배포 가능 및 배포 불가능하게 만들기

애플리케이션에 비즈니스 규칙 및 규칙 세트의 서브세트를 배포하려면 배포 가능하게 만들어야 합니다. 규칙 및 규칙 세트를 배포 가능하게 만들려면 **배포** 뷰에서 해당 이름 옆의 확인란을 선택합니다.

### 주:

배포하려는 비즈니스 규칙 또는 비즈니스 규칙 세트가 하나인 경우 **배포** 뷰에서 해당 규칙이나 규칙 세트를 배포 가능하게 할 필요가 없습니다. 대신 **시스템** 뷰에서 규칙 또는 규칙 세트를 마우스 오른쪽 버튼으로 누르고 **배포**를 선택하여 이를 배포할 수 있습니다.

비즈니스 규칙 또는 규칙 세트를 배포한 후 애플리케이션에서 해당 비즈니스 규칙 또는 규칙 세트를 제거하려면 **배포** 뷰에서 이름 옆의 확인란을 선택 취소합니다. 그런 후 애플리케이션을 마우스 오른쪽 버튼으로 누르고 **배포**를 선택하여 애플리케이션의 전체 배포를 수행할 수 있습니다.

비즈니스 규칙 및 비즈니스 규칙 세트를 배포 가능하게 만들려면 다음을 수행합니다.

1. **배포** 뷰에서 배포할 규칙 또는 규칙 세트가 포함된 애플리케이션 유형 및 애플리케이션을 확장합니다.
2. **배포 예정**을 확장한 다음 배포할 규칙 및 규칙 세트 옆의 확인란을 선택합니다.  
규칙 또는 규칙 세트를 배포하기 전에 구문 오류가 없는지 검증해야 합니다. 검증 기능을 사용하여 규칙 및 규칙 세트를 수동으로 검증할 수 있습니다. **시스템** 뷰에서 [비즈니스 규칙, 비즈니스 규칙 세트, 공식 및 스크립트 구성요소 검증](#)을 참조하십시오.

## 배포 뷰에서 비즈니스 규칙 및 비즈니스 규칙 세트 배포

배포 뷰에서 비즈니스 규칙 및 비즈니스 규칙 세트를 배포할 수 있습니다. 또한 규칙 디자이너(비즈니스 규칙의 경우) 또는 규칙 세트 디자이너(비즈니스 규칙 세트의 경우)에서 Oracle Hyperion Planning, Oracle Essbase 또는 Oracle General Ledger로 하나의

비즈니스 규칙이나 비즈니스 규칙 세트를 배포할 수도 있습니다. [규칙 또는 규칙 세트 디자이너에서 비즈니스 규칙 또는 비즈니스 규칙 세트 배포](#)를 참조하십시오.

Oracle Hyperion Financial Management에 배포하는 경우 애플리케이션의 각 계산 유형에 대해 하나의 비즈니스 규칙 세트를 배포할 수 있으며, 비즈니스 규칙은 Financial Management에 배포할 수 없습니다.

배포 뷰에서 비즈니스 규칙 및 비즈니스 규칙 세트를 배포하려면 다음을 수행합니다.

1. 시스템 뷰에서 뷰, 배포 뷰 순으로 선택합니다.

 주:

비즈니스 규칙과 비즈니스 규칙 세트를 마우스 오른쪽 버튼으로 누르고 [배포]를 선택하여 시스템 뷰에서 배포할 수도 있습니다.

2. 배포 뷰에서 애플리케이션 유형을 확장합니다.
3. 다음 태스크 중 하나를 수행합니다.
  - **Planning 및 Essbase 사용자:** 애플리케이션 내의 모든 규칙 및 규칙 세트를 배포하려면 배포할 모든 규칙 또는 규칙 세트를 선택하고 애플리케이션을 마우스 오른쪽 버튼으로 누른 다음 [배포]를 선택합니다.
  - **Planning 및 Essbase 사용자:** 비즈니스 규칙 및 비즈니스 규칙 세트의 서브세트를 배포하려면(부분 배포) 애플리케이션 및 배포 예정 노드를 확장한 후 다음 단계를 수행합니다.
    - a. 배포할 규칙 세트를 선택하지 않은 경우 선택합니다.
    - b. 배포할 규칙이 들어 있는 계획 유형을 확장합니다.
    - c. 배포할 규칙을 선택하지 않은 경우 선택합니다.
    - d. 마우스 오른쪽 버튼을 누르고 배포를 선택합니다.

 팁:

여러 규칙 또는 규칙 세트를 배포하려면 Ctrl + 누르기 및 Shift + 누르기를 사용하여 선택하고 마우스 오른쪽 버튼을 누른 후 [배포]를 선택합니다.

 주:

다른 애플리케이션에 대한 바로가기를 포함하는 비즈니스 규칙을 배포할 경우 바로가기가 있는 각 애플리케이션으로 비즈니스 규칙이 배포됩니다.

- **Financial Management 사용자만 해당:** 비즈니스 규칙 세트를 배포하려면 다음을 수행합니다.
  - a. 애플리케이션, 배포 예정 노드, 배포할 규칙 세트가 들어 있는 계산 유형 순으로 확장합니다.
  - b. 배포할 규칙 세트를 선택하지 않은 경우 선택합니다. 계산 유형당 하나의 규칙 세트를 배포할 수 있습니다.

 주:

배포에 대해 Generic\_Ruleset를 포함해야 합니다.

- c. 애플리케이션을 마우스 오른쪽 버튼으로 누르고 **배포**를 선택합니다. 배포 시 .xml 형식의 객체가 .rle 형식으로 변환된 후 Financial Management 애플리케이션으로 로드됩니다.

 팁:

여러 규칙 세트를 배포하려면 Ctrl + 누르기 및 Shift + 누르기를 사용하여 선택하고 마우스 오른쪽 버튼을 누른 후 [배포]를 선택합니다.

배포에 성공하면 "배포에 성공했습니다." 메시지가 표시됩니다.

## 규칙 또는 규칙 세트 디자이너에서 비즈니스 규칙 또는 비즈니스 규칙 세트 배포

비즈니스 규칙 또는 규칙 세트를 디자인한 후 규칙 디자이너 또는 규칙 세트 디자이너에서 직접 검증하고 배포할 수 있습니다.

규칙 디자이너 또는 규칙 세트 디자이너에서 비즈니스 규칙 또는 비즈니스 규칙 세트를 배포하려면 다음을 수행합니다.

1. 다음 태스크 중 하나를 수행합니다.
  - 비즈니스 규칙을 배포하려면 해당 규칙이 포함된 계획 유형 또는 데이터베이스, **규칙** 순으로 확장합니다.
  - 비즈니스 규칙 세트를 배포하려면 **규칙 세트**를 확장합니다.
2. 배포하려는 규칙 또는 규칙 세트를 마우스 오른쪽 버튼으로 누르고 **열기**를 선택합니다.
3. **규칙 디자이너** 또는 **규칙 세트 디자이너**에서 **작업**, **배포** 순으로 선택합니다.

배포에 성공하면 "배포에 성공했습니다." 메시지가 표시됩니다.

## 바로가기 있는 비즈니스 규칙 배포

바로가기 있는 비즈니스 규칙을 애플리케이션에 배포하는 경우 바로가기를 생성한 각 애플리케이션에 규칙 사본이 배포됩니다.

바로가기 있는 비즈니스 규칙을 배포하려면 다음을 수행합니다.

1. **시스템 뷰**에서 **뷰**, **배포 뷰** 순으로 선택합니다.
2. 애플리케이션 유형, 애플리케이션, **배포 예정** 노드, 계획 유형 또는 데이터베이스 순으로 확장합니다.
3. 배포하려는 규칙을 마우스 오른쪽 버튼으로 누르고 **모두 배포**를 선택합니다.

## Planning에 표시되는 배포된 비즈니스 규칙 지정

Oracle Hyperion Calculation Manager에 배포된 비즈니스 규칙은 Oracle Hyperion Planning의 비즈니스 규칙 페이지에서 보고 실행할 수 있습니다.

Planning에 표시하려는 비즈니스 규칙을 지정할 수 있습니다. 이 작업을 하려면 다음을 수행합니다.

1. 배포 뷰에서 Planning에 표시하지 않으려는 규칙을 선택 취소합니다.
2. 애플리케이션 노드를 마우스 오른쪽 버튼으로 누르고 배포를 선택합니다.

이전에 Planning에 표시되었지만 Calculation Manager의 배포 뷰에서 선택 취소된 규칙은 애플리케이션이 배포되고 나면 더 이상 Planning에 표시되지 않습니다.

# 10

## 비즈니스 규칙 실행

### 참조:

- [비즈니스 규칙 실행 정보](#)  
Oracle Hyperion Calculation Manager의 시스템 뷰 또는 규칙 디자이너에서 Oracle Hyperion Planning 비즈니스 규칙을 실행할 수 있습니다.
- [시스템 뷰 및 규칙 디자이너에서 Essbase Business Rules 실행](#)
- [명령행 실행 프로그램으로 Essbase Business Rules 실행](#)
- [규칙 디자이너에서 Planning 비즈니스 규칙 실행 및 로그 보기](#)  
Oracle Hyperion Planning 비즈니스 규칙을 실행하고 생성된 로그를 볼 수 있습니다.

## 비즈니스 규칙 실행 정보

Oracle Hyperion Calculation Manager의 시스템 뷰 또는 규칙 디자이너에서 Oracle Hyperion Planning 비즈니스 규칙을 실행할 수 있습니다.

Planning 비즈니스 규칙을 Planning에 배포하고 Planning에서 실행할 수도 있습니다. Planning에서 Planning 비즈니스 규칙을 실행하는 방법에 대한 자세한 내용은 *Oracle Hyperion Planning 사용자 가이드*를 참조하십시오.

### 주:

- Calculation Manager에서 규칙을 검증, 디버깅, 배포, 분석 또는 실행할 때 런타임 프롬프트 변수의 값을 입력하거나 편집할 수 있습니다. 비즈니스 규칙 세트를 검증하거나 배포할 때 런타임 프롬프트 변수를 입력하거나 편집할 수도 있습니다. 런타임 프롬프트에 멤버 한도가 있는 경우 검증 시 적합한 멤버 이름만 확인합니다 (멤버가 한도 내에 있는지 여부는 검증하지 않음). 숫자 및 정수 유형 변수는 런타임 프롬프트 한도가 검증됩니다.
- Calculation Manager에서 규칙을 실행할 때 적합한 조합 및 보안은 적용되지 않습니다.
- 멤버 승인 보안을 적용하도록 런타임 프롬프트를 디자인하는 방법에 대한 자세한 내용은 *Oracle Hyperion Planning 관리자 가이드*에서 [런타임 프롬프트 및 승인 보안 정보](#)를 참조하십시오.

## 시스템 뷰 및 규칙 디자이너에서 Essbase Business Rules 실행

시스템 뷰 또는 규칙 디자이너에서 Oracle Essbase 비즈니스 규칙을 실행할 수 있습니다. Essbase 비즈니스 규칙을 실행하려면 다음을 수행합니다.



1. 시스템 뷰에서 다음 태스크 중 하나를 수행합니다.
  - 실행할 비즈니스 규칙을 마우스 오른쪽 버튼으로 누르고 **실행**을 선택합니다.
  - 실행할 규칙을 두 번 누릅니다. 규칙 디자이너에 규칙이 열리면 **작업**, **실행** 순으로 선택합니다.
2. 런타임 프롬프트 입력에 대한 자세한 내용은 [런타임 프롬프트 변수 입력](#)을 참조하십시오.

## 명령행 실행 프로그램으로 Essbase Business Rules 실행

명령행 실행 프로그램 유틸리티는 %EPM\_ORACLE\_HOME%\common\calcmgr\11.1.2.0\lib의 calcmgrCmdLine.jar 파일에 있습니다. 유틸리티를 실행하는 컴퓨터에 Java Runtime 1.6+ 이진 파일을 설치해야 합니다. 유틸리티를 실행하려면 유틸리티를 실행할 컴퓨터에 jar 파일을 복사하고 다음 명령을 사용합니다.

```
java -jar calcmgrCmdLine.jar <Calc Server URL> <USER> <PASSWORD>
<APPLICATION NAME> <DATABASE NAME> <RULE NAME> [, <VARIABLE
NAME>=<VARIABLE VALUE>]
```

- 명령행 실행 프로그램은 한 번에 하나의 규칙만 실행할 수 있습니다.
- 규칙에서 참조되지 않는 변수 이름이 전달되면 이 이름은 무시되며, 이에 관한 메시지는 생성되지 않습니다.
- 규칙에서 변수를 참조하지만 인수로 전달되지 않는 경우, 생성된 규칙 스크립트에서 검증 값 또는 변수의 기본 값(지정된 경우)을 사용합니다.
- 실행에 성공하면 오류 레벨 0이 반환됩니다. 실행에 실패하면 오류 레벨 1이 반환됩니다.

## 명령행 실행 프로그램 예제 및 인수

다음은 명령행 실행 프로그램의 예입니다.

```
java -jar calcmgrCmdLine.jar http://myBox:19000 admin password Sample
Basic myRule var1=\"Jan\" var2=1000 var3=\"Budget->\"001\"->FY10"
```

- <Calc Server URL> - 프로토콜(예: http 또는 https)과 호스트 이름 및 포트
- <USER> <PWD> - 사용자 이름 및 비밀번호
- <APPLICATION NAME> - 규칙이 생성되는 Oracle Essbase 애플리케이션
- <DATABASE NAME> - Essbase 규칙이 생성되는 데이터베이스(애플리케이션 내)
- <RULE NAME> - Business Rule의 이름
- <VARIABLE NAME> - Business Rule에서 대체될 런타임 프롬프트 변수 이름
- <VARIABLE VALUE> - 런타임 프롬프트 변수의 값

**암호화되지 않은 모드 예**

```
java -jar calcmgrCmdLine.jar <CalcServer URL> <USER NAME> <PWD> <Application Name> <Database Name> <Rule Name> <variable name=value>,<variable name=value>
```

```
java -jar calcmgrCmdLine.jar http://myBox:19000 admin password Sample Basic myRule var1="Jan" var2=1000
```

**암호화된 모드 예**

암호화된 비밀번호를 인수로 명령행 실행 프로그램에 전달해야 하는 경우, 먼저 비밀번호의 암호화된 토큰을 생성합니다. 다음과 같이 기본 키를 사용하여 비밀번호를 암호화할 수 있습니다.

```
java -jar calcmgrCmdLine.jar -encrypt password
```

**기본 키를 사용하여 해독하는 암호화된 모드 예**

기본 키를 사용하여 비밀번호를 해독하려는 경우 다음과 같이 해독합니다.

```
java -jar calcmgrCmdLine.jar -decrypt http://myBox:19000 admin g//F19AB2qrFN9R5m141CPgcLYmjmLyiIxx2CXz7hxbheZe6HDUvjYkQuW214Oer Sample Basic myRule var1="Jan" var2=1000
```

**제공된 키를 사용하여 암호화하는 암호화된 모드 예**

제공된 키를 사용하여 비밀번호를 암호화하려는 경우 다음과 같이 암호화합니다.

```
java -jar calcmgrCmdLine.jar -encrypt -key AAAAAAAAAAAAAAAAAAAAAAAA password
```

**제공된 키를 사용하여 해독하는 암호화된 모드 예**

```
java -jar calcmgrCmdLine.jar -key AAAAAAAAAAAAAAAAAAAAAAAA http://myBox:19000 admin g//F19AB2qrFN9R5m141CPgcLYmjmLyiIxx2CXz7hxbheZe6HDUvjYkQuW214Oer Sample Basic myRule var1="Jan" var2=1000
```

## Windows에서 Bat 파일을 사용하여 명령행 실행 프로그램 실행

다음은 Windows에서 bat 파일을 사용하여 명령행 실행 프로그램을 실행하는 예입니다.

```
@echo off
java -jar calcmgrCmdLine.jar http://myBox:19000 admin password Sample Basic myRule var1=\"Jan\" var2=1000 var3=\"Budget->\"001\"->FY10"
IF %ERRORLEVEL%==0 goto success
echo there is error
goto end
:success
echo success
:end
```

# 규칙 디자이너에서 Planning 비즈니스 규칙 실행 및 로그 보기

Oracle Hyperion Planning 비즈니스 규칙을 실행하고 생성된 로그를 볼 수 있습니다.

비즈니스 규칙이 규칙 디자이너에서 보거나 편집하기 위해 열려 있는 경우 규칙을 실행하고 규칙 디자이너의 로그 메시지 탭에서 생성된 로그를 볼 수 있습니다. 로그를 심표로 구분된 값(.csv) 파일로 익스포트할 수 있습니다.

규칙 디자이너에서 비즈니스 규칙을 실행하고 로그 메시지를 보려면 다음을 수행합니다.

1. 시스템 뷰에서 규칙을 두 번 누릅니다.

2. 규칙 디자이너에서 **작업, 실행** 순으로 선택합니다.

규칙이 실행되면 오류 여부와 관계 없이 규칙이 실행되었음을 나타내는 확인 메시지가 표시됩니다. **확인**을 눌러 확인 메시지를 닫습니다.

3. **로그 메시지** 탭을 선택합니다.

로그 메시지에 다음 정보가 포함되어 있습니다.

- **메시지 번호** - 로그 파일에 표시된 메시지 ID
- **메시지 레벨** - 메시지 심각도/레벨
- **메시지 텍스트** - 메시지의 전체 텍스트
- **메시지 타임스탬프** - 메시지가 생성된 시간을 나타내는 타임스탬프
- **전달 번호** - 현재 전달의 번호. 최고/최종 숫자가 규칙의 전달 수입니다.
- **전달 시간** - 실행 시간(초). 현재 전달에 소요된 시간입니다.
- **누적 시간** - 총 실행 시간(초). 규칙 실행이 시작된 후 경과된 총 시간입니다.

**참고:** 데이터베이스에서는 많은 로그 메시지가 동시에 생성되므로 전달 시간(초)만 각 전달의 시간을 해당 전달에 소요된 시간으로 표시하며 누적 시간(초)는 규칙 시작부터 소요된 시간을 표시합니다. 다른 모든 행은 공백으로 표시합니다.

- **전달 %** - 총 시간 중 해당 규칙의 해당 전달이 차지하는 퍼센트
- **누적 %** - 총 시간 중 해당 규칙의 모든 전달이 차지하는 퍼센트. 모든 전달이 완료되면 누적 퍼센트가 100%가 되어야 합니다.

4. **선택 사항**입니다. 표시된 로그 메시지를 필터링합니다.

다음 기준에 따라 필터링할 수 있습니다.

- **메시지 번호** - 메시지 번호 열 위의 드롭다운에서 값을 선택합니다.
- **메시지 레벨** - 메시지 레벨 열 위의 드롭다운에서 값을 선택합니다.
- **메시지 텍스트** - 메시지 텍스트 열 위의 텍스트 상자에 텍스트를 입력합니다.
- **전달만** - 전달 번호 및 각 전달에 소요된 시간만 보려면 **전달 #** 열 위의 드롭다운에서 **전달만**을 선택합니다. 완료 로그 정보 표시로 돌아가려면 **전달 #** 열 위의 공백 옵션을 선택합니다.

 주:

비즈니스 규칙을 Planning에 배포한 후 Planning에서 실행할 수도 있습니다. 자세한 내용은 *Oracle Hyperion Planning 사용자 가이드*를 참조하십시오.

 주:

로그 메시지를 심표로 구분된 값(.csv) 파일로 내보낼 수 있습니다. [로그 메시지를 파일로 익스포트](#)를 참조하십시오.

# 비즈니스 규칙, 비즈니스 규칙 세트, 템플릿, 공식 및 스크립트 구성요소 익스포트 및 임포트

## 참조:

- [익스포트 및 임포트 정보](#)  
애플리케이션에서 객체를 익스포트 및 임포트합니다.
- [비즈니스 규칙, 비즈니스 규칙 세트, 템플릿, 공식 및 스크립트 구성요소 익스포트](#)  
애플리케이션, 단일 객체 또는 여러 객체를 내보내면 다른 Oracle Hyperion Calculation Manager 애플리케이션으로 가져올 수 있는 xml 파일로 내보내집니다.
- [애플리케이션 익스포트](#)  
애플리케이션을 내보내면 애플리케이션 콘텐츠가 xml 파일에 저장됩니다.
- [로그 메시지를 파일로 익스포트](#)  
Oracle Hyperion Calculation Manager에서 Oracle Hyperion Planning 비즈니스 규칙을 실행하는 경우 규칙 디자이너의 로그 메시지 탭에 로그 메시지가 생성되고 표시됩니다.
- [규칙, 규칙 세트, 템플릿, 공식 및 스크립트 임포트](#)  
Oracle Hyperion Calculation Manager를 사용하는 애플리케이션으로 규칙, 규칙 세트, 템플릿, 공식 및 스크립트를 임포트합니다.
- [Essbase 계산 스크립트를 그래픽 비즈니스 규칙으로 임포트](#)

## 익스포트 및 임포트 정보

애플리케이션에서 객체를 익스포트 및 임포트합니다.

Oracle Hyperion Financial Management, Oracle Hyperion Planning, Oracle Essbase 또는 Oracle General Ledger 애플리케이션에 있는 모든 객체를 익스포트할 수 있습니다. 또한 애플리케이션 내에 있는 개별 비즈니스 규칙, 비즈니스 규칙 세트, 템플릿 및 공식/스크립트 구성요소를 익스포트할 수 있습니다.

또한 Oracle Hyperion Business Rules에서 비즈니스 규칙, 시퀀스, 매크로 및 변수를 내보내고 Oracle Hyperion Calculation Manager에 가져올 수 있습니다. Calculation Manager에서 시퀀스는 비즈니스 규칙 세트로 변환되고, 매크로는 템플릿으로 변환됩니다.

애플리케이션 및 객체를 내보낸 후 다른 Financial Management, Planning, Essbase 또는 Oracle General Ledger 애플리케이션으로 가져올 수 있습니다. 예를 들어 업무 컴퓨터의 애플리케이션에서 비즈니스 규칙 및 비즈니스 규칙 세트를 내보내고 테스트 컴퓨터의 다른 애플리케이션에 이를 가져올 수 있습니다.

Essbase 계산 스크립트를 그래픽 비즈니스 규칙으로 Calculation Manager에 가져올 수도 있습니다.

# 비즈니스 규칙, 비즈니스 규칙 세트, 템플릿, 공식 및 스크립트 구성요소 익스포트

애플리케이션, 단일 객체 또는 여러 객체를 내보내면 다른 Oracle Hyperion Calculation Manager 애플리케이션으로 가져올 수 있는 xml 파일로 내보내집니다.

## 주:

시스템 뷰, 사용자정의 뷰 및 배포 뷰에서 객체를 내보낼 수 있습니다. 단일 객체 또는 여러 객체를 내보낼 수 있습니다.

객체를 내보내려면 다음을 수행합니다.

1. 다음 태스크 중 하나를 수행합니다.
  - **Oracle Hyperion Financial Management 사용자만 해당:** Financial Management 규칙 세트를 익스포트하려면 계산 유형과 익스포트할 규칙 세트가 들어 있는 **규칙 세트** 노드를 확장합니다.
  - 규칙 세트를 내보내려면 **규칙 세트**를 확장합니다.  
Oracle Hyperion Planning, Oracle Essbase 및 Oracle General Ledger 애플리케이션만 해당
  - 규칙, 공식, 스크립트 또는 템플릿을 익스포트하려면 계산 유형, 계획 유형 또는 데이터베이스를 확장한 다음 **규칙, 공식, 스크립트 또는 템플릿**을 확장합니다.
2. 다음 태스크 중 하나를 수행합니다.
  - 하나의 객체만 내보내려면 해당 객체를 마우스 오른쪽 버튼으로 누른 후 **익스포트**를 선택합니다.
  - 여러 객체를 내보내려면 내보낼 객체를 선택한 후 마우스 오른쪽 버튼으로 누르고 **익스포트**를 선택합니다. 다양한 계산 유형, 계획 유형 또는 데이터베이스, 다양한 객체 유형(예: 비즈니스 규칙 및 공식), 애플리케이션 유형 내의 다양한 애플리케이션에서 연속하거나 연속하지 않는 객체를 선택하려면 **Shift + 누르기 및 Ctrl + 누르기**를 사용합니다.  
[익스포트]를 선택하면 생성된 .xml 파일을 열거나 저장할지 묻는 메시지가 표시됩니다.
3. **파일 다운로드**에서 다음 태스크 중 하나를 수행합니다.
  - 생성된 .xml 파일의 내용을 보려면 **열기**를 선택합니다.
  - 생성된 .xml 파일을 먼저 보지 않고 저장하려면 **저장**을 선택하고 파일 이름을 입력하거나 기본 이름을 그대로 사용하고 **저장**을 다시 누릅니다.

## 애플리케이션 익스포트

애플리케이션을 내보내면 애플리케이션 콘텐츠가 xml 파일에 저장됩니다.

애플리케이션을 내보내려면 다음을 수행합니다.

1. 애플리케이션을 마우스 오른쪽 버튼으로 누르고 **익스포트**를 선택합니다.

2. **파일 다운로드**에서 다음 태스크 중 하나를 수행합니다.
  - 생성된 xml 파일의 콘텐츠를 보려면 **열기**를 선택합니다.
  - 해당 콘텐츠를 먼저 보지 않고 생성된 xml 파일을 저장하려면 **저장**을 선택하고 파일 이름을 입력하거나 기본 이름을 그대로 사용하고 **저장**을 다시 누릅니다.

## 로그 메시지를 파일로 익스포트

Oracle Hyperion Calculation Manager에서 Oracle Hyperion Planning 비즈니스 규칙을 실행하는 경우 규칙 디자이너의 로그 메시지 탭에 로그 메시지가 생성되고 표시됩니다.

이러한 로그 메시지를 심표로 구분된 값(.csv) 파일로 내보낼 수 있습니다. [규칙 디자이너에서 Planning 비즈니스 규칙 실행 및 로그 보기](#)를 참조하십시오.

Planning 비즈니스 규칙을 Calculation Manager에서 실행하여 생성된 로그 메시지를 내보내려면 다음을 수행합니다.

1. **시스템 뷰**에서 실행할 규칙을 두 번 누릅니다.
2. 규칙 디자이너에서 규칙이 열리면 **작업, 실행** 순으로 선택합니다.  
규칙이 실행되면 오류 여부와 관계 없이 규칙이 실행되었음을 나타내는 확인 메시지가 표시됩니다.
3. **확인**을 눌러 확인 메시지를 종료합니다.  
로그 메시지는 로그 메시지 탭에 표시됩니다.
4. 규칙이 실행되는 동안 생성된 로그 메시지를 내보내려면 **작업, 익스포트** 순으로 선택합니다.  
익스포트 후 테이블의 로그 메시지가 모두 포함된 RuleLogMessages.csv 파일을 다운로드할 수 있습니다. 파일을 저장하고 심표가 구분자로 포함되는 Microsoft Excel을 사용하여 여십시오.

## 규칙, 규칙 세트, 템플릿, 공식 및 스크립트 임포트

Oracle Hyperion Calculation Manager를 사용하는 애플리케이션으로 규칙, 규칙 세트, 템플릿, 공식 및 스크립트를 임포트합니다.

가져온 후 임포트 결과를 로컬 파일에 저장할 수 있습니다.


객체를 가져오려면 파일 유형이 다음 중 하나여야 합니다.

- **모든 사용자:** .xml, 객체가 포함된 xml 형식 파일
- **모든 사용자:** .zip, zip 파일에는 xml 파일만 있을 수 있음
- **Oracle Hyperion Financial Management 사용자만 해당:** .rle, 객체가 들어 있는 Visual Basic 형식 파일
- **Financial Management 사용자만 해당:** .cmo, 객체가 들어 있는 ASCII 텍스트 형식 파일
- **Oracle Hyperion Planning, Oracle Essbase 및 Oracle General Ledger 사용자만 해당:** .csc, 객체가 들어 있는 Essbase 계산 스크립트 형식 파일

### 주:

한 제품 유형에서 다른 제품 유형으로 임포트할 수는 없습니다.

객체를 가져오려면 다음을 수행합니다.

1. 시스템 뷰에서 작업, 임포트 순으로 선택하거나  를 누릅니다.
2. 임포트 대화상자의 파일 임포트 세부정보에서 찾아보기를 누른 다음 가져올 파일을 선택합니다.
3. 위치 세부정보에서 애플리케이션 유형, 애플리케이션 및 큐브를 입력합니다.
  - .csc 파일을 가져오려면 위치 세부정보를 입력해야 합니다.
  - 파일이 .xml 파일인 경우 위치 세부정보가 임포트 파일에 이미 있으면 위치 세부정보를 입력할 필요가 없습니다.
  - 파일이 .rle 파일이며 위치 세부정보를 입력한 경우, 키워드 'app\_name=<애플리케이션 이름>'이 .rle 파일의 첫번째 라인에 있어야 합니다. 예를 들면 'app\_name=Statutory'입니다.
  - .cmo 파일을 가져오려면 애플리케이션 정보가 키워드에 포함됩니다.

위치 세부정보에 입력된 정보가 임포트 파일에 지정된 위치를 재정의합니다. 임포트 파일에 위치 정보가 지정되지 않은 경우 위치 세부정보에 정보를 입력해야 하며 정보를 입력하지 않으면 임포트에 실패합니다.
4. 임포트 옵션에서 다음 옵션 중 하나를 선택합니다.
  - 기존 객체 덮어쓰기 - 임포트된 객체가 애플리케이션 및 계산, 큐브 또는 데이터베이스의 객체를 바꿉니다.
  - 기존 객체 건너뛰기 - 임포트된 객체가 아직 없는 경우 애플리케이션 및 계산, 큐브 또는 데이터베이스의 객체에 임포트된 객체를 추가합니다. 그렇지 않으면 객체가 임포트되지 않으며 결과의 상태가 "건너뛸"으로 표시됩니다.
  - 중복 오류 - 임포트된 객체가 애플리케이션 및 계산, 큐브 또는 데이터베이스에 이미 있는 객체와 중복될 경우 중복되는 객체의 이름이 로그 파일에 기록되며 객체가 임포트되지 않고 임포트 프로세스가 중지됩니다.
5. **Financial Management 사용자만 해당:** .rle 파일을 임포트하는 경우 다음 변환 옵션을 사용할 수 있습니다.
  - 스크립트 구성요소 - .rle 파일을 Calculation Manager의 스크립트 구성요소로 변환합니다.
  - 그래픽 구성요소 - .rle 파일을 Calculation Manager의 그래픽 구성요소로 변환합니다.

그래픽 구성요소를 선택하는 경우 다음을 수행할 수 있습니다.

  - 플로우 차트 공식 구성요소의 일부로 조건을 포함하는 구성요소 레벨 조건부 블록 사용을 선택합니다. 기본적으로 조건은 공식 구성요소의 일부가 아니라 별도의 조건 구성요소로 순서도에 포함됩니다.
  - 스크립트의 설명에 따라 명령문을 구성요소 객체로 구분하는 방법을 지정하는 옵션을 선택합니다.
    - \* 별도의 구성요소 - 스크립트에 있는 각 설명 라인에 대해 별도의 구성요소가 생성됩니다.
    - \* 복수 블록 구성요소 - 스크립트에 있는 각 설명 라인에 대해 블록 하나가 포함된 단일 구성요소가 생성됩니다.
    - \* 단일 블록 구성요소 - 스크립트에 있는 모든 설명 라인에 대해 블록 하나가 포함된 단일 구성요소가 생성됩니다. 설명 라인에 대한 별도의 블록이 없습니다.



6. 임포트를 누릅니다.
7. **선택 사항:** 임포트 결과를 로컬 파일에 저장하려면 **다른 이름으로 저장**을 누릅니다.

## Essbase 계산 스크립트를 그래픽 비즈니스 규칙으로 임포트

Oracle Essbase 계산 스크립트를 그래픽 비즈니스 규칙으로 Oracle Hyperion Calculation Manager에 임포트할 수 있습니다.

SET RUNTIMESUBVARS 계산 명령이 포함된 Essbase 계산 스크립트를 임포트할 때 모든 Essbase 런타임 대체 변수(스크립트에서 앞에 앰퍼샌드가 옴)가  $\{ \}$  표기가 있는 Calculation Manager 런타임 프롬프트 변수로 변환됩니다. 기본적으로 문자열 유형의 변수는 Essbase 계산 스크립트에 있는 RUNTIMESUBVARS 명령의 변수 값을 사용하여 Calculation Manager에 생성됩니다. Essbase에서 이러한 계산 스크립트를 생성하는 방법에 대한 자세한 내용은 *Oracle Essbase Database Administrator's Guide*를 참조하십시오.

계산 관리자에서 변수 값은 규칙 디자이너에서 Business Rule을 열 때 변수 탭의 값 열에 표시됩니다. (변수 디자이너를 열고 기본 값 열을 확인하여 변수의 기본값을 확인할 수 있습니다.) 그러나 계산 스크립트 변수에 값 또는 기본값이 없으면 계산 관리자에 값이 표시되지 않습니다.

계산 스크립트의 <RTSV\_HINT>와 </RTSV\_HINT> 태그 사이에 텍스트가 있으면, 이 텍스트는 변수의 런타임 프롬프트 텍스트로 사용됩니다. 이러한 태그 사이에 텍스트가 없으면 프롬프트 텍스트가 변수의 이름입니다.

Essbase 계산 스크립트를 그래픽 비즈니스 규칙으로 가져오려면 다음을 수행합니다.

1. **시스템** 뷰에서 데이터베이스를 마우스 오른쪽 버튼으로 누르고 **Essbase 계산 스크립트 임포트**를 선택합니다.
  - 이 메뉴 옵션은 가져올 Essbase 계산 스크립트가 있는 경우에만 사용할 수 있습니다.
2. 각 계산 스크립트에 대해 Calculation Manager에 이미 있는지 여부와 가져올 것인지 여부를 선택합니다. 모든 계산 스크립트를 가져오려면 **모두 선택**을 누릅니다.
3. 다음 옵션 중 하나를 선택합니다.
  - 기존 객체를 같은 이름의 새 객체로 바꾸려면 **기존 객체 재정의**를 선택합니다.
  - 같은 이름의 새 객체가 있을 때 기존 객체를 유지하려면 **기존 객체 건너뛰기**를 선택합니다.
4. **임포트를 누릅니다.**

### 주:

가져온 계산 스크립트를 스크립트 구성요소로 가져온 경우 구문분석 오류가 있을 수 있습니다. 규칙을 검증하고, 구문분석 오류가 있을 경우 다음 중 하나를 수행하여 수정합니다. 1) Calculation Manager에서 구문분석 오류를 수정하고 규칙을 검증합니다. 스크립트 구성요소를 마우스 오른쪽 버튼으로 누르고 [그래픽으로 변환]을 선택한 다음 규칙을 저장합니다. 그래픽으로의 변환을 실행취소하려면 [편집], [변환 실행취소] 순으로 선택합니다. 2) 원래 계산 스크립트를 열어서 구문분석 오류를 수정하고 계산 스크립트를 저장한 다음 Calculation Manager로 다시 계산 스크립트를 임포트합니다. [이미 있음] 아래에서 계산 스크립트에 확인 표시가 있습니다. [임포트] 확인란과 [기존 객체 덮어쓰기]를 선택합니다.

계산 스크립트를 가져온 후 가져온 계산 스크립트가 스크립트 구성요소에 있고 그래픽 형식으로 보려는 경우 규칙 순서도에서 스크립트 구성요소를 마우스 오른쪽 버튼으로 누르고 [그래픽으로 변환]을 선택합니다.

임포트 결과를 로컬 파일에 저장할 수도 있습니다.

# 12

## Planning 애플리케이션에 대한 Essbase 서버, 애플리케이션 및 데이터베이스 관리

### 참조:

- [데이터베이스 등록정보 작업](#)  
일반, 차원, 통계, 트랜잭션 및 수정 등록정보를 보고 편집합니다.
- [MaxL 스크립트 작성 및 실행](#)
- [데이터베이스 객체에서 잠금 제거](#)  
보안 권한에 따라 객체를 보고 잠금 해제할 수 있습니다.
- [애플리케이션 시작 및 중지](#)  
읽기 이상의 권한이 있는 애플리케이션을 시작할 수 있습니다. Oracle Essbase는 새로 시작된 애플리케이션을 Essbase Server의 메모리에 로드합니다.
- [데이터베이스 시작 및 중지](#)  
데이터베이스를 시작하면 Oracle Essbase는 데이터베이스를 Essbase Server의 메모리에 로드합니다.
- [애플리케이션 로그 파일 삭제](#)
- [데이터베이스 재구조화](#)  
밀집 차원에 멤버를 추가하는 등 데이터베이스를 재구조화하는 경우 데이터 블록을 다시 계산해야 할 수도 있습니다.
- [아웃라인 확인](#)  
Oracle Essbase 아웃라인에 오류가 있는지 확인할 수 있습니다.
- [데이터베이스 설정 최적화에 유용한 시뮬레이션 생성](#)
- [데이터베이스에서 데이터 지우기](#)  
집계 저장영역 애플리케이션에서 데이터를 지우고, 블록 저장영역 애플리케이션에서 데이터 블록을 지웁니다.
- [블록 저장영역 애플리케이션에 대한 위치 별칭 작업](#)  
위치 별칭은 데이터 소스에 대한 기술자입니다. 위치 별칭을 표시하고 익스포트할 수 있습니다.
- [집계 저장영역 데이터베이스에서 쿼리 추적 사용](#)  
쿼리 데이터를 사용하여 데이터베이스에 대해 구체화할 가장 적절한 집계 뷰 세트를 선택합니다.
- [집계 저장영역 데이터베이스 아웃라인 압축](#)  
파일을 압축하여 삭제된 멤버의 레코드를 제거하고 아웃라인 파일 크기를 줄입니다.
- [레벨 0 데이터 임포트 및 익스포트](#)  
ASO 및 BSO 큐브에서 레벨 0 데이터를 임포트 및 익스포트합니다.
- [증분 데이터 슬라이스 병합](#)  
모든 증분 데이터 슬라이스를 주 데이터베이스 슬라이스로 병합하거나, 주 데이터베이스 슬라이스가 변경되지 않는 동안 모든 증분 데이터 슬라이스를 단일 데이터 슬라이스로 병합할 수 있습니다.

- **데이터 집계**  
데이터가 포함되고 계산 권한을 부여받은 집계 저장영역 데이터베이스에 대한 집계를 계산합니다.
- **집계 프로세스 실행**  
집계 프로세스를 실행하면 검색 성능이 향상됩니다.
- **요청 관리**  
세션 창의 정보를 사용하여 활성 요청을 관리합니다.
- **Planning 드릴스루 정의 추가**  
Oracle Hyperion Calculation Manager에서는 Oracle Hyperion Planning 계획 유형에 대한 셀 드릴스루 정의를 나열하고, 추가하고, 편집하고, 삭제할 수 있습니다.

## 데이터베이스 등록정보 작업

일반, 차원, 통계, 트랜잭션 및 수정 등록정보를 보고 편집합니다.



### 참조:

- 데이터베이스 등록정보 보기 및 편집
- 일반 데이터베이스 등록정보
- 차원 등록정보
- 통계 등록정보
- 트랜잭션 등록정보
- 수정 등록정보
- 통화 등록정보
- 캐시 등록정보

## 데이터베이스 등록정보 보기 및 편집

Oracle Hyperion Planning 블록 저장영역 및 집계 저장영역 애플리케이션에 대한 데이터베이스 등록정보를 보고 편집할 수 있습니다.

데이터베이스 등록정보를 보거나 편집하려면 다음을 수행합니다.

1. 시스템 뷰에서  를 누릅니다.
2. 엔터프라이즈 뷰에서 Planning 애플리케이션과 블록 저장영역 또는 집계 저장영역 애플리케이션을 확장하고 데이터베이스를 선택합니다.
3. 데이터베이스 등록정보 탭에서 정보를 보거나 편집하고  를 누릅니다.

### 주:

데이터베이스 등록정보를 변경하면 애플리케이션을 중지했다가 다시 시작해야 합니다. [애플리케이션 시작 및 중지](#)를 참조하십시오.

## 일반 데이터베이스 등록정보

데이터베이스에 대한 일반 정보로, 다음 영역의 등록정보가 포함되어 있습니다.

- **일반** - 데이터베이스에 대한 설명을 입력하고 데이터베이스 유형, 데이터베이스 상태 (로드되었는지 여부), 데이터베이스에 대한 최소 액세스 레벨을 확인합니다.
- **계산 - 블록 저장영역 애플리케이션에만 해당**
  - **누락 값 집계** - 데이터베이스 계산 중 누락 값을 집계합니다.  
 기본적으로 Oracle Essbase는 전체 데이터베이스 계산 중에 누락된(#Missing) 값을 집계하지 않습니다. 1차 하위 구성요소 레벨에서 데이터가 로드되지 않은 경우 누락된 값을 집계하면 계산 성능이 향상될 수 있습니다. 데이터베이스 관리자 권한이 있는 데이터베이스의 경우 누락된 값을 집계할지 여부를 선택할 수 있습니다.  
  
 상위 레벨에서 데이터를 로드하지 않는 경우 누락 값을 집계하면 계산 성능이 향상될 수 있습니다. 누락 값을 집계하고 상위 레벨에서 데이터를 로드하면 결과가 #Missing 값이어도 상위 레벨 값이 데이터베이스 통합 결과로 바뀝니다.
  - **방정식에 대한 블록 생성** - 특정 멤버 조합에 대한 데이터 블록을 생성합니다.  
 방정식에 대한 블록을 생성하면 데이터 블록이 없는 멤버 조합에 비상수 값을 지정할 때 Essbase에서 데이터 블록을 생성합니다. 방정식에 대한 블록을 생성하면 매우 큰 데이터베이스가 생성될 수 있습니다.  
  
 회소 차원의 멤버에 상수를 지정하면 Essbase에서 데이터 블록을 생성합니다. 따라서 회소 멤버에 상수를 지정하는 경우(예: "West = 5") [방정식에 대한 블록 생성]을 선택하지 마십시오.  
  
 그러나 상수가 아닌 다른 값을 회소 멤버에 지정하는 경우 블록을 생성하려면 [방정식에 대한 블록 생성]을 선택해야 합니다. 예를 들어 회소 시나리오 차원의 멤버인 실제에 대한 데이터가 없는 경우 2002Forecast = Actuals \* 1.05 할당을 수행하려면 [방정식에 대한 블록 생성]을 선택해야 합니다.
  - **2단계 계산** - 특정 멤버를 다시 계산합니다.  
 [2단계 계산]을 선택하면 기본 계산 후에 2단계로 태그가 지정된 멤버가 다시 계산됩니다. 2단계 태그는 계정으로 태그가 지정된 차원의 멤버 및 임의 차원의 동적 계산과 동적 계산 및 저장 멤버에 적용됩니다.
- **데이터 검색 버퍼**
  - **버퍼 크기** - 검색 버퍼 크기입니다. 스프레드시트 추가 기능 및 보고서 스크립트에서 검색을 처리하고 최적화하는 데 사용됩니다.
  - **정렬 버퍼 크기** - 검색 정렬 버퍼 크기입니다.
- **저장영역 - 블록 저장영역 애플리케이션에만 해당**
  - **현재 I/O 액세스 모드** - 현재 액세스 모드
  - **대기 중인 I/O 액세스 모드** - 기본적으로 다음 옵션 중 하나가 구성되어 있습니다.
    - \* **버퍼된 I/O** - 파일 시스템 버퍼 캐시를 사용합니다. 데이터베이스를 생성할 때 essbase.cfg 파일의 방향 설정에 대해 직접 I/O가 지정되지 않은 경우 버퍼된 I/O가 기본값입니다.
    - \* **직접 I/O** - 파일 시스템 버퍼 캐시를 건너뛰고 겹치는 비동기 I/O를 수행하므로 더 빠른 응답 시간이 제공되고 캐시 크기를 최적화하는 잠재성이 증가합니다. 직접 I/O가 선택된 경우 데이터베이스를 시작할 때마다 Essbase에서 직접 I/O를 사용하려고 합니다. 직접 I/O를 사용할 수 없는 경우 Essbase에서 버퍼된 I/O를 사용합니다. 운영

체제에서 제공하는 대기 없음(비동기) I/O 또는 캐시 메모리 잠금을 사용하려면 직접 I/O를 선택합니다.

- 데이터 압축 - 기본적으로 다음 옵션 중 하나가 구성되어 있습니다.
  - \* **비트맵 인코딩** - 비트맵은 데이터 셀을 나타내는 데 사용됩니다. 비트맵, 블록 머리글 및 기타 제어 정보만 디스크에 저장됩니다. 비트맵 인코딩은 가장 효율적인 데이터 압축 방법입니다. Essbase는 누락되지 않은 값만 저장하고 반복 값이나 0 값을 압축하지 않습니다. 데이터베이스가 데이터 블록을 데이터 캐시로 가져오는 경우 비트맵을 사용하여 누락된 값을 다시 생성하고 블록을 완전히 확장합니다.
  - \* **RLE(Run-Length Encoding)** - 0을 포함하여 연속하는 반복 값이 압축되며, 각 반복 값과 연속해서 반복되는 횟수에 대한 레코드가 유지됩니다. 평균 블록 밀도가 3%보다 크지 않거나 데이터베이스에 연속하는 많은 0 값이나 0이 아닌 연속하는 반복 값이 포함된 경우 RLE를 사용하는 것이 좋습니다.
  - \* **ZLIB** - 압축되는 데이터를 기반으로 하는 데이터 사전이 생성됩니다. 일반적으로 데이터가 매우 밀집한 경우 ZLIB 압축 시 최상의 압축 비율이 제공됩니다. 그러나 다른 압축 방법에서 더 나은 결과가 생성되는 경우도 있습니다. ZLIB 압축을 사용할 경우 저장된 저장영역 공간에 누락된 셀 수나 동일한 값의 연속 셀 수에 대한 관계가 거의 없거나 전혀 없습니다.
  - \* **압축 없음** - 데이터가 압축되지 않습니다.

## 차원 등록정보

데이터베이스에 대한 차원 정보에는 다음 내용이 포함되어 있습니다.

- 데이터베이스의 차원 수
- (블록 저장영역 데이터베이스에만 해당) 차원 유형(밀집 또는 희소)
- 차원의 멤버
- 저장된 멤버
- (집계 저장영역 데이터베이스에만 해당) 각 차원의 레벨 수



주:

차원 등록정보는 읽기 전용입니다.

## 통계 등록정보



주:

통계 등록정보는 읽기 전용입니다.

## 집계 저장영역 애플리케이션에 대한 통계

- **일반** - 다음과 같은 일반 통계 정보

- 데이터베이스 시작 시간 - 데이터베이스 서버 시간대 기준 시작 시간
- 데이터베이스 경과 시간 - 시간:분:초 형식의 경과 시간
- 연결 수 - 연결된 사용자 수
- 집계 저장영역 통계 - 다음과 같은 집계 데이터베이스에 대한 저장영역 통계
  - 애플리케이션에 있는 각 차원의 레벨 수와 레벨을 저장하는 데 사용한 비트 수(집계 저장영역 데이터베이스에서 일부 차원 레벨은 저장되지 않음)
  - 최대 키 길이(비트) - 모든 차원에서 사용된 모든 비트의 합계입니다. 예를 들어, 모든 차원의 키에는 20비트가 포함되며, 처음 네 비트는 연도 차원에 사용됩니다.
  - 최대 키 길이(바이트) - 셀당 키에서 사용한 바이트 수
  - 입력 레벨 셀의 수 - 모두 레벨 0 셀인 경우 사용자가 데이터를 입력할 수 있는, 공식이 포함되지 않은 레벨 0 교차 횡차원의 셀 수
  - 증분식 데이터 슬라이스 수 - 측시가 아니라 증분식으로(필요한 경우에만) 계산할 수 있는 데이터 교차 수
  - 증분식 입력 셀 수 - 측시가 아니라 증분식으로(필요한 경우에만) 계산할 수 있는 입력 셀 수
  - 집계 뷰 수 - 집계 셀이 포함된 뷰 수
  - 집계 셀 수 - 하위 레벨 값에서 롤업되므로, 요청 또는 검색될 때 계산해야 하는 셀 수입니다. 요청될 때마다 집계 셀 값을 계산하거나, 미리 계산한 다음 디스크에 저장할 수 있습니다.
  - 증분식 집계 셀 수 - 필요한 경우에만 업데이트할 수 있는 집계 셀 수
  - 증분식 데이터 쿼리 비용(총 비용에 대한 비율) - 연결된 집계 뷰에서 값을 검색하는 데 소요되는 평균 시간
  - 입력 레벨 데이터 크기(KB) - 모든 레벨 0 셀에 있는 데이터의 크기(KB)
  - 집계 데이터 크기(KB) - 모든 집계 셀의 집계 데이터 크기(KB)
- 런타임 - 다음과 같은 런타임 통계
  - 캐시 적중률 - 디스크에서 정보를 검색하는 데 비해 캐시에서 정보를 찾는 데 성공한 비율
  - 현재 캐시 크기 - 동적으로 생성된 캐시 크기
  - 현재 캐시 크기 한계(KB) - 캐시 크기의 한계(KB)
  - 마지막 시작 이후 읽은 페이지 수 - (자동으로 또는 사용자가) 애플리케이션을 시작한 이후로 읽은 인덱스 페이지 수
  - 마지막 시작 이후 쓴 페이지 수 - (자동으로 또는 사용자가) 애플리케이션을 시작한 이후로 업데이트한 인덱스 페이지 수
  - 페이지 크기(KB) - 페이지 크기(KB)
  - 데이터에 할당된 디스크 공간(KB) - 데이터 저장영역에 할당된 하드 디스크 공간의 총 크기(KB)
  - 데이터에서 사용한 디스크 공간(KB) - 데이터 저장영역에 사용된 디스크 공간의 총 크기(KB)
  - 할당된 임시 디스크 공간(KB) - 데이터 저장영역에 할당된 임시 디스크 공간의 총 크기
  - 사용된 임시 디스크 공간(KB) - 데이터 저장영역에 사용된 임시 디스크 공간의 총 크기

 주:

디스크 공간은 Default 테이블 영역에 사용된 공간이고, 임시 디스크 공간은 Temp 테이블 영역에 사용된 공간입니다. 두 경우 모두 몇몇 파일에서 일부 공간이 사용되지 않을 수 있습니다.

## 블록 저장영역 애플리케이션에 대한 통계

- 일반 - 다음은 일반 통계 정보입니다.
  - 데이터베이스 시작 시간 - 데이터베이스 서버 시간대 기준 시작 시간
  - 데이터베이스 경과 시간 - 시간:분:초 형식의 경과 시간
  - 연결 수 - 연결된 사용자 수
- 블록 - 다음은 블록 저장영역 데이터베이스의 데이터 블록에 대한 통계입니다.
  - 기존 블록 수 - 존재하는(데이터가 포함된) 총 블록 수
  - 블록 크기 - 각 확장된(압축 해제된) 데이터 블록의 크기(셀 개수 \* 8, 8KB에서 100KB 사이가 적합). 블록 크기를 변경하려면 데이터베이스의 밀집-희소 구성을 변경해야 합니다.
  - 잠재적 블록 수 - 최대 블록 수입니다(한 희소 차원의 멤버 수에 다른 희소 차원의 멤버 수를 곱하여 파생됨). 예를 들어 Sample Basic 데이터베이스에는 19개의 제품 멤버와 25개의 시장 멤버가 포함됩니다(공유 또는 레이블 전용 멤버는 계산하지 않음). 제품 및 시장은 데이터를 저장하는 희소 차원이기 때문에  $19 \times 25 = 475$ 개의 잠재적 데이터 블록이 있습니다.
  - 기존 레벨 0 블록 - 존재하는(데이터가 포함된) 레벨 0 블록(희소 차원 멤버에 하위 멤버가 없는 블록)의 총 수. 상위 레벨에서 데이터를 로드할 수 있기 때문에 데이터 입력에서 생성된 블록과 레벨 0 블록이 반드시 동일한 것은 아닙니다.
  - 기존 상위 레벨 블록 - 존재하는(데이터가 포함된) 레벨 0이 아닌 총 블록 수. 상위 레벨 블록에는 상위 레벨 희소 멤버의 모든 조합과 레벨 0 희소 멤버가 있는 상위 레벨 조합이 포함됩니다.
  - 블록 밀도(%) - 기존 데이터 블록의 샘플을 기준으로 각 데이터 블록 내 데이터 포인트의 평균 백분율 채우기. 밀집-희소 구성은 블록 밀도를 최대화해야 합니다. 그러나 블록 밀도를 최대화하면 데이터 블록이 확장될 수 있습니다. 블록 크기 및 블록 확장 고려 사항 때문에 블록 밀도를 최대화하려는 시도가 제대로 작동하지 않을 수 있습니다.
  - 기존 최대 블록 백분율 - 존재하는 블록 수와 잠재적 블록 수를 비교하는 백분율. 백분율은 데이터베이스의 스파스 정도를 측정된 값입니다. 백분율은 매우 작은 경우가 많습니다. 예를 들어 1퍼센트보다 작습니다.
  - 압축 비율 - 디스크에 저장된 블록의 압축 효율성 측정항목. 일반적으로 압축 비율은 블록 밀도를 나타냅니다.
  - 평균 클러스터링 비율 - 데이터(.pag) 파일의 조각 레벨. 최대값 1은 조각 없음을 나타냅니다. 검색, 계산 또는 데이터 로드 성능이 저하되었으며 클러스터링 비율 값이 1보다 작은 경우 데이터를 내보내고 다시 로드하여 데이터 파일을 강제로 재작성하는 것이 좋습니다. 파일을 재작성하면 파일 조각 모음이 발생하여 클러스터링 비율이 1에 더 가까워집니다.
  - 평균 조각화 몫 - 데이터베이스의 사용 가능한 공간입니다. 예를 들어 평균 조각화 몫 값이 3.174765이면 데이터베이스의 3%가 조각화되어 있으며 사용 가능한



공간이 있다는 것을 의미합니다. 데이터를 업데이트하고 계산할 때 블록이 더 이상 원래 공간에 맞지 않는 경우 빈 공간이 발생하여 파일 끝에 추가되거나 충분히 큰 다른 빈 공간에 맞게 됩니다. 숫자가 높을수록 빈 공간 수가 많아지므로 특정 레코드를 가져오는 데 걸리는 시간이 길어집니다. 평균 조각화 몫은 재구조화를 수행해야 하는지 결정하는 데 도움이 됩니다.

• **실행 시간**

– **인덱스 캐시의 적중률** - 디스크에서 인덱스 페이지를 검색할 필요 없이 인덱스 캐시에서 인덱스 정보를 찾는 데 성공한 비율

– **데이터 캐시 적중률** - 데이터 파일 캐시에서 블록을 검색할 필요 없이 데이터 캐시에서 데이터 블록을 찾는 데 성공한 비율

적중률이 늘어나면 디스크에서 검색하는 정보가 줄어드므로 성능이 향상됩니다. 적중률이 1.0이면 디스크에서 데이터를 검색한 적이 없으므로 최대 성능을 표시합니다.

Oracle에서는 작은 단위로 증분하여 캐시에 메모리를 할당하도록 권장합니다. 소형 및 대형 할당의 이점이 동일할 수 있지만, 일반적으로 대형 할당을 수행하면 적중률에 미치는 영향이 제한됩니다.

– **데이터 파일 캐시 적중률** - 데이터베이스가 디스크에서 블록을 검색할 필요 없이 데이터 캐시에서 데이터 블록을 찾는 데 성공한 비율

– **인덱스 페이지 읽기 수** - 데이터 블록을 찾기 위해 읽은 인덱스 페이지 수

– **인덱스 페이지 쓰기 수** - 데이터 블록을 업데이트하기 위해 쓴 인덱스 페이지 수

인덱스(또는 인덱스 파일)는 블록 저장영역 데이터 검색 정보를 저장하고, 디스크에 있으며, 인덱스 페이지를 포함합니다. 인덱스 페이지에는 데이터 블록에 대한 포인터(인덱스 항목이라고 함)가 포함되어 있습니다.

– **데이터 블록 읽기 수** - 읽은 데이터 블록 수

– **데이터 블록 쓰기 수** - 업데이트된 데이터 블록 수

자습서 비디오:



[BSO 데이터베이스 등록정보 관리](#)

## 트랜잭션 등록정보



**주:**

트랜잭션 등록정보는 블록 저장영역 데이터베이스에만 적용됩니다.

트랜잭션 탭에는 데이터베이스 액세스에 대한 정보가 표시됩니다.

- **커밋 액세스**를 사용하면 트랜잭션이 완료되고 커밋될 때까지 트랜잭션에서 트랜잭션과 관련된 모든 데이터 블록에 대해 읽기/쓰기 잠금을 유지할 수 있습니다. 다음 동시성 옵션은 미리 구성될 수도 있습니다.
  - **대기(초)** - 트랜잭션이 잠긴 데이터 블록에 액세스하기 위해 대기하는 시간(초)입니다. 기본값은 20초이나 무한 또는 **대기 없음**과 같은 다른 값이 미리 구성될 수도 있습니다.

- 사전 이미지 액세스 - 다른 동시 트랜잭션 기간 동안 잠기는 데이터 블록에 대한 읽기 전용 액세스 권한이 사용자에게 있습니다.
- 커밋없는 액세스를 사용하면 트랜잭션에서 블록 단위로 읽기/쓰기 잠금을 유지할 수 있습니다(기본 설정). 동기점이 미리 구성될 수 있는 영역은 다음과 같습니다.
  - 블록 커밋 - Oracle Essbase에서 커밋을 수행하기 전에 업데이트된 데이터 블록 수
  - 행 커밋 - Essbase에서 커밋을 수행하기 전에 데이터 로드 중 처리된 데이터 파일의 행 수

## 수정 등록정보

수정 탭에는 데이터베이스에 대해 수행된 가장 최근 작업(아웃라인 업데이트, 데이터 로드 또는 계산)에 대한 정보가 표시됩니다.

- 작업 - 데이터 로드 또는 계산과 같은 작업 유형
- 사용자 - 작업을 수행한 사용자의 이름
- 시작 시간 - 데이터 잠금과 같은 준비 태스크를 비롯하여 태스크가 시작된 Essbase Server 기준 시간(태스크 기간은 애플리케이션 로그의 경과 시간 항목 참조)
- 종료 시간 - 작업이 종료된 Essbase Server 기준 시간
- 참고 - 선택적 설명

## 통화 등록정보



주:

통화 등록정보는 Oracle Essbase 블록 저장영역 데이터베이스에만 적용됩니다.

통화 등록정보는 다음과 같습니다.

- **통화 데이터베이스** - 등록정보를 보고 편집할 통화 데이터베이스입니다. 애플리케이션에서 사용할 수 있는 모든 통화 데이터베이스가 목록에 있습니다.
- **변환 방법** - 통화 변환 방법입니다.
- **기본 통화 유형 멤버** - 통화 변환에서 기본값으로 사용할 통화 유형 멤버입니다. 통화 데이터베이스에 있는 CurType 차원의 적합한 멤버를 지정할 수 있습니다.

통화 데이터베이스는 항상 다음 네 개의 차원으로 구성됩니다.

- **국가** - 각 현지 통화(예: 미국 달러, 캐나다 달러, 독일 유로) 환율을 저장합니다.
- **시간** - 일반적으로 주 데이터베이스의 시간 차원과 동일합니다. 이 차원을 사용하면 통화 데이터베이스가 시간별 통화 변동을 추적하고 주 데이터베이스의 각 시간 슬라이스를 정확하게 변환할 수 있습니다.
- **범주** - 주 데이터베이스에서 계정으로 태그가 지정된 차원의 멤버에 서로 다른 환율을 적용할 수 있습니다. 계정 차원에 대해 정의된 범주는 통화 데이터베이스의 통화 범주 차원에 멤버를 생성하는 데 사용됩니다. 예를 들어 하나의 환율 범주를 사용하여 총 이익과 순 이익을 변환하고 다른 계정은 다른 환율 세트를 사용해야 할 수 있습니다.

- **통화 파티션** - 다양한 통화 변환 시나리오를 허용합니다. 예를 들어 회사에서 실제 환율과 계획 환율을 저장할 수 있습니다. 시나리오 간에 데이터를 변환하려면 사용할 환율 유형을 선택합니다.

## 캐시 등록정보

### 주:

캐시 등록정보는 블록 저장영역 데이터베이스에만 적용됩니다.

캐시 탭은 다음 영역의 캐시 크기 및 캐시 잠금을 보여줍니다.

- **캐시 메모리 잠금** - 데이터베이스 관리자 권한이 있는 데이터베이스에 대해 캐시 메모리 잠금을 사용으로 설정합니다.  
메모리 잠금을 사용하면 시스템 메모리 관리자가 Oracle Essbase 캐시의 메모리를 스왑하고 예약할 필요가 없으므로 데이터베이스 성능이 향상될 수 있습니다. 데이터베이스를 중지하고 재시작하면 메모리 잠금 변경사항이 적용됩니다.
- **인덱스 페이지 설정(KB)** - 인덱스 페이지의 크기(8KB 및 변경할 수 없음)
- **인덱스 페이지 현재 값(KB)** - 데이터베이스에 데이터가 포함된 경우 인덱스 페이지 설정 값
- **캐시 크기** - 인덱스, 데이터 파일 및 데이터 캐시에 할당되는 메모리 크기를 결정하는 데 도움이 되는 다양한 버퍼의 크기  
기본값, 최소값 및 권장 값은 *Oracle Essbase Database Administrator's Guide*를 참조하십시오.
- **인덱스 캐시 설정(KB)** - 인덱스에 사용할 메모리 크기(KB)
- **인덱스 캐시 현재 값(KB)** - 인덱스에 현재 사용 중인 메모리 크기
- **데이터 파일 캐시 설정(KB)** - 압축된 데이터(.pag) 파일을 보관할 메모리 크기(KB)  
(데이터베이스에서 버퍼된 I/O를 사용 중인 경우 해당되지 않음)
- **데이터 파일 캐시 현재 값(KB)** - 데이터 파일에 사용 중인 메모리 크기
- **데이터 캐시 설정(KB)** - 압축되지 않은 데이터 블록을 보관하는 데 사용 중인 메모리 크기(KB)
- **데이터 캐시 현재 값(KB)** - 데이터 캐시에 사용 중인 메모리 크기

## MaxL 스크립트 작성 및 실행

MaxL 스크립트 편집기를 사용하여 Oracle Hyperion Calculation Manager 내에서 Oracle Essbase 관리 태스크를 자동화합니다. 이 편집기를 사용하여 데이터베이스 아웃라인과 메타데이터를 정의할 수 있습니다.

MaxL 스크립트 편집기에서는 MaxL 스크립트를 입력할 수 있는 텍스트 편집 창을 제공합니다.

MaxL 스크립트 편집기 내에서 다음 MaxL 관련 태스크를 수행할 수 있습니다.

- MaxL 스크립트를 생성 및 실행하여 Essbase 관리 태스크를 자동화합니다.
- MaxL 명령문을 대화식으로 입력 및 실행하여 한 번에 하나 이상의 Essbase 작업을 수행합니다.
- 스크립트 실행 결과를 봅니다.

MaxL 스크립트를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 데이터베이스 등록정보 아이콘을 누릅니다.
2. 엔터프라이즈 뷰에서 Oracle Hyperion Planning 애플리케이션 유형, MaxL 스크립트를 생성할 애플리케이션 순으로 확장합니다.
3. 애플리케이션을 마우스 오른쪽 버튼으로 누르고 **MaxL 스크립트 편집기**를 선택합니다. MaxL 스크립트 편집기가 표시됩니다.
4. MaxL 스크립트 편집기에서 스크립트 콘텐츠를 입력합니다.
5. 실행 버튼을 눌러 스크립트를 실행합니다.

스크립트의 콘텐츠는 편집기 내에서 대화식으로 입력, 실행하고 지우는 개별 명령문이 될 수 있습니다. 편집기 내에서 명령문을 실행하기 위해 스크립트를 저장할 필요가 없습니다.

## 데이터베이스 객체에서 잠금 제거

보안 권한에 따라 객체를 보고 잠금 해제할 수 있습니다.

관리자 권한이 있는 사용자는 객체를 잠금 해제할 수 있습니다. 관리자 권한이 없는 사용자는 자신이 잠근 객체만 잠금 해제할 수 있습니다.

서버에서는 계산 스크립트, 보고서 스크립트, 비즈니스 규칙 파일 등의 데이터베이스 객체에 대해 체크아웃 기능을 사용하여 한 번에 한 사용자만 객체를 수정하도록 합니다.

기본적으로 객체를 열어서 수정할 때 객체가 잠기고 객체를 닫을 때 잠금 해제됩니다.

Oracle Hyperion Calculation Manager의 객체에 대해 작업을 수행하는 경우 해당 객체가 잠길 수 있습니다. 객체 잠금을 해제하려면 엔터프라이즈 뷰에서 객체 잠금을 해제할 계획 유형을 선택합니다.

데이터베이스에서 잠금을 제거하려면 다음을 수행합니다.

1. 시스템 뷰에서 데이터베이스 등록정보 아이콘을 누릅니다.
2. 엔터프라이즈 뷰에서 Oracle Hyperion Planning 애플리케이션 유형과 잠금을 제거할 데이터베이스가 포함된 애플리케이션을 확장합니다.
3. 데이터베이스를 마우스 오른쪽 버튼으로 누르고 **잠금 제거**를 선택합니다.

## 애플리케이션 시작 및 중지

읽기 이상의 권한이 있는 애플리케이션을 시작할 수 있습니다. Oracle Essbase는 새로 시작된 애플리케이션을 Essbase Server의 메모리에 로드합니다.

해당하는 상위 애플리케이션이 시작될 때 데이터베이스가 시작되도록 지정할 수 있습니다. 이 경우 애플리케이션을 시작한 다음에 사용자가 데이터베이스에 연결하면 애플리케이션과 모든 관련 데이터베이스가 메모리에 있으므로 데이터베이스 연결 시 처음에 성능이 더 나을 수 있습니다.

애플리케이션을 중지하면 Essbase는 애플리케이션 및 애플리케이션 내의 모든 데이터베이스를 Essbase Server의 메모리에서 언로드합니다. 따라서 사용할 수 있는 메모리가 증가합니다. 애플리케이션 내의 데이터베이스가 손상되지 않도록 애플리케이션을 올바르게 중지해야 합니다.

애플리케이션을 시작하거나 중지하려면 다음을 수행합니다.

1. 시스템 뷰에서 데이터베이스 등록정보 아이콘을 누릅니다.
2. 엔터프라이즈 뷰에서 Oracle Hyperion Planning 애플리케이션 유형과 시작하거나 중지할 애플리케이션을 확장합니다.
3. 애플리케이션을 마우스 오른쪽 버튼으로 누르고 다음을 선택합니다.
  - 애플리케이션 시작
  - 애플리케이션 중지

 주:

데이터베이스 설정을 변경할 때마다 애플리케이션을 중지한 후 다시 시작해야 합니다.

## 데이터베이스 시작 및 중지

데이터베이스를 시작하면 Oracle Essbase는 데이터베이스를 Essbase Server의 메모리에 로드합니다.

인덱스 캐시는 자동으로 할당되고 데이터 파일 및 데이터 캐시는 블록이 요청될 때 할당됩니다. 데이터베이스를 시작한 다음에 사용자가 데이터베이스에 액세스하면 데이터베이스가 메모리에 있으므로 연결 시 처음에 성능이 더 나을 수 있습니다.

시작되지 않은 애플리케이션에서 데이터베이스를 시작하면 애플리케이션과 애플리케이션 내의 모든 데이터베이스가 로드됩니다. 하나의 데이터베이스나 애플리케이션에 대한 모든 데이터베이스를 시작할 수 있습니다.

데이터베이스를 중지하면 Essbase는 데이터베이스를 Essbase Server의 메모리에서 언로드하고 업데이트된 데이터를 디스크에 커밋합니다. 따라서 서버 컴퓨터에서 사용 가능한 메모리가 증가합니다.

하나의 데이터베이스나 애플리케이션에 대한 모든 데이터베이스를 중지할 수 있습니다.

 주:

블록 저장영역 애플리케이션 데이터베이스를 시작 및 중지할 수 있습니다.

데이터베이스를 시작하거나 중지하려면 다음을 수행합니다.

1. 시스템 뷰에서 데이터베이스 등록정보 아이콘을 누릅니다.
2. 엔터프라이즈 뷰에서 Oracle Hyperion Planning 애플리케이션 유형과 시작하거나 중지할 데이터베이스가 포함된 애플리케이션을 확장합니다.
3. 데이터베이스를 마우스 오른쪽 버튼으로 누르고 다음을 선택합니다.
  - 데이터베이스 시작
  - 데이터베이스 중지

애플리케이션에 대한 모든 데이터베이스를 시작하거나 중지하려면 다음을 수행합니다.

1. 시스템 뷰에서 데이터베이스 등록정보 아이콘을 누릅니다.

2. **엔터프라이즈 뷰**에서 Planning 애플리케이션 유형과 시작하거나 중지할 데이터베이스가 포함된 애플리케이션을 확장합니다.
3. 애플리케이션을 마우스 오른쪽 버튼으로 누르고 다음을 선택합니다.
  - 모든 데이터베이스 시작
  - 모든 데이터베이스 중지

## 애플리케이션 로그 파일 삭제

데이터베이스 서버는 서버와 애플리케이션에서 발생하는 작업을 확장자가 .log인 텍스트 파일에 쓰거나 로깅합니다. 서버의 각 Oracle Hyperion Planning 애플리케이션에는 해당 애플리케이션을 따라 이름 지정된 고유한 애플리케이션 로그가 있습니다. 애플리케이션 로그의 정보는 오류가 발생한 위치와 이유를 파악하는 데 도움이 됩니다. 이러한 애플리케이션 레벨 로그 파일을 Oracle Hyperion Calculation Manager에서 삭제할 수 있습니다.

애플리케이션 로그의 정보는 다음을 평가하는 데 도움이 됩니다.

- 특정 작업을 수행한 사람
- 작업이 수행된 시간
- 작업을 수행하거나 시도할 때 발생한 오류
- 최적화에 도움이 되는 차원 및 멤버에 대한 정보
- 아티팩트가 서버 인스턴스에 있는 경우 작업을 수행하는 데 사용된 아티팩트(예: 계산을 수행하는 데 사용된 계산 스크립트)의 이름

애플리케이션 레벨 로그 파일을 삭제하려면 다음을 수행합니다.

1. **시스템 뷰**에서 **데이터베이스 등록정보** 아이콘을 누릅니다.
2. **엔터프라이즈 뷰**에서 Planning 애플리케이션 유형, 삭제할 로그 파일이 포함된 애플리케이션 순으로 확장합니다.
3. 애플리케이션을 마우스 오른쪽 버튼으로 누르고 **로그 삭제**를 선택합니다.
4. 로그 삭제 여부를 확인합니다.

Calculation Manager에서 로그 파일이 성공적으로 삭제되었는지 여부를 알리는 메시지를 표시합니다.

## 데이터베이스 재구조화

밀집 차원에 멤버를 추가하는 등 데이터베이스를 재구조화하는 경우 데이터 블록을 다시 계산해야 할 수도 있습니다.

Oracle Essbase는 모든 데이터 블록을 더티로 표시합니다. 재구조화된 데이터베이스를 계산하는 경우 모든 블록이 계산됩니다.



**주:**

블록 저장영역 애플리케이션 데이터베이스를 재구조화할 수 있습니다.

다음과 같은 경우 데이터베이스를 재구조화해야 합니다.

- 밀집 차원 동적 계산 및 저장 멤버를 추가, 삭제 또는 이동합니다.
- 밀집 차원 동적 계산 및 저장 멤버를 동적 계산 멤버로 변경합니다.
- 밀집 차원 동적 계산 멤버를 동적 계산 및 저장 멤버로 변경합니다.
- 밀집 차원에 있는 비동적 멤버의 저장영역 등록정보를 동적 계산으로 변경합니다.
- 밀집 차원의 저장영역 등록정보를 동적 계산 멤버에서 비동적 값으로 변경합니다.
- 희소 차원에 있는 비동적 멤버의 저장영역 등록정보를 동적 계산 또는 동적 계산 및 저장으로 변경합니다.

데이터베이스를 재구조화하려면 다음을 수행합니다.

1. 시스템 뷰에서 데이터베이스 등록정보 아이콘을 누릅니다.
2. 엔터프라이즈 뷰에서 Oracle Hyperion Planning 애플리케이션 유형, 애플리케이션, 데이터베이스를 재구조화할 계획 유형을 확장합니다.
3. 계획 유형을 마우스 오른쪽 버튼으로 누르고 데이터베이스 재구조화를 선택합니다.
4. 데이터베이스 재구조화 여부를 확인합니다.

Oracle Hyperion Calculation Manager에서 데이터베이스가 성공적으로 재구조화되었는지 여부를 알리는 메시지를 표시합니다.

## 아웃라인 확인

Oracle Essbase 아웃라인에 오류가 있는지 확인할 수 있습니다.

아웃라인 확인 프로세스에서는 아웃라인 유형(집계 저장영역 또는 블록 저장영역)을 고려하여 각 유형의 규칙에 따라 아웃라인을 확인합니다. 아웃라인에 오류가 없으면 멤버 공식을 확인합니다.

아웃라인을 확인하려면 다음을 수행합니다.

1. 시스템 뷰에서 데이터베이스 등록정보 아이콘을 누릅니다.
2. 엔터프라이즈 뷰에서 Planning 애플리케이션 유형, 애플리케이션, 아웃라인을 확인할 계획 유형 순으로 확장합니다.
3. 계획 유형을 마우스 오른쪽 버튼으로 누르고 아웃라인 확인을 선택합니다.

아웃라인을 확인하는 경우 Essbase에서 다음 사항을 확인합니다.

- 모든 멤버 및 별칭 이름이 적합합니다. 멤버 및 별칭은 다른 멤버, 별칭, 세대 또는 레벨과 이름이 동일할 수 없습니다.
- 하나의 차원만 계정, 시간, 통화 유형 또는 국가로 태그 지정되어 있습니다.
- 공유 멤버가 적합합니다.
- 레벨 0 멤버가 레이블 전용으로 태그 지정되어 있지 않습니다.
- 레이블 전용 멤버에 공식이 지정되어 있지 않습니다.
- 통화 범주 및 통화 이름이 통화 아웃라인에 적합합니다.
- 희소 차원의 동적 계산 멤버에 있는 1차 하위 멤버가 100개 이하입니다.
- 상위 멤버의 1차 하위 멤버가 하나이고 해당 1차 하위 멤버가 동적 계산 멤버인 경우 상위 멤버도 동적 계산이어야 합니다.
- 상위 멤버의 1차 하위 멤버가 하나이고 해당 1차 하위 멤버가 동적 계산 2단계 멤버인 경우 상위 멤버도 동적 계산 2단계여야 합니다.

- 부울 속성 차원 멤버 이름 두 개가 아웃라인에 대해 정의된 부울 속성 차원 멤버 이름 두 개와 동일합니다.
- 날짜 속성 차원 레벨 0 멤버 이름은 날짜 형식 이름 설정(mm-dd-yyyy 또는 dd-mm-yyyy)과 일치해야 합니다. 차원에 멤버가 없는 경우 차원 이름이 레벨 0 멤버이므로 차원 이름이 설정과 일치해야 합니다.
- 숫자 속성 차원 레벨 0 멤버 이름은 숫자 값입니다. 차원에 멤버가 없는 경우 차원 이름이 레벨 0 멤버이므로 차원 이름이 숫자 값이어야 합니다.
- 속성 차원은 모든 표준 차원 다음인, 아웃라인 끝에 있습니다.
- 표준 차원의 레벨 0 동적 계산 멤버에는 공식이 있습니다.
- 멤버의 공식이 적합합니다.
- 하이브리드 분석 아웃라인에서는 차원의 레벨 0 멤버만 하이브리드 분석을 사용할 수 있습니다.

또한, Essbase는 아웃라인 확인 중 다음과 같이 적절한 숫자 속성 차원 멤버 이름으로 변환을 수행하여 아웃라인에 표시합니다.

- 멤버 이름의 빼기 기호를 이름 앞에서 뒤로 이동합니다. 예를 들어 -1이 1-가 됩니다.
- 멤버 이름에서 선행 또는 후행 0을 제거합니다. 예를 들어 1.0은 1이 되고 00.1은 0.1이 됩니다.

## 데이터베이스 설정 최적화에 유용한 시뮬레이션 생성

데이터베이스 성능은 많은 매개변수에 따라 달라집니다. 다양한 데이터베이스 매개변수의 최적 값을 얻기 위해 데이터베이스에 각기 다른 매개변수 값 조합을 적용하여 시뮬레이션을 생성하고 성능 통계 변화를 확인할 수 있습니다. Oracle Hyperion Calculation Manager에서는 각 시뮬레이션에서 얻은 성능 통계를 분석하고 이러한 성능 통계가 포함된 보고서를 생성하는 옵션을 제공합니다.

### 주:

블록 저장영역 애플리케이션 데이터베이스에 대한 시뮬레이션을 생성할 수 있습니다.

시뮬레이션 및 시뮬레이션 보고서를 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서 데이터베이스 등록정보 아이콘을 누릅니다.
2. 엔터프라이즈 뷰에서 Oracle Hyperion Planning 애플리케이션 유형과 시뮬레이션을 생성할 데이터베이스가 포함된 애플리케이션을 확장합니다.
3. 계획 유형을 마우스 오른쪽 버튼으로 누르고 DB 설정 조정을 선택합니다.
4. 태스크 선택에서 아웃라인 등록정보에 대한 새 시뮬레이션 생성을 선택하고 다음을 누릅니다.
5. 시나리오 초기화에서 다음 태스크를 수행합니다.
  - a. 실행할 반복 횟수를 입력합니다. 최소값은 1이고 최대값은 6입니다.
  - b. 각 반복 전에 실행할 규칙을 입력하거나 선택합니다.
  - c. 각 반복 중에 실행할 규칙을 입력하거나 선택합니다.



- d. 각 반복 *후에* 실행할 규칙을 입력하거나 선택합니다.
  - e. 다음 오류 메시지 레벨 중 하나를 선택하여 시뮬레이션 중 로그 파일에 쓸 오류 유형을 지정합니다.
    - 정보 - 시뮬레이션 중 서버에서 수행하는 일상적인 작업을 캡처합니다.
    - 경고 - 서버에서 심각한 것으로 간주되지 않는 상태를 캡처합니다.
    - 오류 - 시뮬레이션을 수행하는 동안 발생한 오류를 캡처합니다.
    - 세부정보 - 시뮬레이션 중 계산 통계에 대한 자세한 정보를 캡처합니다.
    - 요약 - 시뮬레이션 중 계산 통계 요약만 캡처합니다.
    - 없음 - 시뮬레이션 중 정보 또는 오류를 캡처하지 않습니다.
  - f. 각 시뮬레이션 후에 새 데이터베이스를 생성할지 여부를 선택합니다.
  - g. 다음을 누릅니다.
6. 반복 매개변수 설정에서 다음 매개변수의 초기 값과 증분 값을 설정합니다.
- 데이터 캐시 설정: 압축되지 않은 데이터 블록을 보관하는 메모리 내의 버퍼입니다.
  - 인덱스 캐시 설정: 인덱스 페이지를 보관하는 메모리 내의 버퍼입니다. 메모리에서 동시에 유지되는 인덱스 페이지 수는 캐시에 할당된 메모리 크기에 따라 달라집니다.
  - 버퍼 크기: 데이터베이스 버퍼는 평가 전에 추출된 행 데이터 셀을 보관하는 데이터베이스당 서버 버퍼입니다. 각 데이터베이스에는 변경할 수 있는 검색 버퍼 설정 (KB)이 있습니다. 기본 버퍼 크기는 32비트 플랫폼의 경우 10KB이고 64비트 플랫폼의 경우 20KB입니다. 버퍼 크기를 늘리는 경우 크기 제한은 100,000KB로 설정되어 있지만 100KB를 초과하지 않는 것이 좋습니다.
  - 정렬 버퍼 크기: 검색 중 정렬할 데이터를 보관하는 서버 버퍼의 크기(KB)를 지정합니다. 데이터베이스별로 버퍼 크기를 조정할 수 있습니다. 기본 버퍼 크기는 32비트 플랫폼의 경우 10KB로 설정되고 64비트 플랫폼의 경우 20KB로 설정됩니다.
7. 완료를 누릅니다.
- 검증 오류가 없으면 다음 정보가 포함된 **보고서 보기** 탭이 표시됩니다.
- **사용 가능한 통계** 영역에서 다음 정보를 볼 수 있습니다.
    - 런타임 통계: 이 탭에는 평균 클러스터, 반복당 인덱스 캐시 적중률 등 데이터베이스 관련 매개변수가 포함되어 있습니다.
    - 실행 시간 통계: 이 탭에는 각 반복에서 실행한 각 규칙의 실행 시간(초)이 표시됩니다.
  - **통계 차트** 영역에서는 각 반복에서 각 규칙을 실행하는 데 걸린 시간(초)의 라인, 막대 또는 세로 막대 그래프를 표시할 수 있습니다.
  - **요약** 영역의 **반복 세부정보** 탭에서는 데이터 및 인덱스 캐시 설정, 버퍼 크기 등 각 반복에 대해 설정한 매개변수의 세부정보를 검토할 수 있습니다.
 

차원 탭에서는 차원의 이름 및 유형(즉, 밀집 또는 희소), 각 차원의 멤버 수, 각 차원에 저장된 멤버 수 등 각 반복에서 사용된 차원 및 멤버에 대한 정보를 볼 수 있습니다.
  - **로그** 영역의 **차트** 탭에서는 각 반복에서 각 규칙에 대해 생성된 로그 메시지 유형의 선형, 막대 또는 열 그래프를 볼 수 있습니다.
 

로그 탭에서는 각 반복에서 각 규칙에 대해 생성된 로그 메시지(있는 경우)의 텍스트를 볼 수 있습니다. 규칙, 반복 및 메시지 레벨을 기준으로 로그 텍스트를 필터링할 수 있습니다.
8. **보고서 보기** 탭에 있는 동안 데이터베이스 매개변수를 변경하려면 **개요** 탭을 선택합니다. , 데이터베이스 매개변수를 수정한 다음 시뮬레이션을 다시 실행할 수 있습니다. 그러면 새 보고서가 표시됩니다.

9. 시뮬레이션 보고서를 익스포트하려면 [파일], [익스포트] 순으로 선택합니다. xml 파일의 이름을 입력합니다.
10. 시뮬레이션 보고서를 임포트하려면 **태스크 선택**에서 **시뮬레이션 보고서 임포트/로드**를 선택합니다.

## 데이터베이스에서 데이터 지우기


집계 저장영역 애플리케이션에서 데이터를 지우고, 블록 저장영역 애플리케이션에서 데이터 블록을 지웁니다.

### 참조:

- [집계 저장영역 애플리케이션에서 데이터 지우기](#)
- [블록 저장영역 애플리케이션에서 데이터 블록 지우기](#)


## 집계 저장영역 애플리케이션에서 데이터 지우기

집계 저장영역 애플리케이션에서 데이터를 지우려면 다음을 수행합니다.

1. 시스템 뷰에서 를 누릅니다.
2. 데이터베이스 등록정보 탭의 **엔터프라이즈** 뷰에서 집계 저장영역 애플리케이션을 확장합니다.
3. ASO 큐브를 마우스 오른쪽 버튼으로 누르고 **지우기**를 선택한 다음 옵션을 선택합니다.
  - **모든 데이터** - 데이터베이스에서 모든 데이터를 지웁니다.
  - **모든 집계** - 데이터베이스에서 모든 집계 데이터 값을 지웁니다.
  - **부분 데이터** - 집계 저장영역 데이터베이스의 지정된 영역에서 데이터를 지우고 다른 지역에 있는 데이터는 보존합니다.

부분 데이터를 지우도록 선택하는 경우 지울 영역을 정의하는 MDX 표현식을 입력하고 **논리적** 또는 **물리적**을 선택하여 데이터 지우기 유형을 지정합니다.

- **논리** - 지정된 영역의 데이터가 음의 보상 값이 있는 새 데이터 슬라이스에 작성되므로, 지울 셀의 값이 0이 됩니다.
- **물리적** - 지정된 영역의 데이터가 집계 저장영역 데이터베이스에서 물리적으로 제거됩니다.

멤버 선택기를 사용하여 MDX 구문을 생성하려면 을 누릅니다. **멤버 선택기** 대화상자에서 멤버를 선택하거나, 각 차원에서 함수를 사용한 다음 **확인**을 눌러 MDX 표현식을 생성합니다.

예를 들면 다음과 같습니다.

```
Crossjoin(Crossjoin(Crossjoin(Crossjoin(Crossjoin(Crossjoin(Crossjoin({ [NI].Levels(0)
{except(DESCENDANTS([Q1]), {[Q1]})), {[FY12], [FY13], [FY14]}},
{[Plan]}}, {[Working]}}, {[100].Levels(0).Members}},
{[P_TP].Levels(0).Members})), {[FY06]}

```

**확인**을 눌러 데이터를 지우는 경우 MDX 표현식이 검증되는지 확인합니다. MDX 표현식을 직접 수정하는 경우 실행하기 전에 검증됩니다.




주:

권한이 있는 데이터베이스에서만 데이터를 지울 수 있습니다.

## 블록 저장영역 애플리케이션에서 데이터 블록 지우기

블록 저장영역 애플리케이션에서 데이터 블록을 지우려면 다음을 수행합니다.

1. 시스템 뷰에서 를 누릅니다.
2. 데이터베이스 등록정보 탭의 엔터프라이즈 뷰에서 블록 저장영역 애플리케이션을 확장합니다.
3. BSO 큐브를 마우스 오른쪽 버튼으로 누르고 **지우기**를 선택한 다음 옵션을 선택합니다.
  - **모든 데이터** - 데이터베이스에서 모든 데이터를 지웁니다.
  - **상위 레벨 블록** - 상위 레벨 데이터 블록만 지웁니다. 상위 레벨 블록의 데이터 값은 #Missing으로 설정됩니다. 상위 레벨 블록은 상위 멤버인 회소 멤버가 하나 이상 포함된 회소 멤버 조합에 대해 생성됩니다.
  - **비입력 블록** - 계산에서 파생된 값이 포함된 데이터 블록(비입력 블록)만 지웁니다. 비입력 블록을 지우면 비입력(계산된) 셀의 데이터 값이 #Missing으로 설정됩니다.
  - **동적 블록** - 동적으로 계산 및 저장된 데이터 블록(동적 계산 및 저장 멤버)만 지웁니다.
  - **빈 블록** - 빈 데이터 블록(모든 값이 #Missing인 블록)만 지웁니다.



주:

권한이 있는 데이터베이스에서만 데이터를 지울 수 있습니다.

## 블록 저장영역 애플리케이션에 대한 위치 별칭 작업

위치 별칭은 데이터 소스에 대한 기술자입니다. 위치 별칭을 표시하고 익스포트할 수 있습니다.

참조:

- [블록 저장영역 애플리케이션에 대한 위치 별칭 작업](#)
- [위치 별칭 목록 표시](#)
- [위치 별칭 익스포트](#)
- [위치 별칭 생성](#)
- [위치 별칭 삭제](#)

## 블록 저장영역 애플리케이션에 대한 위치 별칭 작업


위치 별칭은 데이터 소스에 대한 기술자입니다. 위치 별칭은 데이터베이스의 별칭 이름을 해당 데이터베이스의 물리적 위치에 매핑합니다. 위치 별칭은 데이터베이스 또는 애플리케이션 레벨에서 설정되고 별칭, 서버, 애플리케이션, 데이터베이스, 사용자 이름 및 비밀번호를 지정합니다.

 주:

위치 별칭은 집계 저장영역 데이터베이스에 적용되지 않습니다.



## 위치 별칭 목록 표시

블록 저장영역 애플리케이션에 대한 위치 별칭 목록을 표시하려면 다음을 수행합니다.

1. 시스템 뷰에서  을 누릅니다.
2. 엔터프라이즈 뷰에서 블록 저장영역 애플리케이션을 마우스 오른쪽 버튼으로 누르고 위치 별칭을 선택합니다.


## 위치 별칭 익스포트

위치 별칭을 내보내려면 다음을 수행합니다.

1. 시스템 뷰에서  을 누릅니다.
2. 엔터프라이즈 뷰에서 블록 저장영역 애플리케이션을 마우스 오른쪽 버튼으로 누르고 위치 별칭을 선택합니다.
3. 위치 별칭 대화상자에서  를 누르고 내보낼 파일을 선택한 후 확인을 누릅니다.


## 위치 별칭 생성

위치 별칭을 생성하려면 다음을 수행합니다.

1. 시스템 뷰에서  을 누릅니다.
2. 엔터프라이즈 뷰에서 블록 저장영역 애플리케이션을 마우스 오른쪽 버튼으로 누르고 위치 별칭을 선택합니다.
3. 새 위치 별칭을 누릅니다.
4. 요청된 정보를 위치 별칭 대화상자에 입력합니다.

## 위치 별칭 삭제

위치 별칭을 삭제하려면 다음을 수행합니다.

1. 시스템 뷰에서  을 누릅니다.
2. 엔터프라이즈 뷰에서 블록 저장영역 애플리케이션을 마우스 오른쪽 버튼으로 누르고 위치 별칭을 선택합니다.
3. 위치 별칭 대화상자에서 위치 별칭을 마우스 오른쪽 버튼으로 누르고 삭제를 선택합니다.

## 집계 저장영역 데이터베이스에서 쿼리 추적 사용

쿼리 데이터를 사용하여 데이터베이스에 대해 구체화할 가장 적절한 집계 뷰 세트를 선택합니다.

쿼리 추적을 통해 데이터베이스에 대해 수행되는 각 쿼리의 비용에 대한 데이터를 캡처할 수 있습니다. 쿼리 비용은 뷰에서 값을 검색하는 데 필요한 평균 검색 시간의 추정치입니다. 첫 번째 뷰 (기본적으로 선택됨)에 대한 추정치는 모든 가능한 쿼리의 평균입니다. 쿼리 추적이 사용되는 뷰에 대한 추정치는 추적된 쿼리의 평균입니다. 따라서 뷰는 조건에 따라 다른 추정치를 표시할 수 있습니다. 특정 뷰를 사용하는 이점을 평가하는 백분율을 계산하려면 뷰에 대한 쿼리 비용 값을 레벨 0 값만 포함된 뷰 사용에 대한 쿼리 비용 값으로 나눕니다.

활성화된 쿼리 추적은 다음의 하나가 발생할 때까지 계속됩니다.


- 이 항목에 설명된 대로 데이터베이스에 대한 쿼리 추적이 비활성화됩니다.
- 애플리케이션이 종료됩니다. 애플리케이션이 종료되면 애플리케이션이 다시 시작될 때 쿼리 추적이 자동으로 다시 시작되지 않습니다.
- 데이터베이스에 대한 추가 집계 뷰가 구체화됩니다. 추가 뷰가 구체화될 때 쿼리 추적 데이터가 부적합해지므로 새 집계 뷰를 구체화하면 쿼리 추적이 재설정됩니다.

메모리에만 저장되는 쿼리 추적에는 Oracle Hyperion Web Analysis, 그리드 API, 보고서 스크립트, Java API 등의 쿼리가 포함됩니다.

### 주:

쿼리 추적은 집계 저장영역 데이터베이스에서만 사용할 수 있습니다.

쿼리 추적을 활성화 또는 비활성화하려면 다음을 수행합니다.

1. 시스템 뷰에서  을 누릅니다.
2. 엔터프라이즈 뷰에서 계획 유형을 마우스 오른쪽 버튼으로 누르고 쿼리 추적을 선택한 후 다음 옵션 중 하나를 선택합니다.
  - 사용 - 쿼리 추적을 활성화합니다.
  - 사용 안 함 - 쿼리 추적을 비활성화합니다.

쿼리 추적을 활성화하면 데이터베이스에서는 쿼리 정보를 기록하고, 쿼리 추적을 비활성화하면 데이터베이스에서는 쿼리 정보 기록을 중지하고 메모리에서 쿼리 데이터를 지웁니다.
3. 확인을 눌러 쿼리 추적을 활성화 또는 비활성화할지를 확인합니다.

## 집계 저장영역 데이터베이스 아웃라인 압축

파일을 압축하여 삭제된 멤버의 레코드를 제거하고 아웃라인 파일 크기를 줄입니다.

예를 들어, 집계 저장영역 아웃라인(.otl) 파일은 변경될 때(예: 멤버가 추가 또는 삭제될 때) 파일 크기가 늘어날 수 있습니다. 아웃라인 파일이 압축되고 나면 파일 크기가 이전처럼 계속 증가합니다.


아웃라인 파일을 압축하면 데이터베이스에서 아웃라인을 재구조화합니다. 아웃라인을 압축하면 데이터베이스에서는 데이터를 지우지 않습니다.

멤버가 아웃라인에서 삭제되면 아웃라인 파일에서 이 멤버의 해당 레코드는 삭제됨으로 표시되지만 레코드가 아웃라인 파일에 남아 있습니다. 아웃라인 파일을 압축해도 삭제된 멤버의 레코드는 제거되지 않습니다.

 **주:**

집계 저장영역 데이터베이스 아웃라인만 압축할 수 있습니다. 아웃라인 압축 프로세스는 다른 사용자 또는 프로세스가 데이터베이스를 활발히 사용하지 않을 때만 수행할 수 있습니다.

집계 저장영역 데이터베이스 아웃라인을 압축하려면 다음을 수행합니다.

1. 시스템 뷰에서  을 누릅니다.
2. 엔터프라이즈 뷰에서 아웃라인을 압축할 데이터베이스가 포함된 계획 유형을 마우스 오른쪽 버튼으로 누르고 **아웃라인 압축**을 선택합니다.
3. 확인을 눌러 데이터베이스 아웃라인을 압축할지를 확인합니다.  
아웃라인 압축 작업 상태 대화상자에 압축 진행 상태가 표시됩니다. 아웃라인이 압축되면 아웃라인 압축 작업 상태 대화상자에 성공 메시지가 표시됩니다.
4. 아웃라인 압축 작업 상태 대화상자에서 세부정보 표시를 눌러 아웃라인 압축의 세부정보를 확인하거나 확인을 눌러 대화상자를 닫습니다.

## 레벨 0 데이터 임포트 및 익스포트


ASO 및 BSO 큐브에서 레벨 0 데이터를 임포트 및 익스포트합니다.

**참조:**

- [ASO 큐브에서 레벨 0 데이터 임포트](#)
- [ASO 큐브에서 레벨 0 데이터 익스포트](#)
- [BSO 큐브에서 레벨 0 데이터 임포트](#)
- [BSO 큐브에서 레벨 0 데이터 익스포트](#)

## ASO 큐브에서 레벨 0 데이터 임포트

ASO 큐브에서 레벨 0 데이터를 임포트하려면 다음을 수행합니다.

1. 시스템 뷰에서  를 누릅니다.
2. 데이터베이스 등록정보 탭의 엔터프라이즈 뷰에서 집계 저장영역 애플리케이션을 확장합니다.
3. ASO 큐브를 마우스 오른쪽 버튼으로 누르고 **레벨 0 데이터 임포트**를 선택합니다.
4. 큐브의 레벨 0 데이터 임포트 대화상자에서 다음 정보를 입력하고 확인을 누릅니다.
  - **Zip 파일 이름** - 임포트된 데이터의 Zip 파일 이름입니다.
  - **중복 집계 방법** - 동일한 셀의 여러 값을 조합하는 방법을 정의합니다.
    - **중복 값 추가** - 버퍼에 동일한 셀의 여러 값이 포함되어 있는 경우 값을 추가합니다.

- **동일한 값 가정** - 동일한 셀의 여러 값이 동일한지 확인합니다. 동일한 경우 중복 값을 무시합니다. 동일한 셀의 값이 서로 다른 경우 데이터 로드가 중지되고 오류 메시지가 표시됩니다.
  - **마지막 값 사용** - 로드 버퍼에 마지막으로 로드된 셀의 값을 사용하여 중복 셀을 조합합니다. 이 옵션은 최대 10,000개 셀의, 상대적으로 작은 규모의 데이터 로드에서 사용하기 위한 것입니다. 이 옵션을 사용하는 경우 중복 값이 없더라도 데이터가 로드 속도가 크게 저하됩니다.
  - **옵션** - 원하는 경우 다음 옵션을 선택할 수 있습니다.
    - **누락된 값 무시** - 들어오는 데이터 스트림에서 #MISSING 값을 무시합니다.
    - **0값 무시** - 들어오는 데이터 스트림에서 0을 무시합니다.
5. **레벨 0 데이터 импорт 상태** 대화상자에서 **세부정보 표시**를 눌러 импорт 세부정보를 표시하고 **확인**을 누릅니다.


데이터를 импорт하고 나면 Oracle Hyperion Planning 양식을 사용하여 데이터를 보거나 큐브 통계를 확인하여 값이 변경된 방식을 볼 수 있습니다.

 **주:**

데이터가 импорт되는 위치가 아닌 소스에서 импорт된 데이터에 Planning 텍스트 값, 스마트 목록 값 또는 통화 값이 포함되어 있는 경우 데이터가 손상될 수 있습니다.

## ASO 큐브에서 레벨 0 데이터 익스포트

ASO 큐브에서 레벨 0 데이터를 익스포트하려면 다음을 수행합니다.


1. 시스템 뷰에서  를 누릅니다.
2. 데이터베이스 등록정보 탭의 **엔터프라이즈 뷰**에서 집계 저장영역 애플리케이션을 확장합니다.
3. ASO 큐브를 마우스 오른쪽 버튼으로 누르고 **레벨 0 데이터 익스포트**를 선택합니다.
4. 큐브의 **레벨 0 데이터 익스포트** 대화상자에서 익스포트된 데이터의 zip 파일 이름을 입력하고 **확인**을 누릅니다.

익스포트된 레벨 0 데이터가 포함된 .zip 파일을 Oracle Hyperion Planning에서 보려면 다음을 수행합니다.

1. Planning 홈 페이지에서 **애플리케이션**, **개요** 순으로 누릅니다.
2. **애플리케이션** 페이지에서 **작업**, **인박스/아웃박스 탐색기** 순으로 누릅니다.
3. **인박스/아웃박스 탐색기**에서 .zip 파일 옆의 [작업] 아이콘을 눌러 **파일 다운로드**를 선택합니다.

## BSO 큐브에서 레벨 0 데이터 импорт


BSO 큐브에서 레벨 0 데이터를 импорт하려면 다음을 수행합니다.

1. 시스템 뷰에서  를 누릅니다.
2. 데이터베이스 등록정보 탭의 **엔터프라이즈 뷰**에서 블록 저장영역 애플리케이션을 확장합니다.
3. BSO 큐브를 마우스 오른쪽 버튼으로 누르고 **레벨 0 데이터 импорт**를 선택합니다.

4. 큐브의 **레벨 0 데이터 임포트** 대화상자에서 임포트된 데이터의 zip 파일 이름을 입력하고 **확인**을 누릅니다.

## BSO 큐브에서 레벨 0 데이터 익스포트

BSO 큐브에서 레벨 0 데이터를 익스포트하려면 다음을 수행합니다.

1. 시스템 뷰에서 를 누릅니다.
2. 데이터베이스 등록정보 탭의 **엔터프라이즈 뷰**에서 집계 저장영역 애플리케이션을 확장합니다.
3. BSO 큐브를 마우스 오른쪽 버튼으로 누르고 **레벨 0 데이터 익스포트**를 선택합니다.
4. 큐브의 **레벨 0 데이터 익스포트** 대화상자에서 익스포트된 데이터의 zip 파일 이름을 입력하고 **확인**을 누릅니다.

익스포트된 레벨 0 데이터가 포함된 .zip 파일을 Oracle Hyperion Planning에서 보려면 다음을 수행합니다.

1. Planning 홈 페이지에서 **애플리케이션**, **개요** 순으로 누릅니다.
2. **애플리케이션** 페이지에서 **작업**, **인박스/아웃박스 탐색기** 순으로 누릅니다.
3. **인박스/아웃박스 탐색기**에서 .zip 파일 옆의 [작업] 아이콘을 눌러 **파일 다운로드**를 선택합니다.

## 증분 데이터 슬라이스 병합


모든 증분 데이터 슬라이스를 주 데이터베이스 슬라이스로 병합하거나, 주 데이터베이스 슬라이스가 변경되지 않는 동안 모든 증분 데이터 슬라이스를 단일 데이터 슬라이스로 병합할 수 있습니다.

슬라이스를 병합하려면 데이터 로드와 필요한 것과 같은 권한이 있어야 합니다(관리자 또는 데이터베이스 관리자 권한).

### 주:

집계 저장영역 데이터베이스에 대한 증분 데이터 슬라이스만 병합할 수 있습니다.

증분 데이터 슬라이스를 병합하려면 다음을 수행합니다.

1. 시스템 뷰에서 을 누릅니다.
2. **엔터프라이즈 뷰**에서 데이터를 병합할 데이터베이스가 포함된 계획 유형을 마우스 오른쪽 버튼으로 누르고 **데이터 병합**을 선택하고 다음 옵션 중 하나를 선택합니다.
  - **모두** - 모든 데이터 슬라이스를 하나로 병합합니다.
    - 값이 0인 셀 유지(기본값)
    - 값이 0인 셀 제거
  - **증분** - 증분 데이터 슬라이스를 하나로 병합하고 다음 작업 중 하나를 수행합니다.
    - 값이 0인 셀 유지(기본값)



- 값이 0인 셀 제거

3. 데이터 병합 작업 확인 대화상자에서 확인을 눌러 데이터를 병합할지를 확인합니다.

## 데이터 집계


데이터가 포함되고 계산 권한을 부여받은 집계 저장영역 데이터베이스에 대한 집계를 계산합니다.

집계를 수행하려면 시스템 권장 뷰를 사용합니다. 선택한 뷰와 집계 프로세스가 서버에서 수행되는 하나의 구성 불가능한 작업으로 결합됩니다. 선택적으로 결과 파일에 대해 최대 디스크 공간을 지정하고 사용자 쿼리 패턴에 따라 뷰를 선택하도록 하고 뷰 선택에 롤업 계층 구조를 포함할 수 있습니다.

### 주:

집계 저장영역 데이터베이스에 대한 데이터만 집계할 수 있습니다. 데이터 집계 시 모범 사례의 예는 [집계 프로세스 실행](#)을 참조하십시오.

집계를 수행하려면 다음을 수행합니다.

1. 시스템 뷰에서  을 누릅니다.
2. 엔터프라이즈 뷰에서 아웃라인을 압축할 데이터베이스가 포함된 계획 유형을 마우스 오른쪽 버튼으로 누르고 집계 실행을 선택합니다.
3. 집계 실행 작업 - 권장 뷰 사용 대화상자에서 다음 옵션 중 하나를 선택합니다.
  - 쿼리 데이터를 기반으로 합니까? - 수집된 사용자 쿼리 패턴에 따라 서버에서 선택된 뷰를 집계합니다. 이 옵션은 쿼리 추적이 설정된 경우에만 사용할 수 있습니다.
  - 롤업 옵션을 포함합니까? - 뷰 선택 프로세스에 보조 계층 구조(기본 레벨 사용 포함)를 포함합니다.
  - 증가 크기 옵션을 포함합니까? - 집계된 데이터베이스의 최대 증가가 지정한 제한을 초과할 때까지 서버에서 선택된 뷰를 집계합니다. 서버에서 집계를 중지해야 하는 제한 크기(MB)를 입력합니다.
4. 확인을 누릅니다.
 

기존 집계 데이터가 있으면 기본 집계를 삭제하고 집계 프로세스를 다시 실행할지 묻는 메시지가 표시됩니다. 기존 데이터가 있으면 집계 프로세스가 다시 실행되기 전에 삭제됩니다.
5. 확인을 눌러 기존 집계 데이터를 삭제하고 집계를 다시 실행합니다.

## 집계 프로세스 실행

집계 프로세스를 실행하면 검색 성능이 향상됩니다.

ASO 큐브는 계산 스크립트를 사용하여 데이터를 집계하지 않습니다. ASO는 상위 레벨 멤버를 동적으로 계산하려고 합니다. 이렇게 하면 벤티를 더 빠르게 처리할 수 있지만 검색 시간이 더 오래 걸릴 수 있습니다. 이러한 상황을 개선하기 위해 쿼리 추적을 설정하여 양식 작업 및 임시 보고서 실행 같은 작업을 위해 ASO 큐브에 대한 쿼리를 캡처할 수 있습니다. 이러한 쿼리는 집계 프로세스에 사용되어 Oracle Essbase가 쿼리 추적에서 선택된 쿼리 패턴을 사용하여 집계 뷰를 빌드하도록 합니다. 집계 뷰가 생성되면 검색 성능 향상을 확인할 수 있습니다.


집계 프로세스를 실행하려면 먼저 다음을 수행합니다.

- 증분 데이터 슬라이스 병합 및 0 값 셀 제거
- 쿼리 추적 사용
- 쿼리 생성 작업 수행
- 쿼리 추적을 사용하여 집계 실행

## 증분 데이터 슬라이스 병합 및 0 값 셀 제거

논리적 지우기 및 데이터 로드와 같은 일부 Oracle Essbase 작업은 값이 0인 셀이 포함된 증분 데이터 슬라이스를 생성할 수 있습니다. Essbase에서는 증분 데이터 슬라이스를 사용한 집계 허용되지 않습니다. 따라서 병합 작업을 수행하여 증분 데이터 슬라이스를 병합해야 할 수도 있습니다.

병합 작업을 수행하여 값이 0인 셀을 제거하려면 다음을 수행합니다.

1. 시스템 뷰에서  를 누른 후 등록정보를 로드할 집계 저장영역 데이터베이스를 선택합니다.
2. 엔터프라이즈 뷰에서 큐브를 마우스 오른쪽 버튼으로 누르고 **데이터 병합, 모두, 값이 0인 셀 제거** 순으로 선택합니다.
3. 확인을 눌러 데이터 병합 작업을 확인합니다.

이 작업은 모든 증분 슬라이스를 주 데이터베이스 슬라이스로 병합하고 값이 0인 셀을 제거합니다. 영역에서 논리적으로 데이터를 지우면 값이 0인 셀이 발생합니다. 따라서 데이터베이스 크기가 크게 감소됩니다.

병합이 필요하지 않으면 다음과 같은 메시지가 표시됩니다. "There is no incremental data or it is already merged. The specified merge is not necessary." 전체 메시지를 보려면 **병합 작업 상태** 대화상자에서 **세부정보 표시**를 누릅니다.

## 쿼리 추적 사용

ASO 데이터베이스에 대해 쿼리 추적을 사용하여 의미 있는 쿼리 세트를 기록하고 기록된 쿼리 데이터를 사용하여 해당 데이터베이스에 대해 구체화할 가장 적절한 집계 뷰 세트를 선택할 수 있습니다.

Oracle Hyperion Planning에서 수행되는 큐브 새로고침 작업에서 아웃라인 재구조화 작업을 수행합니다. 재구조화 작업의 일부로 Oracle Essbase는 추적된 쿼리를 제거합니다. 병합 작업에서도 추적된 쿼리를 제거합니다.

쿼리 추적을 사용하려면 다음을 수행합니다.

1. 엔터프라이즈 뷰에서 **계획**, ASO 애플리케이션 순으로 확장하고 ASO 큐브를 마우스 오른쪽 버튼으로 누른 후 **쿼리 추적 설정**을 선택합니다.
2. 확인을 눌러 쿼리 추적 작업을 확인합니다.
3. 데이터베이스에 대한 쿼리 추적이 사용되었음을 알려주는 **정보** 대화상자에서 **확인**을 누릅니다.

쿼리 추적을 사용하도록 설정한 후에는 위 단계를 반복하여 사용하지 않도록 설정할 수 있습니다. 쿼리 추적을 사용하지 않는 경우 **정보** 대화상자에 쿼리 추적이 사용됨을 알려주고 쿼리 추적을 사용하지 않을 것인지, 애플리케이션을 중지할 것인지, 집계 프로세스를 실행할 것인지 묻는 메시지가 표시됩니다.

## 쿼리 생성 작업 수행

메모리에만 저장되는 쿼리 추적은 Oracle Hyperion Planning의 ASO 큐브를 사용하는 양식 열기, 임시 보고서 실행 등의 작업에 대한 쿼리를 추적합니다.

쿼리 추적을 사용하도록 설정한 후에도 양식 로드, 비즈니스 규칙 실행 및 보고서 실행을 계속 수행할 수 있습니다. Oracle Essbase는 계속 쿼리를 추적하고 통계를 저장합니다. 이러한 추적된 통계는 집계를 수행하는 동안 사용할 수 있습니다.

## 쿼리 추적을 사용하여 집계 실행

충분한 양의 쿼리를 실행한 경우 쿼리 추적 정보를 사용하여 집계 프로세스를 실행할 수 있습니다.

쿼리 추적 정보를 사용하여 집계 프로세스를 실행하려면 다음을 수행합니다.

1. **엔터프라이즈 뷰에서 계획**, ASO 애플리케이션 순으로 확장하고 ASO 큐브를 마우스 오른쪽 버튼으로 누른 후 **집계 실행**을 선택합니다.
2. **집계 실행 작업** 대화상자에서 **쿼리 데이터 기반**을 선택하고 **확인**을 누릅니다.

이 작업을 완료하는 데 약간의 시간이 걸릴 수 있습니다.

Oracle Hyperion Calculation Manager는 집계 실행 프로세스가 실패하도록 할 수 있는 다음 조건을 확인합니다.

- 쿼리 추적을 사용하도록 설정되지 않았거나 쿼리 추적 데이터가 없는 경우 경고 메시지가 표시됩니다. 쿼리 추적 옵션은 사용할 수 없으나 집계는 계속 처리할 수 있습니다. 쿼리 추적 정보를 사용하려면 쿼리 추적을 사용하도록 설정하거나 쿼리 추적 정보를 생성하는 작업을 수행해야 합니다.
- Oracle Essbase는 여러 데이터베이스 슬라이스가 존재하는 경우 ASO 큐브에서 집계 뷰를 생성하도록 허용하지 않습니다. 경고 메시지가 표시되며 슬라이스가 병합될 때까지 집계를 계속 처리할 수 없습니다.  
이 경우 **엔터프라이즈 뷰**에서 큐브 노드를 마우스 오른쪽 버튼으로 누르고 **데이터 병합, 모두, 값이 0인 셀 제거** 순으로 선택합니다. 병합은 추적된 쿼리 정보를 모두 지웁니다. 따라서 쿼리 추적 정보를 생성하는 작업을 수행해야 합니다.
- 집계가 이미 있는 경우 경고 메시지가 표시됩니다.  
이 경우 **확인**을 누르고 정보 대화상자에서 다음 중 하나를 수행합니다.
  - **확인**을 눌러 기존 집계를 삭제하고 집계 프로세스를 다시 실행합니다.
  - **취소**를 누르고 기존 집계를 지우고 쿼리 추적 정보를 생성하는 작업을 수행합니다.
 집계를 지우려면 다음을 수행합니다.
  1. ASO 큐브를 마우스 오른쪽 버튼으로 누르고 **지우기, 모든 집계** 순으로 선택합니다.
  2. **데이터베이스 집계 지우기 확인** 대화상자에서 **확인**을 누릅니다.


## 요청 관리

세션 창의 정보를 사용하여 활성 요청을 관리합니다.

[세션] 창에는 서버, 애플리케이션 또는 데이터베이스에 대한 활성 사용자 세션 및 요청이 나열됩니다. 사용자는 언제든지 세션을 두 개 이상 열 수 있습니다. 예를 들어 한 사용자에게 두 데이터베이스에 대한 열린 세션이 있을 수 있습니다.

관리자 또는 애플리케이션 관리자 권한이 있으면 사용자 세션을 연결 해제하거나 세션 중에 생성된 특정 요청을 종료할 수 있습니다.

요청을 종료하려면 다음을 수행합니다.

1. 시스템 뷰에서  을 누릅니다.
2. 엔터프라이즈 뷰에서 애플리케이션을 마우스 오른쪽 버튼으로 누르고 세션을 선택합니다.  
세션 창에 활성 세션 및 요청 목록이 표시됩니다. 관리자 권한이 있으면 이 창에는 서버의 모든 사용자에 대한 활성 사용자 세션이 나열됩니다. 애플리케이션 관리자 권한이 있으면 이 창에는 애플리케이션 관리자 권한을 가진 모든 애플리케이션에 연결된 자신을 포함한 모든 사용자에 대한 활성 세션이 나열됩니다.
3. 사용자를 로그오프하려면 옵션 아래 태스크 드롭다운에서 로그오프를 선택한 후 다음 태스크 중 하나를 수행합니다.
  - 엔티티에서 선택한 사용자를 선택하고 로그오프할 사용자를 선택합니다. 그런 다음 적용을 눌러 사용자를 로그오프합니다.
  - 엔티티에서 모든 사용자를 선택하고 다음 태스크 중 하나를 수행합니다.
    - 소스에서 선택한 서버에서 선택하여 선택한 서버에서 모든 사용자를 로그오프합니다. 그런 다음 적용을 누릅니다.
    - 소스에서 선택한 애플리케이션에서 선택하여 선택한 애플리케이션에서 모든 사용자를 로그오프합니다. 그런 다음 적용을 누릅니다.
    - 소스에서 선택한 데이터베이스에서 선택하여 선택한 데이터베이스에서 모든 사용자를 로그오프합니다. 그런 다음 적용을 누릅니다.
  - 엔티티에서 모든 사용자 인스턴스를 선택하고 다음 태스크 중 하나를 수행합니다.
    - 소스에서 선택한 서버에서 선택하고 로그오프할 사용자 인스턴스를 선택합니다. 적용을 눌러 서버에서 사용자의 모든 인스턴스를 로그오프합니다.
    - 소스에서 선택한 애플리케이션에서 선택하고 로그오프할 사용자 인스턴스를 선택합니다. 적용을 눌러 애플리케이션에서 사용자의 모든 인스턴스를 로그오프합니다.
    - 소스에서 선택한 데이터베이스에서 선택하고 로그오프할 사용자 인스턴스를 선택합니다. 적용을 눌러 데이터베이스에서 사용자의 모든 인스턴스를 로그오프합니다.
4. 요청을 종료하려면 옵션 아래 태스크 드롭다운에서 중지를 선택한 후 다음 태스크 중 하나를 수행합니다.
  - 엔티티에서 선택한 요청을 선택하고 세션 목록에서 종료할 요청을 선택합니다. 그리고 나서 적용을 눌러 선택한 요청을 종료합니다.
  - 엔티티에서 모든 요청을 선택하고 다음 태스크 중 하나를 수행합니다.
    - 소스에서 선택한 서버에서 선택하여 선택한 서버에서 모든 사용자의 모든 요청을 종료합니다. 그런 다음 적용을 누릅니다.
    - 소스에서 선택한 애플리케이션에서 선택하여 선택한 애플리케이션에서 모든 사용자의 모든 요청을 종료합니다. 그런 다음 적용을 누릅니다.
    - 소스에서 선택한 데이터베이스에서 선택하여 선택한 데이터베이스에서 모든 사용자의 모든 요청을 종료합니다. 그런 다음 적용을 누릅니다.
  - 엔티티에서 사용자의 모든 요청을 선택하고 다음 태스크 중 하나를 수행합니다.

- 소스에서 **선택한 서버에서**를 선택하고 세션 목록에서 사용자 요청을 선택합니다. 적용을 눌러 서버에서 이 사용자의 모든 요청을 종료합니다.
  - 소스에서 **선택한 애플리케이션에서**를 선택하고 세션 목록에서 사용자 요청을 선택합니다. 적용을 눌러 애플리케이션에서 이 사용자의 모든 요청을 종료합니다.
  - 소스에서 **선택한 데이터베이스에서**를 선택하고 세션 목록에서 사용자 요청을 선택합니다. 적용을 눌러 데이터베이스에서 이 사용자의 모든 요청을 종료합니다.
5. 열을 표시하거나 숨기려면 **뷰**, **열 순**으로 선택합니다. 다음 중 하나를 수행합니다.
    - **모두 표시**를 선택하여 모든 열을 표시합니다. 기본적으로 **연결 소스** 열을 제외한 모든 열이 표시됩니다.
    - **열 관리**를 선택하여 **숨겨진 열** 목록과 **표시되는 열** 목록 간에 열을 이동합니다.
  6. 열 순서를 조정하려면 **뷰**, **열 순서 재지정** 순으로 선택합니다. **열 순서 재지정** 대화상자에서 위쪽 또는 아래쪽 화살표 키를 사용하여 원하는 대로 열 순서를 조정합니다.
  7. 세션 목록을 열별로 정렬하려면 다음을 수행합니다.
    - 열을 오름차순으로 정렬하려면 열 머리글을 누르거나 열 머리글 옆에 있는 **위쪽** 화살표를 누릅니다.  
예를 들어 **사용자** 열을 알파벳순으로 정렬하려면 열 머리글을 누릅니다.
    - 열을 내림차순으로 정렬하려면 **Shift**를 누르고 열 머리글을 누르거나, 열 머리글 옆에 있는 **아래쪽** 화살표를 누릅니다.  
예를 들어 가장 긴 로그인 시간이 목록에 첫 번째로 나타나도록 **로그인 시간** 열을 정렬하려면 **Shift** 키를 누르고 열 머리글을 누릅니다.
  8. 세션 목록을 새로고치려면 **새로고침**을 누릅니다. 예를 들어 처리 중으로 표시되는 세션을 종료하고 **새로고침**을 누르면 세션 목록에는 해당 세션이 더 이상 처리 중이 아닌 것으로 표시됩니다.
  9. 세션 데이터를 Microsoft Excel 스프레드시트로 내보내려면 **익스포트** 버튼을 누르고 파일을 열거나 저장하고 나서 **확인**을 누릅니다.

## Planning 드릴스루 정의 추가

Oracle Hyperion Calculation Manager에서는 Oracle Hyperion Planning 계획 유형에 대한 셀 드릴스루 정의를 나열하고, 추가하고, 편집하고, 삭제할 수 있습니다.

Planning 계획 유형 및 Oracle Hyperion Financial Data Quality Management, Enterprise Edition과 같은 소스에서 데이터를 로드할 멤버가 포함된 Planning 양식으로 작업하는 경우 드릴스루를 통해 해당 셀 데이터 소스에 대한 자세한 내용을 볼 수 있습니다.


### 주:

Planning 양식을 드릴스루할 수 있도록 설정하는 방법에 대한 자세한 내용은 *Oracle Hyperion Planning 관리자 가이드*에서 "드릴스루 정보를 위한 양식 디자인"을 참조하십시오.

Planning 드릴스루 정의를 추가하려면 다음을 수행합니다.

1. **시스템** 뷰에서  를 누릅니다.

2. 엔터프라이즈 뷰에서 애플리케이션을 마우스 오른쪽 버튼으로 누르고 드릴스루 정의를 선택합니다.

3. 드릴스루 정의 대화상자에서  를 누릅니다.

4. 드릴스루 정의 생성에서 다음 정보를 입력하여 드릴스루 정의를 생성합니다.

- **URL 이름** - 드릴스루 정의를 확인할 이름
- **XML 콘텐츠** - URL 링크를 정의하는 XML

서버 및 포트 정보 없이 URL을 입력합니다. 이 URL에는 TDATASEG 테이블의 매개변수 이름과 열 이름이 \$ 기호로 묶여 포함되어야 합니다. 예를 들어 LEDGER\_ID=\$ATTR1\$&GL\_PERIOD=\$ATTR2\$를 입력합니다.. 이 예에서 값 ATTR1은 LEDGER\_ID 매개변수의 값으로 전달되고 ATTR2는 GL\_PERIOD 매개변수의 값으로 전달됩니다. 매개변수는 & 문자로 구분됩니다.



드릴 URL 형식에 대한 클라이언트와 서버 간의 요청-응답을 지정하려면 다음 중 하나를 입력합니다.

- GET - 양식 데이터를 URL로 인코딩

예를 들어 GET@http://www.oracle.com/를 입력합니다. 메소드를 입력하지 않으면 GET을 요청 응답으로 가정합니다.

- POST - 메시지 본문에 양식 데이터 표시

예를 들어 POST@http://www.oracle.com/를 입력합니다.

XML 콘텐츠를 입력하면  을 눌러 XML 콘텐츠 영역으로 파일을 가져올 수 있으며  을 눌러 XML 콘텐츠를 Microsoft Excel로 내보낼 수 있습니다.

- **레벨 0 플래그** - URL을 해당 영역의 레벨 0 하위 멤버에만 적용할 것인지 지정

예를 들어 드릴 가능 영역 DESCENDANTS("Market"),@CHILDREN(Qtr1)에 대해 레벨 0 플래그를 사용하는 경우 모든 상태의 "Market"에, "Qtr1"의 모든 월 동안, 나머지 차원의 모든 레벨 0 멤버에 대해 URL을 적용할 수 있습니다.

- **영역** - 지정된 URL을 사용하여 드릴스루를 허용해야 하는 데이터베이스 영역을 정의하는 멤버 지정

하나 이상의 차원 멤버에 대한 멤버 지정을 사용하여 드릴 가능 영역을 정의합니다. 보안 필터를 정의하는 데 사용하는 것과 동일한 Oracle Essbase 멤버 세트 계산 언어를 사용하여 멤버 지정을 정의합니다. 예를 들어 @REMOVE(@DESCENDANTS("Eastern Region"), "New York"), @CHILDREN(Qtr1)은 "Qtr1" 기간(월) 동안 "New York"을 제외한 모든 동부 주가 적합한 멤버로 지정됨을 나타냅니다.

영역을 추가하려면 **영역 추가**를 누릅니다.

5. 저장, 확인 순으로 누릅니다.

# A

## 관리자 기능 사용

### Calculation Manager로 전환

이 릴리스에서는 Financial Management 또는 Planning에서 생성된 애플리케이션으로 작업하는 Oracle Hyperion Financial Management 및 Oracle Hyperion Planning 사용자와 Oracle Essbase 블록 저장영역 및 집계 저장영역 애플리케이션 사용자가 Oracle Hyperion Calculation Manager에서 비즈니스 규칙을 생성하고 관리할 수 있습니다.

이 표에서는 Calculation Manager, Oracle Hyperion Business Rules(Planning 및 Essbase 사용자의 경우) 및 Financial Management(Financial Management 사용자의 경우)의 비즈니스 규칙 기능을 비교합니다.

표 A-1 Calculation Manager, Financial Management 및 Business Rules의 비즈니스 규칙 기능 비교

기능/작업	Calculation Manager의 경우	Business Rules(Planning 및 Essbase 사용자의 경우)의 경우	Financial Management의 경우
비즈니스 규칙 생성	<p>규칙 디자이너의 순서도에 그래픽 형식으로 비즈니스 규칙을 생성할 수 있습니다.</p> <p>Planning 및 Essbase 블록 저장영역 애플리케이션의 구성요소 5개와 Financial Management 애플리케이션의 구성요소 6개를 사용하여 비즈니스 규칙을 디자인할 수 있습니다.</p> <ol style="list-style-type: none"> <li>1. 공식 구성요소: 멤버, 함수 및 조건문(선택적)으로 작성되거나 디자인된 계산 명령문이 포함됩니다.</li> <li>2. 스크립트 구성요소: Visual Basic(Financial Management) 또는 Essbase(Planning) 계산 스크립트 명령문만 포함됩니다.</li> <li>3. 조건 구성요소: true 또는 false인 조건문(If...Then 문)이 포함됩니다.</li> <li>4. 루프 구성요소: 메타데이터 멤버 목록(예: 계정 목록)이 포함됩니다.</li> <li>5. 멤버 범위 구성요소: 메타데이터 멤버 목록(예: 계정 목록)이 포함됩니다.</li> <li>6. (Financial Management 사용자에게만 해당) 데이터 범위 구성요소: 데이터 레코드 목록(예: 계정 값 목록)이 포함됩니다.</li> </ol> <p>Essbase 합산 저장영역 애플리케이션의 경우 3개의 구성요소가 있습니다.</p> <ol style="list-style-type: none"> <li>1. POV 구성요소: 멤버 범위 구성요소처럼 메타데이터 멤버 목록이 포함됩니다.</li> <li>2. 할당 구성요소: 데이터베이스 아웃라인의 단일 레벨에 있는 멤버의 데이터를 아웃라인의 다른</li> </ol>	<p>Oracle Essbase Administration Services 관리 콘솔의 Business Rules 노드 및 Business Rules 그래픽 디자이너에서 비즈니스 규칙을 생성할 수 있습니다.</p> <p>그래픽 비즈니스 규칙을 디자인하는 데 사용할 수 있는 4개의 작업과 8개의 공식이 있습니다.</p> <ol style="list-style-type: none"> <li>1. 데이터 집계 작업</li> <li>2. 데이터 복사 작업</li> <li>3. 데이터 지우기 작업</li> <li>4. 블록 생성 작업</li> <li>5. 비례 비율 공식</li> <li>6. 배포 요소 공식</li> <li>7. 균등 분할 공식</li> <li>8. 증가-감소 공식</li> <li>9. 단위-비율 공식</li> <li>10. 결합 공식</li> <li>11. 사용자정의 공식</li> <li>12. 변수 공식</li> </ol>	<p>규칙 편집기 또는 텍스트 편집기에서 비즈니스 규칙을 생성할 수 있습니다. Financial Management 비즈니스 규칙에서 Visual Basic과 Financial Management 함수 및 멤버를 사용할 수 있습니다.</p>



표 A-1 (계속) Calculation Manager, Financial Management 및 Business Rules의 비즈니스 규칙 기능 비교

기능/작업	Calculation Manager의 경우	Business Rules(Planning 및 Financial Management의 Essbase 사용자의 경우)의 경우
	<p>멤버에 배포하기 위한 계산이 포함됩니다.</p> <p>3. 공식 구성요소: 멤버, 함수 및 변수를 사용하여 디자인하는 계산 명령문이 포함됩니다.</p> <p>구성요소에서 멤버, 변수 및 함수를 사용할 수 있습니다.</p>	

표 A-1 (계속) Calculation Manager, Financial Management 및 Business Rules의 비즈니스 규칙 기능 비교

기능/작업	Calculation Manager의 경우	Business Rules(Planning 및 Essbase 사용자의 경우)의 경우	Financial Management의 경우
시스템 템플리트 (Calculation Manager에서) 및 작업/공식 (Business Rules에서)을 사용하여 비즈니스 규칙 디자인	<p><b>참고:</b> Essbase 집계 저장영역 애플리케이션에는 시스템 템플리트가 지원되지 않습니다. Calculation Manager에는 8개의 Planning 및 Essbase 블록 저장영역 시스템 템플리트가 있습니다.</p> <ol style="list-style-type: none"> <li>1. 합산 템플리트는 데이터를 합산합니다.</li> <li>2. 데이터 복사 템플리트는 데이터를 복사합니다.</li> <li>3. 데이터 지우기 템플리트는 데이터를 삭제합니다.</li> <li>4. 단순 할당 템플리트는 한 위치에서 다른 위치로 데이터를 할당합니다.</li> <li>5. 레벨별 할당 템플리트는 여러 레벨에서 데이터를 할당합니다.</li> <li>6. 금액-비율-단위 템플리트는 두 개의 변수 값을 제공하면 다른 하나의 변수를 계산합니다.</li> <li>7. 데이터 익스포트 템플리트는 데이터를 데이터베이스나 파일로 내보냅니다.</li> <li>8. SET 명령 템플리트를 통해 계산 스크립트의 성능을 최적화하는 스크립트 명령을 입력할 수 있습니다.</li> </ol> <p>Calculation Manager에는 8개의 Financial Management 시스템 템플리트가 있습니다.</p> <ol style="list-style-type: none"> <li>1. 재무 반올림 템플리트는 통계 반올림이 아닌 재무 반올림을 사용하여 데이터를 반올림합니다. 재무 반올림 함수는 4 내림/5 올림 방법을 사용하여 지정된 소수 자릿수로 데이터를 반올림합니다.</li> <li>2. 월에서 일 수 가져오기 템플리트는 지정한 연도 및</li> </ol>	<p>Business Rules에는 Calculation Manager의 몇 가지 시스템 템플리트처럼 작동하는 4개의 작업이 있습니다.</p> <ol style="list-style-type: none"> <li>1. 데이터 합산 작업은 Calculation Manager의 합산 템플리트처럼 작동합니다.</li> <li>2. 데이터 복사 작업은 Calculation Manager의 데이터 복사 템플리트처럼 작동합니다.</li> <li>3. 데이터 지우기 작업은 블록 생성 작업과 함께 Calculation Manager의 데이터 지우기 템플리트처럼 작동합니다.</li> <li>4. 블록 생성 작업은 데이터 지우기 작업과 함께 Calculation Manager의 데이터 지우기 템플리트처럼 작동합니다.</li> </ol> <p>Business Rules는 Calculation Manager 시스템 템플리트처럼 작동하는 공식도 제공합니다.</p> <ol style="list-style-type: none"> <li>1. 비례 비율 공식은 Calculation Manager의 단순 할당 템플리트처럼 작동합니다.</li> <li>2. 3개의 단위-비율 공식은 Calculation Manager의 금액-비율-단위 템플리트처럼 작동합니다.</li> </ol>	해당 없음

표 A-1 (계속) Calculation Manager, Financial Management 및 Business Rules의 비즈니스 규칙 기능 비교

기능/작업	Calculation Manager의 경우	Business Rules(Planning 및 Financial Management의 Essbase 사용자의 경우)의 경우
	<p>월을 기준으로 하여 한 달 내의 일 수를 생성합니다. 월은 일반적인 범위인 1~12를 벗어나는 숫자로 입력할 수 있으며 이 경우 함수가 오프셋됩니다. 예를 들어 연도 2008과 기간 번호 0을 입력하면 2007년 12월의 일 수가 반환됩니다. 연도 2008과 기간 번호 14를 입력하면 2009년 2월의 일 수가 반환됩니다. 이 함수는 윤년도도 고려합니다.</p>	
	<p>3. 목록에 있음 템플리트는 지정한 차원 멤버가 지정된 멤버 목록의 멤버인지 여부를 테스트합니다.</p>	
	<p>4. 평균 대차대조표 템플리트는 MTD, QTD, HYTD 및 YTD 빈도에 대한 평균 대차대조표 비율을 계산합니다. 데이터를 MTD &lt; QTD, HYTD &lt; YTD 또는 매일 누적 잔액으로 입력할 수 있습니다.</p>	
	<p>5. 기초 잔액 템플리트는 지정된 검색 방법에 따라 계정의 기초 잔액을 계산합니다. 같은 값의 통화나 엔티티 통화 합계에서 기초 잔액을 검색할 수 있습니다.</p>	
	<p>6. 복합 통합 템플리트는 각 엔티티에 지정된 통합 메소드(Holding, Global, Proportional 또는 Equity)에 따라 통합 그룹에 있는 각 엔티티에 대한 통합 및 제거를 수행하는 사전 작성 통합 규칙입니다. 이 템플리트는 자본, 투자, 순이익 및 표준 제거에 대한 계산을 수행합니다. 감사 트랜잭션은 사용자가 설정한 감사 플래그를 기준으로</p>	

표 A-1 (계속) Calculation Manager, Financial Management 및 Business Rules의 비즈니스 규칙 기능 비교

기능/작업	Calculation Manager의 경우	Business Rules(Planning 및 Essbase 사용자의 경우)의 경우	Financial Management의 경우
	<p>생성됩니다. 이 템플리트의 통합 규칙은 일반적인 법적 요구사항을 준수하기 위한 대부분의 계산을 제공합니다.</p> <p>7. 표준 통합 템플리트는 기본 통합 프로세스를 사용하여 통합 그룹의 각 엔티티에 대한 통합 및 제거를 수행하는 사전 작성 통합 규칙입니다.</p> <p>8. 엔티티 할당 템플리트는 그룹 상위 엔티티의 소스 계정을 지정된 할당 가중치에 따라 목록의 각 엔티티에 대한 대상 계정에 할당합니다.</p>		
사용자정의 템플리트 (Calculation Manager에서) 및 매크로 (Business Rules에서)를 생성하여 비즈니스 규칙 디자인	마법사를 통해 재사용 가능한 사용자정의 템플리트를 디자인하여 비즈니스에 대한 고유한 계산을 수행할 수 있습니다. 시스템 템플리트를 복사하고 새 이름으로 저장하여 사용자정의 템플리트의 시작점으로 사용할 수 있습니다.	재사용 가능한 매크로를 디자인하여 비즈니스에 대한 고유한 계산을 수행할 수 있습니다.	해당 없음
디자인 시간 프롬프트 생성	마법사를 사용하여 Calculation Manager의 사용자정의 템플리트에 대한 디자인 시간 프롬프트를 생성할 수 있습니다.	매크로의 변수를 사용하여 정보 확인 프롬프트를 표시할 수 있습니다.	해당 없음
구성요소 공유	계획(Planning의 경우), 계산(Financial Management의 경우) 및 데이터베이스(Essbase의 경우) 유형 및 애플리케이션에 걸쳐 스크립트 및 공식 구성요소를 공유할 수 있습니다.	해당 없음	해당 없음
구성요소 끌어서 놓기	규칙 디자이너 내의 비즈니스 규칙의 순서도에 구성요소를 끌어서 놓을 수 있습니다.	그래픽 디자이너에서 작업과 공식을 비즈니스 규칙의 프로세스 표시줄로 끌어서 놓을 수 있습니다.	해당 없음
세부정보 표시 및 숨기기	순서도 내에서 확대 또는 축소하여 세부정보를 더 많이 (또는 더 적게) 표시할 수 있습니다.	해당 없음	해당 없음

표 A-1 (계속) Calculation Manager, Financial Management 및 Business Rules의 비즈니스 규칙 기능 비교

기능/작업	Calculation Manager의 경우	Business Rules(Planning 및 Essbase 사용자의 경우)의 경우	Financial Management의 경우
계산 스크립트 형식으로 비즈니스 규칙 편집	비즈니스 규칙을 계산 스크립트 형식으로 편집하고 규칙 디자이너에서 그래픽 형식 편집으로 돌아갈 수 있습니다.	비즈니스 규칙을 계산 스크립트 형식으로 편집할 수 있지만 그래픽 형식 편집으로 돌아갈 수는 없습니다.	텍스트 모드에서만 비즈니스 규칙을 편집할 수 있습니다.
비즈니스 규칙 검증	Financial Management 또는 Planning 애플리케이션 관리에서 생성된 애플리케이션으로 작업하는 경우: Financial Management 또는 Planning 각각에 대해서만 비즈니스 규칙을 검증할 수 있습니다. <b>참고:</b> Essbase 비즈니스 규칙은 Essbase에 대해서만 검증할 수 있습니다.	Essbase 또는 Planning에 대해 비즈니스 규칙을 검증할 수 있습니다.	스캔 함수를 통해 Visual Basic 스크립트 파일을 스캔하여 함수가 정확한 매개변수 수를 포함하여 적합한지와 차원 멤버가 애플리케이션에 적합한지 확인할 수 있습니다.
비즈니스 규칙 배포	계산 유형별로 하나의 비즈니스 규칙 세트를 Financial Management에 배포할 수 있습니다. 하나 이상의 비즈니스 규칙과 비즈니스 규칙 세트를 Planning 및 Essbase에 배포할 수 있습니다. Financial Management, Planning 또는 Administration Services(Essbase의 경우) 내에서 실행하려면 비즈니스 규칙과 비즈니스 규칙 세트를 배포해야 합니다. 배포하는 애플리케이션과 계산 유형, 계획 유형 또는 데이터베이스에 따라 실행 위치가 결정됩니다.	해당 없음	해당 없음
비즈니스 규칙 실행	Financial Management 비즈니스 규칙 세트와 Planning 비즈니스 규칙을 Calculation Manager에서 배포한 후 Financial Management 및 Planning 내에서 실행합니다. Essbase 비즈니스 규칙을 배포한 후 Calculation Manager 또는 Administration Services 내에서 실행할 수 있습니다. 실행 위치는 비즈니스 규칙과 비즈니스 규칙 세트를 배포한 위치에 따라 결정됩니다. Calculation Manager에서 비즈니스 규칙의 바로가기를 생성하면 여러 개의 실행 위치를 정의할 수 있습니다.	비즈니스 규칙이 실행될 수 있는 위치를 선택하여 하나 또는 모든 위치에서 비즈니스 규칙을 실행할 수 있습니다. Administration Services 관리 콘솔의 규칙 노드, Business Rules 그래픽 디자이너, Business Rules 웹 실행 프로그램, 명령행 프롬프트 또는 Planning 웹에서 비즈니스 규칙을 실행할 수 있습니다.	계산, 환산, 통합 및 할당을 수행할 때 비즈니스 규칙을 실행합니다.

표 A-1 (계속) Calculation Manager, Financial Management 및 Business Rules의 비즈니스 규칙 기능 비교

기능/작업	Calculation Manager의 경우	Business Rules(Planning 및 Essbase 사용자의 경우)의 경우	Financial Management의 경우
뷰 작업	Calculation Manager에는 액세스 권한이 있는 객체의 네 가지 뷰가 포함되어 있습니다.  1. 시스템 뷰 2. 사용자정의 뷰 3. 배포 뷰 4. 목록 뷰	Business Rules에는 뷰가 포함되어 있지 않습니다. 관리 콘솔 내에서 객체 유형별로 비즈니스 규칙과 기타 객체를 플랫폼 목록에 표시합니다.	해당 없음
시스템 뷰 사용	시스템 뷰에는 Financial Management, Planning 및/또는 Essbase 애플리케이션, 해당 계산 유형, 계획 유형 및/또는 데이터베이스와 액세스 권한이 있는 객체의 계층 목록이 표시됩니다. 이 뷰를 사용하면 비즈니스 규칙을 디자인하는 대상 애플리케이션과 계산 유형, 계획 유형 또는 데이터베이스를 볼 수 있습니다. 이 뷰는 Calculation Manager의 기본 뷰입니다.	해당 없음	해당 없음
사용자정의 뷰 사용	사용자정의 뷰에는 생성하는 폴더와 폴더에 끌어서 놓은 객체의 목록이 표시됩니다. 이 뷰를 사용하여 사용자에게 편리한 방식으로 객체를 구성할 수 있습니다.	<sup>1</sup> 해당 없음	해당 없음
배포 뷰 사용	배포 뷰에는 배포되거나 배포되지 않은 규칙 및 규칙 세트, 해당 배포 및 검증 상태의 목록이 애플리케이션 유형 및 애플리케이션별로 표시됩니다.	해당 없음	해당 없음
목록 뷰 사용	목록 뷰에는 [필터] 대화상자에서 선택하는 객체의 목록이 표시됩니다. [필터] 대화상자를 사용하면 애플리케이션, 계산 및 계획 유형과 액세스 권한이 있는 객체의 애플리케이션 유형별로 필터링된 목록을 생성할 수 있습니다.	해당 없음	해당 없음

표 A-1 (계속) Calculation Manager, Financial Management 및 Business Rules의 비즈니스 규칙 기능 비교

기능/작업	Calculation Manager의 경우	Business Rules(Planning 및 Essbase 사용자의 경우)의 경우	Financial Management의 경우
비즈니스 규칙 마이그레이션	Business Rules의 이전 릴리스에서 Calculation Manager의 현재 릴리스로 비즈니스 규칙 및 기타 객체를 마이그레이션할 수 있습니다. Financial Management 및 Essbase 비즈니스 규칙을 마이그레이션하려면 Calculation Manager의 импорт 기능을 사용합니다.	Business Rules의 이전 릴리스에서 Calculation Manager의 현재 릴리스로 또한 Business Rules의 이전 릴리스에서 Business Rules의 현재 릴리스로 비즈니스 규칙을 마이그레이션할 수 있습니다.	Financial Management의 이전 릴리스에서 Calculation Manager의 현재 릴리스로 또한 Financial Management의 이전 릴리스에서 Financial Management의 현재 릴리스로 비즈니스 규칙을 마이그레이션할 수 있습니다.
비즈니스 규칙 импорт	비즈니스 규칙과 템플릿, 구성요소 등의 기타 객체를 Calculation Manager 내의 다른 Financial Management, Planning 또는 Essbase 애플리케이션이나 Business Rules에서 Calculation Manager로 импорт할 수 있습니다. Essbase 계산 스크립트를 Calculation Manager로 가져올 수도 있습니다. 계산 스크립트 파일을 가져오면 해당 파일은 Calculation Manager에서 그래픽 비즈니스 규칙이 됩니다.	XML 파일에서 Business Rules로 비즈니스 규칙을 가져올 수 있습니다.	[규칙 로드] 옵션을 사용하여 적합한 Visual basic 스크립트 파일을 애플리케이션으로 가져올 수 있습니다.
비즈니스 규칙 익스포트	Calculation Manager에서 xml 파일로 하나 이상의 비즈니스 규칙과 기타 객체를 내보낼 수 있습니다.	Business Rules에서 xml 파일로 비즈니스 규칙과 기타 객체를 내보낼 수 있습니다.	[규칙 추출] 옵션을 사용하여 Financial Management에서 외부 Visual Basic 스크립트 파일로 비즈니스 규칙을 내보낼 수 있습니다.
비즈니스 규칙 바로가기 사용	여러 애플리케이션과 계산 또는 계획 유형에 비즈니스 규칙에 대한 바로가기를 생성할 수 있습니다. 비즈니스 규칙을 바로가기와 함께 배포하면 규칙 사본이 바로가기가 생성된 애플리케이션과 계산 또는 계획 유형에 배포됩니다.	해당 없음	해당 없음

표 A-1 (계속) Calculation Manager, Financial Management 및 Business Rules의 비즈니스 규칙 기능 비교

기능/작업	Calculation Manager의 경우	Business Rules(Planning 및 Essbase 사용자의 경우)의 경우	Financial Management의 경우
변수 작업	<p>Calculation Manager에는 Planning에 대한 변수 유형 4개, Financial Management에 대한 변수 유형 2개 및 Essbase에 대한 변수 유형 3개가 있습니다.</p> <ol style="list-style-type: none"> <li>1. 글로벌: Planning 또는 Financial Management 애플리케이션에서 사용할 수 있습니다.</li> <li>2. 애플리케이션: 변수가 생성된 Planning, Financial Management 또는 Essbase 애플리케이션에서만 사용할 수 있습니다.</li> <li>3. 계획 유형 또는 데이터베이스: 변수가 생성된 Planning 계획 유형 또는 Essbase 데이터베이스에서만 사용할 수 있습니다.</li> <li>4. 비즈니스 규칙: 변수가 생성된 Planning 또는 Essbase 비즈니스 규칙에서만 사용할 수 있습니다.</li> </ol>	<p>Business Rules에는 2개의 변수 유형이 있습니다.</p> <ol style="list-style-type: none"> <li>1. 글로벌: 비즈니스 규칙에서 사용할 수 있습니다.</li> <li>2. 로컬: 변수가 생성된 비즈니스 규칙에서만 사용할 수 있습니다.</li> </ol>	<p>Financial Management에는 2개의 변수 유형이 있습니다.</p> <ol style="list-style-type: none"> <li>1. 글로벌: 전체 계산 프로세스에 적용됩니다.</li> <li>2. 로컬: 개별 서브루틴에만 적용됩니다.</li> </ol>
비즈니스 규칙을 생성 및 편집할 액세스 권한 지정	<p>Calculation Manager에서 비즈니스 규칙과 기타 객체를 생성, 표시 및 편집하는 기능은 Oracle Hyperion Shared Services에서 지정된 역할과 Calculation Manager에서의 객체 소유권에 따라 결정됩니다. 기본적으로 사용자는 사용자가 생성한 비즈니스 규칙과 기타 객체를 소유합니다. 객체의 관리자 또는 소유자는 소유권을 다른 사용자에게 지정할 수 있습니다.</p>	<p>비즈니스 규칙에서 비즈니스 규칙과 기타 객체를 수정할 수 있는 사용자와 그룹을 선택하여 해당 항목을 편집할 권한을 지정합니다.</p>	<p>Financial Management에서 비즈니스 규칙을 생성하거나 편집하는 데는 액세스 권한이 필요하지 않습니다.</p>



표 A-1 (계속) Calculation Manager, Financial Management 및 Business Rules의 비즈니스 규칙 기능 비교

기능/작업	Calculation Manager의 경우	Business Rules(Planning 및 Essbase 사용자의 경우)의 경우	Financial Management의 경우
비즈니스 규칙을 실행할 액세스 권한 지정	비즈니스 규칙 및 비즈니스 규칙 세트가 Calculation Manager에서 배포된 후 Financial Management, Planning 또는 Administration Services(Essbase의 경우)에서 비즈니스 규칙 및 비즈니스 규칙 세트를 실행할 수 있는 권한을 지정합니다.	비즈니스 규칙과 시퀀스를 실행할 수 있는 데이터베이스 위치(또는 모든 위치)를 선택하고 이를 실행할 사용자와 그룹을 선택하여 Business Rules에서 비즈니스 규칙과 시퀀스를 실행할 수 있는 권한을 지정합니다.	계산 프로세스가 속한 애플리케이션에 대한 올바른 역할 보안 액세스를 지정하여 Financial Management에서 계산 프로세스를 실행할 수 있는 권한을 지정합니다. 예를 들어 통합 프로세스를 실행하려면 사용자에게 통합에 대한 역할 보안이 있어야 합니다.

1 Business Rules에서 프로젝트를 생성하여 편리한 방식으로 비즈니스 규칙, 시퀀스, 매크로 및 변수를 구성할 수 있습니다.