

# Oracle® Data Relationship Management Suite 관리자 가이드



릴리스 11.2.x  
F28493-03  
2022년 7월

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

ORACLE®

Oracle Data Relationship Management Suite 관리자 가이드NOT\_SUPPORTED릴리스 11.2.x

F28493-03

Copyright © 1999, 2023, Oracle and/or its affiliates.

주요 작성자: EPM Information Development Team

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

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

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

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

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

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

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

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

# 목차

설명서 접근성

---

설명서 피드백

---

1 개정 내역

---

2 Data Relationship Management Suite 정보

---

3 시작

---

Data Relationship Management 애플리케이션 관리	3-1
Data Relationship Management 액세스	3-1
비밀번호 변경	3-2
문제 해결 및 팁	3-2

4 사용자 관리

---

사용자 권한	4-1
사용자 역할	4-7
분석 역할	4-10
사용자 생성	4-10
사용자 인증	4-12
사용자 수정	4-13
비밀번호 변경	4-13
사용자 잠금	4-13
사용자 잠금 해제	4-13
사용자 역할 및 지정 변경	4-14
사용자 삭제	4-14
사용자 로그인 상태 보기	4-14
시스템 정의 사용자	4-15

공통 사용자 프로비저닝	4-15
사전 필수 조건	4-16
사용자 및 그룹 프로비저닝	4-16
Data Relationship Management 사용자 및 그룹 멤버십 동기화	4-16
수동 동기화	4-16
스케줄링된 동기화	4-17
부분 동기화	4-17

## 5 노드 액세스 그룹 관리

---

워크플로우 그룹 유형 노드 액세스 레벨	5-2
노드 액세스 그룹 생성	5-3
노드 액세스 그룹 편집	5-3
노드 액세스 그룹 삭제	5-4
노드 액세스 그룹 보안 지정	5-4

## 6 객체 액세스 그룹 관리

---

객체 액세스 그룹 생성	6-2
객체 액세스 그룹 편집	6-2
객체 액세스 그룹 삭제	6-3

## 7 도메인 관리

---

도메인 생성	7-1
도메인 편집	7-2
도메인 삭제	7-2

## 8 등록정보 범주 관리

---

등록정보 범주	8-1
등록정보 범주 생성	8-2
등록정보 범주 편집	8-3
등록정보 범주 삭제	8-3

## 9 등록정보 정의 관리

---

데이터 유형	9-2
외부 록업	9-4
등록정보 생성	9-5
계층 제약 조건 사용	9-9
등록정보 정의 편집	9-9

등록정보 삭제	9-10
---------	------

## 10 검증 관리

---

검증 클래스	10-1
검증 레벨	10-4
검증 생성	10-6
이동 관련 스크립트 검증 생성	10-7
검증 지정	10-7
검증 편집	10-8
검증 삭제	10-8

## 11 공식 관리

---

함수 작업	11-1
특수 문자	11-1
리터럴	11-2
형식 문자열 매개변수	11-2
날짜-시간 형식 문자열	11-4
공식 평가	11-6
공식 구문 확인	11-6
구문 확인의 등록정보 이름	11-6
공식 사용 시의 고려 사항	11-7
공식 생성	11-8
함수 정의	11-9
함수 그룹	11-58

## 12 동적 스크립트 관리

---

실행 컨텍스트	12-1
스크립트를 사용한 파생 등록정보	12-1
스크립트를 사용한 검증	12-2
스크립트를 사용한 관리 요청	12-3
열거 상수	12-3
지원되는 JavaScript 데이터 유형	12-4
데이터 유형 변환	12-5
숫자 형식 지정	12-7
날짜 형식 지정	12-8
Data Relationship Management 객체	12-10
실행 환경	12-23

동적 스크립트 생성	12-24
------------	-------

## 13 노드 유형 관리

---

노드 유형 정의	13-1
노드 유형 편집	13-1
노드 유형 삭제	13-2
노드 그래픽 기호 작업	13-2

## 14 시스템 환경설정 작업

---

시스템 환경설정	14-1
트랜잭션 내역 로깅 레벨 설정	14-10
변경 승인 설정	14-10
시스템 환경설정 구성	14-11

## 15 외부 연결 작업

---

외부 연결 정의	15-1
외부 연결 편집	15-4
외부 연결 삭제	15-5

## 16 관리 워크플로우 구성

---

워크플로우 태스크 관리	16-1
태스크 등록정보	16-1
태스크 및 등록정보 지침	16-1
태스크 검증	16-2
계산된 이름 및 상위 멤버 등록정보	16-2
외부 커밋	16-3
워크플로우 태스크 생성	16-3
워크플로우 태스크 편집	16-6
워크플로우 태스크 복사	16-6
워크플로우 태스크 삭제	16-7
워크플로우 모델 관리	16-7
워크플로우 단계	16-7
모델 필터	16-12
요청 및 클레임 기간	16-12
워크플로우 모델 생성	16-13
워크플로우 모델 편집	16-14
워크플로우 모델 복사	16-15

워크플로우 모델 이름 바꾸기	16-15
워크플로우 모델 숨기기	16-15
워크플로우 모델 삭제	16-16

## 17 Data Relationship Management Analytics 관리

---

Data Relationship Management Analytics 액세스	17-2
환경설정 작업	17-2
실행 계획 작업	17-3
실행 계획 생성	17-3
실행 계획 편집	17-4
실행 계획 비활성화 및 재활성화	17-4
실행 계획 삭제	17-5
활동 보기	17-5

## 18 외부 워크플로우 애플리케이션 통합

---

외부 요청	18-1
-------	------

## 19 Data Relationship Management 메타데이터 마이그레이션

---

마이그레이션 유틸리티 열기	19-2
메타데이터 추출	19-2
메타데이터 로드	19-4
메타데이터 비교	19-5
메타데이터 보기	19-6
메타데이터 파일 제한 사항	19-6
보고서 생성	19-7

# 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

## 개정 내역

이 가이드 릴리스에서는 다음 항목이 업데이트되었습니다.

항목	변경
데이터 유형	기본값이 정의되지 않은 경우 Float 및 Integer 모두에 대해 0이 익스포트된다는 내용의 참고 사항이 추가되었습니다. 날짜, 날짜/시간 및 시간 데이터 유형은 고정 문화권에서 형식이 지정된다는 내용이 추가되었습니다.
함수 정의	Equals 함수에서 비교 시 대소문자를 구분한다는 내용이 업데이트되었습니다.
동적 스크립트 생성	상위 이름을 계산할 때 특수 문자가 사용된 경우 표준 JavaScript 규칙에 따라 특수 문자를 이스케이프해야 한다는 내용의 참고 사항이 추가되었습니다.
Data Relationship Management 메타데이터 마이그레이션	마이그레이션 로드 및 추출 시 외부 연결에 사용되는 연결 문자열, 사용자 ID 및 비밀번호는 함께 마이그레이션되지 않음을 나타내는 내용의 참고 사항이 추가되었습니다. "핵심 등록정보 구성 및 설정 마이그레이션"이라는 새 섹션이 추가되었습니다.
Data Relationship Management 객체	RequestItemObject의 NodeNamePendingInRequest 메소드에 대한 설명이 업데이트되었습니다.
계산된 이름 및 상위 멤버 등록정보	이름 또는 상위 멤버가 수동으로 재정의되는 경우 동작을 설명하는 내용의 참고 사항이 추가되었습니다.
워크플로우 모델 생성	이름 또는 상위 멤버가 수동으로 재정의되는 경우 동작을 설명하기 위해 6단계의 태스크 등록정보 재계산 글머리 기호에 참고 사항이 추가되었습니다.
동적 스크립트 관리	HierarchyObject 메소드 테이블에서 NodeExists(abbrev)에 대한 설명이 업데이트되었습니다. RequestItemDetailObject에 두 개의 새 등록정보가 추가되었습니다. <ul style="list-style-type: none"><li>• CalcValue</li><li>• HasCalcValue</li></ul>
통지	통지 동작 설명 및 업데이트에 대한 다양한 내용이 업데이트되었습니다.
지원되는 JavaScript 데이터 유형	배열 사용을 설명하는 내용의 참고 사항이 추가되었습니다.
이동 관련 스크립트 검증 생성	이동 관련 스크립트 검증 생성이라는 새 항목이 추가되었습니다.

항목	변경
시스템 환경설정	FindByProperties 시스템 환경설정 설명에 참고 사항이 추가되었습니다. SharedNodeDelimiter 및 SharedNodeSequenceSeparator 시스템 환경설정에 대한 설명이 업데이트되었습니다.
함수 정의	여러 함수의 로컬 사용에 대한 설명이 추가되었습니다.
문제 해결 및 팁	시작 장에 "문제 해결 및 팁"이라는 새 섹션이 추가되었습니다. 필드에 붙여넣기에 대한 해결 방법 정보가 추가되었습니다. 애플리케이션 성능 정보가 추가되었습니다.
검증 클래스	UniqueProp 검증에서 인덱싱된 등록정보를 사용하라는 권장 사항이 추가되었습니다.

# 2

## Data Relationship Management Suite 정보

Oracle Data Relationship Management Suite는 다음으로 구성됩니다.

- Oracle Data Relationship Management
- Oracle Data Relationship Management Read Only Access
- Oracle Data Relationship Steward
- Oracle Data Relationship Governance
- Oracle Data Relationship Management Analytics
- Oracle Data Relationship Management for Oracle Hyperion Enterprise Planning Suite
- Oracle Data Relationship Management for Oracle Hyperion Financial Close Suite

# 3 시작

## 참조:

- [Data Relationship Management 애플리케이션 관리](#)
- [Data Relationship Management 액세스](#)
- [문제 해결 및 팁](#)

## Data Relationship Management 애플리케이션 관리

Oracle Data Relationship Management는 애플리케이션을 사용하여 데이터를 관리하고 사용자의 데이터 액세스 및 변경 요청을 처리합니다. 단일 Data Relationship Management 설치에서 하나 이상의 애플리케이션을 지원할 수 있습니다. 각 애플리케이션은 해당 시스템 메타데이터 및 보안 구성을 사용하여 데이터를 관리하고 액세스합니다. 동일한 애플리케이션에서 공통 데이터 세트 및 제한된 데이터 세트에 대한 다양한 액세스 레벨을 가진 여러 사용자와 여러 데이터 세트를 지원할 수 있습니다. 그러나 애플리케이션 내의 모든 시스템 메타데이터는 동일한 사용자에게 의해 공유 및 관리됩니다. 시스템 메타데이터 변경사항은 즉시 적용되며 모든 사용자와 데이터가 영향을 받을 수 있습니다. 다른 그룹의 메타데이터 변경사항으로부터 다른 사용자 그룹을 격리해야 하는 경우 각 그룹이 별도의 애플리케이션을 사용하는 것이 좋습니다.

애플리케이션은 Data Relationship Management 기본 애플리케이션 서버에서 액세스되는 구성 콘솔에서 생성됩니다. 새 애플리케이션 생성에 대한 자세한 내용은 *Oracle Data Relationship Management 설치 가이드*의 "애플리케이션 생성"을 참조하십시오.

새 Data Relationship Management 애플리케이션에는 기본 관리 사용자와 등록정보 정의 및 범주 같은 핵심 메타데이터 객체가 포함됩니다. 이 초기 구성에서는 기본 사용자가 네 가지 태스크를 수행하여 애플리케이션을 빌드하고 채우고 프로비전닝할 수 있습니다.

- 버전 및 계층 구조 생성
- 쿼리, 비교, импорт, 혼합기, 익스포트 등의 사용자 메타데이터 객체 정의
- 도메인, 등록정보 정의, 검증 및 노드 유형을 포함하여 시스템 메타데이터 객체 설정 및 구성
- 사용자 추가 및 제품 기능, 객체 및 데이터 액세스에 대한 보안 구성

이 가이드에서는 Data Relationship Management 애플리케이션에 대한 사용자 보안 및 시스템 메타데이터와 관련된 관리 태스크를 설명합니다. 버전, 계층 구조 및 사용자 메타데이터 객체 관리에 대한 자세한 내용은 *Oracle Data Relationship Management 사용자 가이드*를 참조하십시오.

## Data Relationship Management 액세스

Oracle Data Relationship Management 클라이언트를 시작하려면 다음을 수행합니다.

1. 시작, 프로그램, Oracle EPM System, Data Relationship Management, 웹 클라이언트 순으로 선택합니다.
2. 사용자 이름과 비밀번호를 입력합니다.  
사용자 이름과 비밀번호는 대소문자를 구분합니다.

3. 애플리케이션을 선택하고 **로그온**을 누릅니다.  
자세한 내용은 [비밀번호 변경](#)을 참조하십시오.

## 비밀번호 변경

비밀번호를 변경하려면 다음을 수행합니다.

1. Oracle Data Relationship Management 홈 페이지에서 **환경설정**을 선택합니다.
2. **내 비밀번호 변경**을 누릅니다.
3. 현재 비밀번호를 입력합니다.
4. 새 비밀번호를 입력합니다.

### 주:

사용자가 기본적으로 인증되고 PasswordPolicyEnabled 시스템 환경설정이 True로 설정된 경우 비밀번호에 다음 요소 중 3개가 포함되어야 합니다.

- 대문자
- 소문자
- 숫자
- 특수 문자

### 주:

그렇지 않고 사용자가 Oracle Hyperion Shared Services를 통해 인증될 때 외부 디렉토리에 의한 경우가 아니면 비밀번호가 제한되지 않습니다.

5. 새 비밀번호를 다시 입력합니다.
6. **확인**을 누릅니다.

## 문제 해결 및 팁

### 항목 필드에 붙여넣기

일부 경우에는 마우스 오른쪽 단추를 누른 다음 **붙여넣기**를 사용하여 콘텐츠를 클립보드에서 붙여넣을 수 없습니다. 이 문제를 해결하려면 Ctrl-V를 사용하거나 **편집**을 누른 다음 **붙여넣기**를 선택하여 콘텐츠를 클립보드에서 붙여넣습니다.

### 애플리케이션 성능

애플리케이션 성능을 유지 관리하기 위해, 표준 프로그래밍 사례를 이용하여 문자열 데이터에 보다 빠른 액세스를 제공하는 문자열 인터닝 기능이 사용되었습니다. 문자열 인터닝에는 각 문자열의 변경 불가능한 복사본이 한 번 저장되며 애플리케이션이 실행되는 동안 이후 액세스를 위해 이 복사본이 유지 관리됩니다. 따라서 데이터에 액세스하고 콘텐츠를 관리하는 동안 애플리케이션이 다시 시작될 때까지 엔진의 메모리 사용 공간이 점진적으로 증가합니다.

# 4

## 사용자 관리

### 참조:

- 사용자 권한
- 사용자 역할
- 사용자 생성
- 사용자 인증
- 사용자 수정
- 사용자 삭제
- 사용자 로그인 상태 보기
- 시스템 정의 사용자
- 공동 사용자 프로비저닝

## 사용자 권한

Oracle Data Relationship Management는 세 가지 권한 레벨을 사용하여 제품 기능과 데이터에 대한 사용자 액세스를 제어합니다. 일부 상위 레벨 권한에는 하위 레벨 권한도 포함됩니다. 사용자에게 상위 레벨 권한이 부여된 경우 모든 하위 레벨 권한도 부여됩니다. 예를 들어 사용자에게 레벨 1 권한이 부여된 경우 아래에 있는 레벨 2 및 3 권한도 모두 부여됩니다.

### 버전 권한

표 4-1 버전 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
버전 관리 – 사용자가 [버전] 및 [계층] 메뉴 옵션에 액세스할 수 있습니다.	버전 찾아보기 – 사용자가 노드 액세스 그룹에서 권한이 부여된 모든 버전에 액세스할 수 있습니다.	해당 없음
	버전 생성 – 사용자가 소유자인 모든 버전을 관리(업데이트/삭제)할 수 있습니다. 사용자가 [버전] 메뉴 옵션에 액세스할 수 있습니다.	해당 없음
	참고: [버전 관리] 권한을 가진 사용자가 소유자를 변경할 때까지 버전을 생성한 사용자가 소유자입니다.	
	계층 관리 – 사용자가 [계층] 메뉴 옵션에 액세스할 수 있습니다.	계층 찾아보기 – 사용자가 노드 액세스 그룹에서 권한이 부여된 모든 계층에 액세스할 수 있습니다. 사용자가 [편집] 노드 액세스 권한 이상이 있는 경우 [노드] 메뉴 옵션에 액세스할 수 있습니다.

표 4-1 (계속) 버전 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
		<p><b>계층 생성</b> - 사용자가 소유자인 모든 계층을 관리(업데이트/삭제)할 수 있습니다. 사용자가 [계층] 메뉴 옵션에 액세스할 수 있습니다. 사용자가 소유자인 모든 계층에 대해 노드 유형을 비활성화할 수 있습니다.</p> <p><b>참고:</b> [계층 관리] 권한을 가진 사용자가 소유자를 변경할 때까지 계층을 생성한 사용자가 소유자입니다.</p>

요청 권한

표 4-2 요청 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
<p><b>요청 관리</b> - 사용자가 시스템에서 아직 커밋되지 않은 모든 요청을 삭제할 수 있습니다.</p>	<p><b>요청 생성</b> - 사용자가 시스템의 모든 요청을 쿼리하고 소유자인 모든 요청을 관리(업데이트/삭제)할 수 있습니다.</p>	해당 없음
<p><b>워크플로우 참가자</b> - 사용자는 관리 워크플로우 모델을 사용하여 요청에 참여할 수 있습니다.</p>	해당 없음	해당 없음

쿼리 권한

표 4-3 쿼리 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
<p><b>시스템 쿼리 관리</b> - 사용자가 시스템 쿼리 및 [쿼리] 메뉴 옵션에 액세스할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>	<p><b>사용자 쿼리 관리</b> - 사용자가 사용자 및 표준 쿼리를 보고 실행할 수 있습니다. 사용자가 표준 쿼리에 대한 [쿼리] 메뉴 옵션에 액세스할 수는 없습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>	<p><b>쿼리 실행</b> - 사용자가 모든 표준 쿼리를 보고 실행할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다. 사용자가 [편집] 노드 액세스 권한 이상이 있는 경우 [노드] 메뉴 옵션에 액세스할 수 있습니다.</p>
	<p><b>표준 쿼리 관리</b> - 사용자가 표준 쿼리에 대한 [쿼리] 메뉴 옵션에 액세스할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>	해당 없음

## 비교 권한

표 4-4 비교 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
<b>시스템 비교 관리</b> – 사용자가 시스템 비교 및 [비교] 메뉴 옵션에 액세스할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.	<b>사용자 비교 관리</b> – 사용자가 사용자 및 표준 비교를 보고 실행할 수 있습니다. 사용자가 표준 비교에 대한 [비교] 메뉴 옵션에 액세스할 수는 없습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.	<b>비교 실행</b> – 사용자가 모든 표준 비교를 보고 실행할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다. 사용자가 [편집] [노드] 메뉴 옵션에 액세스할 수 있습니다.
	<b>표준 비교 관리</b> – 사용자가 표준 비교에 대한 [비교] 메뉴 옵션에 액세스할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.	해당 없음

## 임포트 권한

표 4-5 임포트 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
<b>시스템 임포트 관리</b> – 사용자가 시스템 임포트 및 [임포트] 메뉴 옵션에 액세스할 수 있습니다. 사용자가 등록정보 범주 보안에 따라 등록정보 선택기에 제한적으로 액세스할 수 있습니다.	<b>사용자 임포트 관리</b> – 사용자가 사용자 및 표준 임포트를 보고 실행할 수 있습니다. 사용자가 표준 임포트에 대한 [임포트] 메뉴 옵션에 액세스할 수는 없습니다. 사용자가 등록정보 범주 보안에 따라 등록정보 선택기에 제한적으로 액세스할 수 있습니다.	<b>임포트 실행</b> – 사용자가 모든 표준 임포트를 보고 실행할 수 있습니다. 사용자가 등록정보 범주 보안에 따라 등록정보 선택기에 제한적으로 액세스할 수 있습니다.
	<b>표준 임포트 관리</b> – 사용자가 표준 임포트에 대한 [임포트] 메뉴 옵션에 액세스할 수 있습니다. 사용자가 등록정보 범주 보안에 따라 등록정보 선택기에 제한적으로 액세스할 수 있습니다.	해당 없음

## 혼합기 권한

표 4-6 혼합기 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
<p><b>시스템 혼합기 관리</b> - 사용자가 시스템 혼합기 및 [혼합기] 메뉴 옵션에 액세스할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>	<p><b>사용자 혼합기 관리</b> - 사용자가 사용자 및 표준 혼합기를 보고 실행할 수 있습니다. 사용자가 표준 혼합기에 대한 [혼합기] 메뉴 옵션에 액세스할 수는 없습니다.</p>	<p><b>혼합기 실행</b> - 사용자가 모든 표준 혼합기를 보고 실행할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>
	<p><b>표준 혼합기 관리</b> - 사용자가 표준 혼합기에 대한 [혼합기] 메뉴 옵션에 액세스할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>	<p>해당 없음</p>

## 익스포트 권한

표 4-7 익스포트 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
<p><b>시스템 익스포트 관리</b> - 사용자가 시스템 익스포트 및 [익스포트] 메뉴 옵션에 액세스할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>	<p><b>사용자 익스포트 관리</b> - 사용자가 사용자 및 표준 익스포트와 장부를 보고 실행할 수 있습니다. 사용자가 표준 익스포트 및 장부에 대한 [익스포트] 메뉴 옵션에 액세스할 수는 없습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>	<p><b>익스포트 실행</b> - 사용자가 모든 표준 익스포트를 보고 실행할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>
	<p><b>표준 익스포트 관리</b> - 사용자가 표준 익스포트 및 장부에 대한 [익스포트] 메뉴 옵션에 액세스할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>	<p>해당 없음</p>

## 스크립트 권한

표 4-8 스크립트 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
<p><b>작업 스크립트 실행</b> – 사용자가 작업 스크립트를 실행할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>	해당 없음	해당 없음

## 감사 권한

표 4-9 사용자 트랜잭션 감사 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
<p><b>사용자 트랜잭션 감사</b> – 사용자가 수행한 모든 트랜잭션을 쿼리할 수 있습니다. 트랜잭션에는 데이터 및 메타데이터 변경사항과 로그인, 비동기 작업 실행 등의 로깅된 작업이 포함될 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>	해당 없음	해당 없음

표 4-10 데이터 트랜잭션 감사 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
<p><b>데이터 트랜잭션 감사</b> – 사용자가 권한 또는 노드 액세스 그룹에서 액세스 권한이 있는 데이터 객체에 대한 모든 트랜잭션을 쿼리할 수 있습니다. 트랜잭션에는 사용자가 수행한 트랜잭션과 다른 사용자가 수행한 변경사항이 포함될 수 있습니다. 노드 레벨 트랜잭션의 경우 사용자에게 모든 하위 멤버에 대한 읽기 액세스 권한도 있다고 가정하여 사용자가 노드 및 모든 하위 멤버([하위 노드 포함] 옵션)에 대한 트랜잭션을 쿼리할 수 있습니다. 사용자가 노드 액세스 그룹 지정 및 등록정보 범주 보안에 따라 버전, 계층, 노드 및 등록정보 선택기에 제한적으로 액세스할 수 있습니다.</p>	해당 없음	해당 없음

표 4-11 시스템 트랜잭션 감사 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
시스템 트랜잭션 감사 – 사용자가 수행한 모든 트랜잭션을 쿼리할 수 있습니다. 트랜잭션에는 데이터 및 메타데이터 변경사항과 로그인, 비동기 작업 실행 등의 로깅된 작업이 포함될 수 있습니다.	해당 없음	해당 없음

애플리케이션 권한

표 4-12 애플리케이션 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
애플리케이션 관리	범주 관리	범주 찾아보기 – 사용자가 등록정보 범주 보안에서 권한이 부여된 모든 등록정보 범주에 액세스할 수 있습니다.
	등록정보 관리	등록정보 찾아보기 – 사용자가 등록정보 범주 보안에서 권한이 부여된 등록정보 범주의 모든 등록정보에 액세스할 수 있습니다.
		등록정보 목록 관리 – 사용자가 등록정보 정의의 록업 테이블 및 값 목록을 관리할 수 있습니다.
	검증 관리	해당 없음
	노드 유형 관리	해당 없음
	환경설정 관리	해당 없음

액세스 권한

표 4-13 액세스 권한

권한 레벨 1	권한 레벨 2	권한 레벨 3
액세스 관리	사용자 관리 – 사용자가 해당 사용자 프로파일을 편집하거나 삭제할 수 없습니다.	해당 없음
	역할 관리 – 사용자가 해당 역할 지정을 편집할 수 없습니다.	해당 없음
	액세스 그룹 관리 – 사용자가 해당 노드 액세스 그룹 지정을 편집할 수 없습니다.	해당 없음
	등록정보 액세스 관리 – 사용자가 해당 등록정보 범주 지정을 편집할 수 없습니다.	해당 없음

## 사용자 역할

Oracle Data Relationship Management 권한은 역할을 사용하여 사용자에게 지정됩니다. 각 사용자 역할은 제품 기능 또는 데이터에 대한 액세스를 제공하는 권한 세트와 연결됩니다. 사용자에게 하나 이상의 역할을 지정할 수 있으며, 이 경우 모든 역할의 결합된 권한이 부여됩니다. 충돌하는 액세스 레벨이 있는 두 역할을 사용자에게 지정하면 상위 액세스 레벨이 사용자에게 부여됩니다.

Data Relationship Management는 표시된 권한이 지정된 다음 사용자 역할을 제공합니다.

표 4-14 사용자 역할 - 권한

권한			사용자 역할							
레벨 1	레벨 2	레벨 3	액세스 관리자	익명 사용자	애플리케이션 관리자	데이터 생성자	데이터 관리자	대화식 사용자	워크플로우 사용자	관리 사용자
버전 관리							X			
	버전 찾아보기		X	X	X	X		X	X	X
	버전 생성					X				
	계층 구조 관리						X			
		계층 구조 찾아보기	X	X	X	X		X	X	X
		계층 구조 생성				X				
요청 관리							X			
	요청 생성				X				X	
시스템 쿼리 관리					X					
	사용자 쿼리 관리					X	X	X		
		쿼리 실행		X					X	
	표준 쿼리 관리						X			
시스템 비교 관리					X					

표 4-14 (계속) 사용자 역할 - 권한

권한			사용자 역할							
레벨 1	레벨 2	레벨 3	액세스 관리자	익명 사용자	애플리케이션 관리자	데이터 생성자	데이터 관리자	대화식 사용자	워크플로우 사용자	관리 사용자
	사용자 비교 관리					X	X	X		
		비교 실행		X					X	
	표준 비교 관리						X			
시스템 임포트 관리					X					
	사용자 임포트 관리					X	X			
		임포트 실행								
	표준 임포트 관리						X			
시스템 혼합기 관리					X					
	사용자 혼합기 관리					X	X			
		혼합기 실행								
	표준 혼합기 관리						X			
시스템 익스포트 관리					X					
	사용자 익스포트 관리					X	X	X		
		익스포트 실행		X					X	
	표준 익스포트 관리						X			
작업 스크립트 실행					X	X	X	X		

표 4-14 (계속) 사용자 역할 - 권한

권한			사용자 역할							
레벨 1	레벨 2	레벨 3	액세스 관리자	익명 사용자	애플리케이션 관리자	데이터 생성자	데이터 관리자	대화식 사용자	워크플로우 사용자	관리 사용자
사용자 트랜잭션 감사			X		X	X	X	X	X	
데이터 트랜잭션 감사					X	X	X	X		
시스템 트랜잭션 감사			X		X					
애플리케이션 관리					X					
	범주 관리									
		범주 찾아보기	X	X		X	X	X	X	X
	등록정보 관리									
		등록정보 찾아보기	X	X		X	X	X	X	X
		등록정보 목록 관리					X			
	검증 관리									
	노드 유형 관리				X					
	환경설정 관리									
액세스 관리										
	사용자 관리		X							
	역할 관리		X							
	액세스 그룹 관리		X							
	등록정보 액세스 관리		X							
워크플로우 참가자										X

## 분석 역할

Oracle Data Relationship Management Analytics 역할을 결합하여 여러 기능을 지원할 수 있습니다. 예를 들어 분석 사용자, 관리자 및 데이터 관리자 역할을 가진 사용자는 모든 대시보드 및 관리 콘솔에 액세스할 수 있습니다. 액세스 관리자 및 애플리케이션 관리자 역할을 가진 사용자는 모든 보고서에 액세스할 수 있습니다.

표 4-15 대시보드 및 관리 콘솔의 분석 역할

역할	대시보드 및 관리 콘솔					권한
	요청	모델	변경	증가	관리 콘솔	
분석 사용자			X	X		버전 및 계층 구조 찾아보기
관리자	X	X				버전 및 계층 구조 찾아보기
애플리케이션 관리자					X	해당 없음
데이터 관리자					X	해당 없음

표 4-16 보고서 분석 역할

역할	보고서						
	사용자 역할 지정	액세스 그룹 멤버십	계층 액세스 그룹 지정	워크플로우 액세스 그룹 지정	객체 액세스 그룹 권한 부여	사용자 로그인 활동	메타데이터 객체 사용
액세스 관리자	X	X		X	X	X	
애플리케이션 관리자			X	X			X
데이터 관리자			X	X			

## 사용자 생성

사용자를 생성할 때 경우 고유 이름을 정의하고 역할을 지정합니다. 사용자에게 데이터 관리자 역할이 지정되지 않은 경우 노드 액세스 그룹과 등록정보 범주를 사용자에게 지정하여 데이터 액세스를 제어할 수 있습니다.

 주:

사용자 ID의 @@ 접두어는 내부 전용 사용자를 나타냅니다. 이 접두어를 가진 사용자는 생성할 수 없습니다. 다른 @@ 사용자로 @@SYSTEM 및 @@STANDARD가 있습니다.

사용자를 생성하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **새로 작성**에서 **사용자**를 선택합니다.
3. 고유한 사용자 이름과 사용자의 전체 이름을 입력합니다.

 **주:**

부서, 전화 및 전자메일 주소는 선택 사항입니다. 데이터 관리 워크플로우 사용자는 전자메일 통지를 수신하도록 구성된 전자메일 주소가 있어야 합니다.

4. Oracle Data Relationship Management 애플리케이션에 대해 혼합 인증을 사용하는 경우 사용자 인증 방법을 선택합니다.
  - **내부** – 사용자가 Data Relationship Management 내에서 인증됩니다.
  - **CSS(외부)** - 사용자가 Oracle Hyperion Shared Services를 통해 외부적으로 인증됩니다.
5. **선택 사항:** 다음 옵션 중에서 선택합니다.
  - **비밀번호가 만료되지 않음** – PasswordDuration 시스템 환경설정 지정이 무시됩니다.
  - **로그인 세션이 만료되지 않음** – IdleTime 시스템 환경설정 지정이 무시됩니다.

 **주:**

이 옵션이 선택된 경우 허용되는 최대 유효 시간은 24시간입니다. 24시간의 유효 시간 후에 로그인 세션이 만료됩니다.

- **사용자가 잠금 측정항목에서 면제됨** – 이 사용자에게 대해 잠금 제한이 무시됩니다.
6. **역할 탭의 사용가능** 목록에서 사용자에게 지정할 역할을 선택합니다. 화살표를 사용하여 역할을 **선택됨** 목록으로 이동합니다.

 **주:**

역할에 대한 자세한 내용은 [사용자 역할](#)을 참조하십시오.

7. **노드 액세스 그룹** 탭의 **사용가능** 목록에서 사용자에게 지정할 그룹을 선택합니다. 화살표를 사용하여 그룹을 **선택됨** 목록으로 이동합니다.
8. **등록정보 범주** 탭의 **사용가능** 목록에서 사용자에게 지정할 범주를 선택합니다. 화살표를 사용하여 범주를 **선택됨** 목록으로 이동합니다.
9. 선택된 목록의 각 범주에 대해 다음을 수행합니다.
  - a. **작업 열의**  을 누르고 범주에 대한 사용자 액세스 권한(읽기 또는 편집)을 설정합니다.
  - b. **작업 열의**  을 선택하여 변경사항을 저장합니다.
10.  을 누릅니다.  
[비밀번호 변경] 대화상자가 표시됩니다.
11. 사용자의 비밀번호를 입력합니다.

12. 비밀번호를 다시 입력합니다.
13. **선택사항: 사용자가 다음 로그인 시 비밀번호를 변경해야 함**을 선택하여 사용자가 다음 로그인 시 비밀번호를 변경하도록 해야 합니다.
14. 확인을 누릅니다.

## 사용자 인증

Oracle Data Relationship Management는 저장된 비밀번호 정보를 사용하여 애플리케이션에서 기본적으로 인증된 사용자 또는 외부 사용자 디렉토리에서 인증된 사용자를 지원합니다. 각 Data Relationship Management 애플리케이션은 두 사용자 유형 중 하나 또는 둘 다를 지원하도록 구성되었습니다.

Data Relationship Management 콘솔의 [인증 설정] 탭에서 애플리케이션 인증을 설정합니다. 자세한 내용은 *Oracle Data Relationship Management 설치 가이드*를 참조하십시오.

다음 시스템 환경설정에 대해 정의된 값은 사용자 비밀번호의 특성 및 인증된 내부 사용자의 비밀번호가 만료되는 시기를 결정합니다.

- PasswordPolicyEnabled – 활성화된 경우 비밀번호에 다음 요소 중 3개가 포함되어야 합니다.
  - 대문자
  - 소문자
  - 숫자
  - 특수 문자
- PasswordMaxLength – 비밀번호의 최대 문자 길이를 결정합니다.
- PasswordMinLength – 비밀번호의 최소 문자 길이를 결정합니다.
- PasswordDuration – 비밀번호가 적합한 일수를 결정합니다.
- PasswordWarningPeriod – 사용자가 더 이상 로그인할 수 없게 되기 전에 비밀번호를 변경하도록 경고할 비밀번호 만료 날짜 이전(-) 또는 이후(+) 일수를 나타냅니다. 음수 값(예: -3)은 비밀번호 만료 이전 3일 동안 로그인 시 사용자에게 경고가 표시됨을 나타냅니다. 양수 값(예: 5)은 비밀번호 만료 이후 5일 동안 로그인 시 사용자에게 경고가 표시됨을 나타냅니다. 5일 기간 후에는 사용자가 비밀번호를 변경하지 않고 로그인할 수 없습니다.

### 주:

PasswordDuration 및 PasswordWarningPeriod 값의 변경사항은 다음 비밀번호 변경 시까지 사용자에게 영향을 주지 않습니다. 예를 들어 PasswordDuration이 30일로 설정되고 User1의 비밀번호가 26일 전에 변경된 경우 비밀번호가 4일 후에 만료됩니다. PasswordDuration 값을 60일로 변경해도 User1의 비밀번호는 4일 후에 만료됩니다. 사용자가 비밀번호를 변경한 후 새 비밀번호는 60일 후에 만료됩니다.

## 사용자 수정

사용자 비밀번호를 변경하거나 사용자를 잠금 또는 잠금 해제하거나 역할, 그룹 또는 범주 지정을 변경할 수 있습니다.

### 비밀번호 변경

사용자 비밀번호를 변경하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **보안**에서 **사용자**를 확장합니다.
3. 사용자를 선택하고  을 누릅니다.
4.  을 누릅니다.
5. 사용자의 새 비밀번호를 입력합니다.
6. 비밀번호를 다시 입력합니다.
7. **선택사항: 사용자가 다음 로그인 시 비밀번호를 변경해야 함**을 선택하여 사용자가 다음 로그인 시 비밀번호를 변경하도록 해야 합니다.
8. **확인**을 누릅니다.

### 사용자 잠금

사용자를 잠가 Oracle Data Relationship Management 애플리케이션에 대한 액세스를 차단할 수 있습니다. 사용자를 잠글 때 사용자정의 잠금 이유를 입력할 수 있습니다. 이 이유는 사용자가 애플리케이션에 로그인할 때 표시됩니다.

사용자를 잠그려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **보안**에서 **사용자**를 확장합니다.
3. 사용자를 선택하고  을 누릅니다.
4.  을 누릅니다.
5. 잠금 이유를 입력합니다.
6. **확인**을 누릅니다.

### 사용자 잠금 해제

잠긴 사용자를 잠금 해제하면 애플리케이션에 대한 액세스가 활성화됩니다.

사용자 잠금을 해제하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **보안**에서 **사용자**를 확장합니다.
3. 사용자를 선택하고  을 누릅니다.

4.  을 누릅니다.
5. **확인**을 누릅니다.

## 사용자 역할 및 지정 변경

사용자 역할 및 지정을 변경하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **보안**에서 **사용자**를 확장합니다.
3. 사용자를 선택하고  을 누릅니다.
4. **역할** 탭의 **사용가능** 목록에서 사용자에게 지정할 역할을 선택합니다. 화살표를 사용하여 역할을 **선택됨** 목록으로 이동합니다.
5. **노드 액세스 그룹** 탭의 **사용가능** 목록에서 사용자에게 지정할 그룹을 선택합니다. 화살표를 사용하여 그룹을 **선택됨** 목록으로 이동합니다.
6. **등록정보 범주** 탭의 **사용가능** 목록에서 사용자에게 지정할 범주를 선택합니다. 화살표를 사용하여 범주를 **선택됨** 목록으로 이동합니다.
7. 선택됨 목록의 각 범주에 대해 다음을 수행합니다.
  - a.  을 누르고 범주에 대한 사용자 액세스 권한(읽기 또는 편집)을 설정합니다.
  - b.  을 선택하여 변경사항을 저장합니다.
8.  을 누릅니다.

## 사용자 삭제

더 이상 활성 상태가 아닌 사용자를 애플리케이션에서 삭제할 수 있습니다. 사용자를 삭제하면 사용자와 연결된 사용자 레벨 메타데이터 객체도 모두 삭제됩니다. 이러한 메타데이터 객체에는 쿼리, 비교, 임포트, 혼합기, 익스포트 및 장부가 포함됩니다.

사용자를 삭제하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **보안**에서 **사용자**를 확장합니다.
3. 사용자를 선택하고  을 누릅니다.
4. **이 항목 삭제**를 눌러 삭제를 확인합니다.

## 사용자 로그인 상태 보기

각 사용자에게 대해 로그인 통계와 정보를 볼 수 있습니다.

- 사용자의 최종 유효 로그인 날짜 및 시간
- 부적합한 로그인 시도 횟수
- 사용자가 잠겼는지 여부
- 사용자가 잠긴 날짜 및 시간
- 잠금 이유

사용자 로그인 상태를 보려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **보안**에서 **사용자**를 확장합니다.
3. 사용자를 선택하고  을 누릅니다.
4. **로그인 상태** 탭을 선택합니다.

## 시스템 정의 사용자

Oracle Data Relationship Management 애플리케이션에는 애플리케이션 저장소를 생성하는 동안 추가된 네 명의 기본 사용자가 포함되어 있습니다.

- **ADMIN** - 애플리케이션의 기본 관리 사용자입니다. 이 사용자의 비밀번호는 저장소 생성 프로세스 중에 처음으로 구성됩니다.
- **@PROCESS** - 서버 구성요소 사이의 프로세스 간 통신을 처리하도록 설정된 내부 사용자입니다. 이 사용자는 Web Client에서 액세스하거나 구성할 수 없습니다. 애플리케이션 엔진이 시작될 때마다 이 사용자에게 대한 트랜잭션이 기록됩니다.
- **@STANDARD** - 표준 객체 액세스 그룹에서 사용자 메타데이터 객체를 관리하도록 설정된 내부 사용자입니다. 이 사용자는 Web Client에서 액세스하거나 구성할 수 없습니다.
- **@SYSTEM** - 시스템 객체 액세스 그룹에서 사용자 메타데이터 객체를 관리하도록 설정된 내부 사용자입니다. 이 사용자는 Web Client에서 액세스하거나 구성할 수 없습니다.

## 공통 사용자 프로비저닝

공통 사용자 프로비저닝 기능을 사용하는 경우 Oracle Hyperion Shared Services를 사용하여 Oracle Data Relationship Management 애플리케이션에 사용자 및 그룹이 프로비저닝될 수 있습니다. 이 구성은 Data Relationship Management 사용자가 다른 Oracle EPM 애플리케이션과의 공통 위치에 프로비저닝될 수 있도록 합니다. 또한 공통 사용자 프로비저닝을 사용하면 Data Relationship Management 애플리케이션에서 사용자를 별도로 프로비저닝할 필요가 없습니다. 필요 시 또는 예약을 통해 Shared Services와 Data Relationship Management 간에 프로비저닝 정보를 동기화할 수 있습니다.

동기화가 진행되면 Data Relationship Management에서 다음 작업이 수행됩니다.

- 사용자 추가 또는 업데이트
  - 사용자 이름
  - 전체 이름
  - 전자메일 주소
- 사용자에게 역할 지정
- 노드 액세스 그룹에 사용자 지정
- 등록정보 범주에 사용자 지정
- 사용자 역할 제거(Shared Services에서 프로비저닝 해제된 경우)

공통 사용자 프로비저닝을 사용으로 설정하면 모든 외부 Data Relationship Management 사용자 및 사용자의 역할이 Shared Services에서 관리되며 Data Relationship Management에서는 관리할 수 없습니다.

## 사전 필수 조건

공통 사용자 프로비저닝은 기본적으로 Oracle Data Relationship Management에서 사용되지 않으며 반드시 다음 사전 필수 조건 단계를 완료한 후 설정해야 합니다.

1. Oracle Hyperion Shared Services에서 Data Relationship Management 사용자 역할을 추가합니다(*Oracle Data Relationship Management 설치 가이드*의 "Data Relationship Management 사용자 역할로 Shared Services 구성" 참조).
2. Shared Services에 Data Relationship Management 애플리케이션을 등록합니다(*Oracle Data Relationship Management 설치 가이드*의 "EPM 레지스트리 설정 구성" 참조).
3. 공통 사용자 프로비저닝을 사용할 수 있도록 설정합니다(*Oracle Data Relationship Management 설치 가이드*의 "공통 사용자 프로비저닝 구성" 참조).

## 사용자 및 그룹 프로비저닝

Oracle Hyperion Shared Services에서 액세스할 수 있는 모든 사용자 또는 그룹은 공통 사용자 프로비저닝을 사용하여 Oracle Data Relationship Management 애플리케이션에 프로비저닝할 수 있습니다. Data Relationship Management 애플리케이션에 대해 그룹(그룹 및/또는 사용자 포함) 및 개별 사용자를 프로비저닝할 수 있습니다. Shared Services에서 Data Relationship Management 애플리케이션에 대해 프로비저닝된 사용자 및 그룹은 동기화 태스크가 실행될 때 Data Relationship Management에서 동기화됩니다. 등록된 여러 Data Relationship Management 애플리케이션에 사용자를 개별적으로 프로비저닝할 수 있습니다.

Oracle EPM System 사용자 보안 관리 가이드에서 "사용자 및 그룹 프로비저닝"을 참조하십시오.

## Data Relationship Management 사용자 및 그룹 멤버십 동기화

Oracle Hyperion Shared Services에서 Oracle Data Relationship Management 애플리케이션으로의 사용자 및 그룹 변경에 대한 전체 동기화는 수동으로 실행하거나 백그라운드에서 실행되도록 예약할 수 있습니다. 동기화하면 Data Relationship Management 애플리케이션에서 사용자가 생성되거나 업데이트되며, 외부에서 관리하도록 구성된 등록정보 범주나 노드 액세스 그룹에 대한 그룹 멤버십이 업데이트됩니다.

동기화 결과에는 생성 및 업데이트된 사용자 수, 업데이트된 노드 액세스 그룹 수, 업데이트된 등록정보 범주 수가 표시됩니다. 동기화를 실행하는 동안 생성된 오류 및 경고 메시지 목록도 표시됩니다. 추가 검토 및 사용을 위해 결과를 복사하여 외부 편집기에 붙여넣을 수 있습니다.

## 수동 동기화

Oracle Data Relationship Management에서 공통 사용자 프로비저닝을 사용하는 경우 액세스 관리자 역할의 사용자가 Oracle Hyperion Shared Services에서 관리되는 사용자 및 그룹을 수동으로 동기화할 수 있습니다. 작업 결과가 표시되며 감사 태스크의 작업 페이지에서도 이 결과를 볼 수 있습니다.

사용자 및 그룹을 수동으로 동기화하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.

2. 툴바에서  (공통 사용자 프로비저닝 동기화)을 선택합니다.

## 스케줄링된 동기화

Oracle Data Relationship Management에서 공통 사용자 프로비저닝을 사용하는 경우 24시간마다 지정된 시간에 백그라운드에서 동기화가 실행되도록 예약할 수 있습니다. 감사 태스크의 작업 페이지에서 해당 작업으로 이동하여 스케줄링된 작업의 결과를 볼 수 있습니다.

- 작업 보기에 대한 자세한 내용은 *Oracle Data Relationship Management 사용자 가이드*의 "작업 내역 보기"를 참조하십시오.
- 동기화 예약에 대한 자세한 내용은 *Oracle Data Relationship Management 설치 가이드*의 "공통 사용자 프로비저닝 구성"을 참조하십시오.

## 부분 동기화

이 시나리오에서는 Oracle Hyperion Shared Services에서 관리되는 사용자 및 그룹에 대해 실시간 부분 동기화가 자동으로 수행됩니다.

- 사용자 로그인 - 세션이 생성되기 전에 인증하려는 개별 사용자에게 대한 프로비저닝 정보가 자동으로 동기화됩니다.
- 노드 액세스 그룹 멤버십 - 그룹을 저장하면 개별 노드 액세스 그룹에 대한 사용자 멤버십이 자동으로 동기화됩니다.
- 등록정보 범주 멤버십 - 범주를 저장하면 개별 등록정보 범주에 대한 사용자 멤버십이 자동으로 동기화됩니다.

# 5

## 노드 액세스 그룹 관리

Oracle Data Relationship Management는 노드 액세스 그룹을 사용하여 계층 노드 및 해당 등록정보에 대한 세분화된 사용자 액세스를 제어합니다. Data Relationship Management 버전 내 계층 구조 서브세트의 특정 노드에 대한 액세스 권한이 부여된 그룹에 사용자를 지정할 수 있습니다. 노드 액세스 그룹은 상속을 사용하여 액세스 레벨이 명시적으로 지정된 계층 노드의 하위 멤버 노드에 유사한 액세스 권한을 지정합니다. 이 액세스 레벨을 하위 레벨에서 재정의하거나, 재정의의 차단하기 위해 잠글 수 있습니다.

일반적으로 노드 액세스 그룹은 조직의 기능 영역을 나타내며 한 사용자에게 여러 그룹에 대한 지정이 필요할 수 있습니다. 지정된 액세스 레벨이 충돌할 경우 가장 높은 보안 레벨이 사용됩니다.

두 가지 유형의 노드 액세스 그룹이 있습니다. 그룹 유형은 해당 그룹의 사용자에게 지정될 수 있는 데이터 액세스 유형을 제어합니다. 각 노드 액세스 그룹은 단일 그룹 유형일 수만 있습니다.

- 대화식 - 사용자가 지정된 액세스 레벨에 따라 데이터 찾아보기, 검색 및 수정에 직접 액세스할 수 있습니다.
- 워크플로우 - 사용자가 지정된 액세스 레벨에 따라 관리 워크플로우를 사용하여 데이터 찾아보기, 검색 및 수정에 제한적으로 액세스할 수 있습니다.

**표 5-1 대화식 그룹 유형-노드 액세스 레벨**

레벨	설명	사용 예
읽기	읽기 전용 액세스를 활성화합니다. 변경은 허용되지 않습니다.	보기 및 보고
제한된 삽입	사용자에게 (최소한) 글로벌 삽입 권한이 있는 노드를 삽입할 수 있게 합니다.	삽입
편집	등록정보 값을 편집할 수 있게 합니다.	편집
삽입	노드를 삽입, 이동 또는 제거할 수 있게 합니다.	편집, 삽입, 복사, 이동, 제거
비활성화	노드를 비활성화 및 재활성화할 수 있게 합니다.	편집, 삽입, 이동, 제거, 비활성화, 재활성화
Add	노드를 추가 또는 삭제할 수 있게 합니다.	편집, 삽입, 복사, 이동, 제거, 비활성화, 재활성화, 추가, 삭제

다음 사항에 유의하십시오.

- 액세스 레벨은 누적됩니다. 편집 액세스 레벨을 지정할 경우 읽기 전용 및 제한된 삽입 액세스 레벨도 부여됩니다. 추가 액세스 레벨을 지정할 경우 다른 모든 액세스 레벨도 부여됩니다.
- 노드 액세스 그룹 보안은 계층 레벨에서만 적용됩니다. 노드 액세스 그룹은 단독 등의 글로벌 노드 목록에 대한 액세스를 제어하지 않습니다.
- 림 및 리프 노드에 대해 별도로 액세스 레벨이 지정되므로 각각 다른 액세스 레벨을 정의할 수 있습니다. 이 기능은 사용자가 계층의 롤업 구조를 유지할 수 있어야 하지만 리프 노드의

등록정보를 편집할 수 없어야 하는 경우 또는 사용자가 기존 롤업 구조에 리프 노드를 삽입할 수 있지만 구조 자체를 재구조화할 수는 없는 경우에 유용합니다.

- 액세스 관리자 역할이 부여된 사용자만 노드 액세스 그룹을 정의합니다.
- 노드 액세스 그룹은 관련 노드에 대한 액세스 지정을 위해 로컬 상속을 사용합니다. 제어 계층에 지정된 액세스 레벨에 따라 글로벌 상속을 사용하기 위해 노드 액세스 그룹을 글로벌로 정의할 수 있습니다.
- 글로벌 노드 액세스 그룹을 생성할 수 있으며, 이 그룹에는 각 버전에 대해 정의된 제어 계층이 있어야 합니다. 이 작업을 위해 제어된 노드 액세스 그룹을 계층에 지정합니다. 자세한 정보는 *Oracle Data Relationship Management 사용자 가이드*를 참조하십시오.
- 대화식 및 워크플로우 노드 액세스 그룹은 계층 구조의 노드 표시를 서로 다르게 처리합니다. 대화식 액세스 그룹은 그룹에 액세스 권한이 있는 노드가 계층에 하나라도 있으면 사용자에게 전체 계층을 표시합니다. 반면, 워크플로우 액세스 그룹은 액세스 권한이 지정된 계층 구조의 노드만 제한적으로 사용자에게 표시합니다. 두 그룹 유형에서 모두, 그룹의 멤버는 액세스 권한이 지정되지 않은 계층 구조를 볼 수 없습니다.

## 워크플로우 그룹 유형 노드 액세스 레벨

관리 사용자 역할을 가진 사용자는 워크플로우 노드 액세스 레벨을 사용하여 데이터 액세스 권한을 결정합니다.

표 5-2 워크플로우 그룹 유형-노드 액세스 레벨

레벨	설명
통지	노드에 대한 변경 요청 통지를 활성화합니다.
제출	변경 요청의 일부로 노드를 제출할 수 있습니다.
승인	변경 요청의 일부로 노드를 승인할 수 있습니다.
강화	변경 요청의 일부로 노드를 강화할 수 있습니다.
커밋	노드에 대한 변경사항을 Oracle Data Relationship Management에 커밋할 수 있습니다.

워크플로우 노드 액세스 레벨은 계층 액세스에 대해 누적되지만 워크플로우 단계를 기준으로 필터링되기도 합니다.

표 5-3 계층 액세스별 워크플로우 노드 액세스 레벨

계층 액세스	단계 액세스			
	제출	승인	강화	커밋
액세스	제출	승인	강화	커밋
통지	통지	통지	통지	통지
제출	제출	통지	통지	통지
승인	제출	승인	통지	통지
강화	제출	승인	강화	통지
커밋	제출	승인	강화	커밋

## 노드 액세스 그룹 생성

노드 액세스 그룹을 생성하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **작성**에서 **노드 액세스 그룹**을 선택합니다.
3. 그룹의 이름, 레이블 및 설명을 입력합니다.

### 주:

노드 액세스 그룹은 Custom 네임스페이스에 지정됩니다. 그룹의 전체 이름은 고유해야 합니다. 이름을 입력하면 [레이블] 필드가 자동으로 채워집니다. 노드 액세스 그룹 레이블은 애플리케이션 관리 옆의 모든 기능에 대해 표시되는, 사용자에게 친숙한 기술자입니다. 편의상 여러 노드 액세스 그룹이 동일한 레이블을 사용할 수 있습니다.

4. 노드 액세스 그룹에 대한 **그룹 유형**을 선택합니다.
  - **대화식** – 대화식 액세스 레벨을 사용하려면 **대화식 노드 액세스 레벨**을 참조하십시오.
  - **워크플로우** – 요청 제출, 강화, 승인, 커밋 및 통지 제공의 컨텍스트에서 버전, 계층 구조 및 노드에 대한 워크플로우 지향 액세스를 사용합니다. **워크플로우 노드 액세스 레벨**을 참조하십시오.
5. **선택사항**: **글로벌**을 선택하여 그룹을 글로벌 노드 액세스 그룹으로 설정합니다.

### 주:

글로벌 노드 액세스 그룹에는 그룹이 사용되는 각 버전에 정의된 제어 계층이 있어야 합니다. 그룹이 생성된 후 각 버전의 단일 계층에 그룹을 제어된 노드 액세스 그룹으로 지정할 수 있습니다.

6. 공동 사용자 프로비저닝을 사용하는 경우 **외부 그룹**에서 Oracle Hyperion Shared Services의 Oracle Data Relationship Management 애플리케이션에 프로비저닝된 사용자 그룹을 선택합니다. Shared Services에서 동기화가 진행되는 경우 이러한 외부 그룹의 사용자에게 노드 액세스 그룹에 대한 멤버십이 지정됩니다.
7. **사용 가능** 목록에서 그룹에 지정할 사용자를 선택합니다. 화살표를 사용하여 사용자를 **선택됨** 목록으로 이동합니다.
8.  을 누릅니다.

## 노드 액세스 그룹 편집

노드 액세스 그룹을 편집하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **보안**에서 **노드 액세스 그룹**을 확장합니다.
3. 그룹을 선택하고  을 누릅니다.

4. **사용 가능** 목록에서 그룹에 지정할 사용자를 선택합니다. 화살표를 사용하여 사용자를 **선택됨** 목록으로 이동합니다.
5.  을 누릅니다.

## 노드 액세스 그룹 삭제

노드 액세스 그룹을 삭제하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **보안**에서 **노드 액세스 그룹**을 확장합니다.
3. 그룹을 선택하고  을 누릅니다.
4. **이 항목 삭제**를 눌러 삭제를 확인합니다.

### 주:

노드 액세스 그룹을 삭제하면 계층 노드는 물론 사용자에서 그룹 지정이 제거됩니다.

## 노드 액세스 그룹 보안 지정

노드 액세스 그룹 보안은 데이터 관리자 역할이 부여된 사용자에 의해 데이터에 적용됩니다.

### 주:

노드 액세스 그룹 보안을 지정하기 전에 적절한 노드 액세스 그룹이 생성되었으며 적절한 사용자가 그룹에 지정되었는지 확인합니다.

노드 액세스 그룹 보안을 설정하려면 다음을 수행합니다.

1. 버전 및 계층을 열고 노드를 선택합니다.
2. 노드에서 **지정**, **노드 액세스** 순으로 선택합니다.
3. 등록정보 그리드에서 [리프 액세스] 또는 [림 액세스] 범주를 선택합니다.
4. 각 노드 액세스 그룹에 대해 액세스 레벨을 지정합니다.

각 노드 액세스 그룹에 지정할 수 있는 액세스 레벨은 해당 그룹 유형(대화식 또는 워크플로우)을 기반으로 합니다.

5. **저장**을 누릅니다.

 주:

DRG 노드 선택기에서 선택할 노드를 시각화할 수 있으려면 리프 노드 1개 이상과 선행 리프 노드 1개에 대한 리프 및 리프 워크플로우 NAG 액세스를 둘 다 "없음" 이외의 값으로 지정해야 합니다. 일반적으로 리프 WNAG 액세스는 계층의 최상위 노드에서 설정됩니다.

# 6

## 객체 액세스 그룹 관리

Oracle Data Relationship Management의 객체 액세스 그룹은 익스포트, 장부, 임포트, 혼합기, 비교, 쿼리, 버전 변수 및 외부 연결을 포함하여 사용자가 액세스할 수 있는 메타데이터 객체를 결정합니다.

표 6-1 객체 액세스 그룹 유형

객체 액세스 그룹 유형	설명	권한
사용자	각 사용자에게는 개인 메타데이터 객체에 대한 핵심 객체 액세스 그룹이 있습니다.	사용자는 자체 객체 액세스 그룹에 대한 실행 및 관리 권한을 보유합니다.
표준	표준이라는 핵심 객체 액세스 그룹은 모든 공용 객체에 사용할 수 있습니다.	모든 사용자는 표준 객체 액세스 그룹의 객체에 대한 암시적 실행 권한을 보유합니다. 표준 [객체] 관리 역할 권한이 있는 사용자만 표준 객체 액세스 그룹에 대한 관리 권한을 보유합니다.
시스템	시스템이라는 핵심 객체 액세스 그룹은 모든 시스템 작업/통합 객체에 사용할 수 있습니다.	데이터 관리자 또는 애플리케이션 관리자 역할을 가진 사용자만 시스템 객체 액세스 그룹에 대한 관리 권한을 보유합니다.
사용자정의	사용자정의 객체 액세스 그룹	액세스 관리자 역할을 가진 사용자만 사용자정의 객체 액세스 그룹을 생성, 편집 또는 삭제할 수 있습니다. 실행 권한을 가진 사용자는 그룹의 객체를 실행할 수 있습니다.

사용자정의 객체 액세스 그룹은 사용자 메타데이터 객체(쿼리, 비교, 임포트, 혼합기, 익스포트 및 장부)의 서브셋에 대한 액세스 권한을 특정 사용자 그룹에 제공합니다. 객체 액세스 그룹은 사용자 및 노드 액세스 그룹 목록을 정의하고 각 사용자 및 노드 액세스 그룹에 대한 권한 레벨(실행 또는 관리)을 설정합니다. 메타데이터 객체는 생성 시 객체 액세스 그룹에 지정되며, 이후에 다른 그룹으로 복사하거나 이동할 수 있습니다.

- 실행 - 사용자가 그룹의 객체를 실행할 수 있지만 편집하고 객체 변경사항을 저장할 수 없습니다.
- 관리 - 사용자가 그룹의 객체를 생성, 편집 또는 삭제하고 실행할 수 있습니다.

다음은 객체 액세스 그룹 사용에 대한 지침입니다.

- 객체 액세스 그룹을 사용하면 사용자가 직접 또는 해당 노드 액세스 그룹 지정을 통해 그룹의 멤버가 될 수 있습니다. 둘 다 필수는 아닙니다.
- 사용자 및 노드 액세스 그룹은 둘 이상의 객체 액세스 그룹에 지정될 수 있습니다.
- 객체 액세스 그룹의 각 사용자에게 관리 또는 실행 권한이 지정됩니다.

- 객체 액세스 그룹에서 사용자의 권한 지정이 사용자의 역할 보안을 재정의할 수 있습니다. 예를 들어 객체 액세스 그룹에서 관리 권한이 있는 대화식 사용자 역할은 객체 액세스 그룹 내에서 객체를 생성하거나 수정할 수 있습니다.
- 사용자, 표준, 시스템 등의 핵심 객체 액세스 그룹은 사용자 존재 및 해당 역할 지정에 따라 암시적으로 관리됩니다.
- 사용자 메타데이터 객체를 저장하거나 복사할 경우 사용자는 관리 권한을 보유한 객체 액세스 그룹에 객체를 지정해야 합니다.
- 사용자 메타데이터 객체는 객체 액세스 그룹 하나에만 지정될 수 있습니다.
- 데이터 관리자 역할 사용자는 핵심 표준 객체 액세스 그룹에 대한 암시적 관리 권한을 보유하며 사용자정의 객체 액세스 그룹에 명시적으로 지정될 수 있습니다.
- 애플리케이션 관리자 역할 사용자는 모든 표준, 시스템 및 사용자정의 객체 액세스 그룹에 대한 암시적 관리 권한을 보유합니다. 이러한 사용자는 객체 액세스 그룹에 대한 메타데이터 객체를 마이그레이션할 수 있어야 합니다.

## 객체 액세스 그룹 생성

사용자정의 객체 액세스 그룹을 생성하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **새로 작성**에서 **객체 액세스 그룹**을 선택합니다.
3. 그룹의 이름을 입력합니다. 설명은 선택 사항입니다.
4. **사용자** 탭의 **사용가능** 목록에서 그룹에 지정할 사용자를 선택합니다. 화살표를 사용하여 사용자를 **선택됨** 목록으로 이동합니다.

### 주:

기본적으로 각 사용자에게 실행 액세스 권한이 부여됩니다. 사용자의 액세스 권한을 변경하려면  을 누릅니다. **액세스**에서 **관리**를 선택합니다.

5. **노드 액세스 그룹** 탭의 **사용가능** 목록에서 그룹에 지정할 노드 액세스 그룹을 선택합니다. 화살표를 사용하여 노드 액세스 그룹을 **선택됨** 목록으로 이동합니다.

### 주:

기본적으로 각 노드 액세스 그룹에게 실행 액세스 권한이 부여됩니다. 그룹의 액세스 권한을 변경하려면  을 누릅니다. **액세스**에서 **관리**를 선택합니다.

6.  을 누릅니다.

## 객체 액세스 그룹 편집

사용자정의 객체 액세스 그룹을 편집하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **보안**에서 **객체 액세스 그룹**을 확장합니다.

3. 그룹을 선택하고  을 누릅니다.
4. 사용자 및 노드 액세스 그룹 탭에서 선택된 사용자와 그룹 및 액세스 권한을 변경합니다.
5.  을 누릅니다.

## 객체 액세스 그룹 삭제

객체 액세스 그룹을 삭제하려면 다음을 수행합니다.

1. 홈 페이지에서 관리를 선택합니다.
2. 보안에서 객체 액세스 그룹을 확장합니다.
3. 그룹을 선택하고  을 누릅니다.
4. 이 객체 액세스 그룹 삭제를 눌러 삭제를 확인합니다.

### ▲ 주의:

객체 액세스 그룹이 삭제되면 지정된 모든 메타데이터 객체도 삭제됩니다. 이 연산은 실행취소할 수 없습니다

# 7

## 도메인 관리

도메인은 동일한 Oracle Data Relationship Management 애플리케이션 내에서 다양한 소스의 여러 노드 세트에 대해 참조 무결성을 관리하는 데 사용됩니다. 도메인은 동일한 애플리케이션 내에서 이러한 노드의 여러 버전을 일관되게 관리할 수 있도록 해주는 공통 유형의 등록된 노드 목록입니다. 도메인은 다음 작업을 수행하는 간단한 방법을 제공합니다.

- 노드 이름을 한정하여 고유성 보장
- 다양한 버전에서 식별 등록정보 공유
- 이름 바꾸기, 승격, 강등, 노드 삭제 등 특정 유형의 변경 제한
- 검증을 지정하여 버전에 관계없이 비즈니스 규칙의 일관성 보장

도메인 노드는 도메인에 대한 멤버십이 있는 버전의 글로벌 노드입니다. 도메인 노드를 멤버로 지정한 후에는 도메인에서 제거할 수 없으며 이름을 바꿀 수 없습니다. 도메인 지정에 관계없이 도메인 노드에는 고유 이름이 있어야 합니다. 도메인 노드 이름은 노드의 중립 식별자를 나타내거나, 동일한 버전의 다른 도메인 노드와 함께 사용할 때 참조 무결성을 보장하기 위해 접두어 또는 접미어로 한정될 수 있습니다. 도메인 노드 설명과 비활성 상태/날짜는 도메인 노드가 있는 임의 버전의 도메인 노드에서 공유됩니다.

## 도메인 생성

도메인을 생성하려면,

1. 홈 페이지에서 **관리**를 선택합니다.
2. **새로 작성**에서 **도메인**을 선택합니다.
3. 다음 정보를 입력합니다.

- **이름**
- **설명**(선택사항)
- **식별자**(선택 사항) – 노드 이름을 한정하는 데 사용되는 텍스트입니다. 두 도메인이 동일한 식별자 텍스트를 사용할 수는 없습니다. **접두어** 또는 **접미어**를 선택하여 식별자의 위치를 나타냅니다.

### 주:

도메인에 지정된 노드가 있는 경우 식별자 텍스트를 변경할 수 없습니다.

- **구분자**(선택 사항) – 노드 이름에서 도메인 식별자 텍스트를 구분하는 데 사용되는 단일 선택적 문자입니다.

 주:

도메인에 지정된 노드가 있는 경우 구분자를 변경할 수 없습니다.

- **노드 삭제 허용** – 사용자가 버전에서 노드를 삭제할 수 있게 하려는 경우 선택합니다.
  - **리프 편집 허용** – 사용자가 도메인에 있는 노드의 리프 시스템 등록정보 값을 변경할 수 있게 하려는 경우 선택합니다.
4. **사용가능한 검증 목록**에서 도메인 멤버에 대해 적용할 노드 레벨 검증을 선택하고 **선택한 검증 목록**으로 이동합니다.

 주:

도메인 레벨 검증 지정은 노드에 설정되었거나 상위 멤버 노드, 계층 또는 버전 레벨 지정에서 상속된 것과 동일한 검증의 지정 값을 재정의합니다.

5.  을 누릅니다.

## 도메인 편집

도메인을 생성한 후 편집할 수 있지만 다음 두 가지 예외가 있습니다.

- 이름은 변경할 수 없습니다.
- 도메인에 노드가 지정된 후에는 식별자와 구분자를 변경할 수 없습니다.

도메인을 편집하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. 도메인을 선택하고  을 누릅니다.
3. 도메인을 변경하고  을 누릅니다.

## 도메인 삭제

도메인을 삭제할 수 있습니다. 도메인을 삭제하면 도메인 노드 레코드도 제거됩니다.

 주:

노드가 지정된 도메인을 삭제하면 도메인에 지정된 모든 노드도 비도메인 노드로 돌아갑니다.

도메인을 삭제하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. 도메인을 선택하고  을 누릅니다.

3. 이 도메인 삭제를 선택합니다.

# 8

## 등록정보 범주 관리

참조:

- [등록정보 범주](#)
- [등록정보 범주 생성](#)
- [등록정보 범주 편집](#)
- [등록정보 범주 삭제](#)

### 등록정보 범주

등록정보 범주를 사용하면 Oracle Data Relationship Management 등록정보를 그룹화할 수 있으며 등록정보 세트에 대한 보안 권한 지정을 제어할 수 있습니다. 기본적으로 사용 가능한 핵심 등록정보는 하나의 등록정보 범주에만 있습니다. 애플리케이션 관리자가 생성한 사용자정의 등록정보는 여러 등록정보 범주에 연결할 수 있습니다.

Data Relationship Management에는 다음 표에 설명된 핵심 등록정보 범주가 있습니다.

표 8-1 등록정보 범주

범주	설명
시스템	노드의 기본 식별 특성과 관련된 등록정보입니다(예: ID, 이름 및 설명). 이 범주에 대해 수행할 수 있는 변경사항은 개별 사용자에게 읽기 전용 플래그를 지정하는 것뿐입니다. 읽기 액세스 권한을 가진 사용자는 값을 편집할 수 없고 볼 수만 있습니다. 이 범주에는 등록정보를 지정할 수 없습니다.
공유 정보	기본/공유 노드에 대한 정보와 관련 공유 노드 목록을 제공하고 기본 노드가 누락되었는지 여부를 식별합니다. 이 범주는 시스템 환경설정을 통해 공유 노드를 활성화한 경우에만 표시됩니다. <b>참고:</b> 이 범주의 등록정보는 모두 읽기 전용입니다.
통계	하위 멤버 수, 동위 멤버 수 등 노드에 대한 통계 정보를 제공하는 등록정보입니다. <b>참고:</b> 이 범주의 등록정보는 모두 읽기 전용입니다.
검증	노드에 대해 지정된 검증이며 각 검증에 대해 하나의 등록정보가 있습니다.
리프 액세스	노드에 대한 노드 보안 그룹 및 해당 리프 액세스 레벨이며 각 그룹에 대해 하나의 등록정보가 있습니다.

표 8-1 (계속) 등록정보 범주

범주	설명
림 액세스	노드에 대한 노드 보안 그룹 및 해당 림 액세스 레벨이며 각 그룹에 대해 하나의 등록정보가 있습니다.

 주:

사용자 액세스가 특정 범주로 제한될 수 있고 노드 유형을 필터링할 수 있기 때문에 모든 등록정보가 모든 사용자에게 표시되는 것은 아닙니다. 검증, 리프 액세스 및 림 액세스 범주는 데이터 관리자 역할이 지정된 사용자만 사용할 수 있고, 검증 또는 노드 액세스 그룹 보안을 지정할 때만 액세스할 수 있습니다.

## 등록정보 범주 생성

등록정보 범주를 생성하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **새로 작성**에서 **등록정보 범주**를 선택합니다.
3. 등록정보 범주의 이름과 설명을 입력합니다.
4. 공통 사용자 프로비저닝을 사용하는 경우 **외부 그룹 - 편집** 및 **외부 그룹 - 읽기**에서 Oracle Hyperion Shared Services의 Oracle Data Relationship Management 애플리케이션에 프로비저닝된 사용자 그룹을 선택합니다. Shared Services에서 동기화가 진행되는 경우 이러한 외부 그룹의 사용자에게 지정된 액세스 레벨(편집 또는 읽기)의 등록정보 범주 멤버십이 지정됩니다.
5. **등록정보** 탭의 **사용가능** 목록에서 등록정보 범주에 지정할 등록정보를 선택하고 화살표를 사용하여 등록정보를 **선택됨** 목록으로 이동합니다.

 주:

**Ctrl+누르기** 또는 **Shift+누르기**를 사용하면 여러 개의 등록정보를 선택할 수 있습니다. 등록정보를 두 번 눌러 선택하거나 선택 취소합니다.

6. 화살표를 사용하여 선택한 등록정보를 다시 정렬하거나  을 눌러 선택한 등록정보를 사전순으로 표시합니다.
7. **사용자** 탭의 **사용가능** 목록에서 등록정보 범주에 지정할 사용자를 선택하고 화살표를 사용하여 사용자를 **선택됨** 목록으로 이동합니다.
8. [선택됨] 목록에서 사용자 행을 선택하고 **작업** 열의  을 누릅니다.
9. **액세스** 열에서 [읽기] 또는 [편집]을 선택하여 등록정보 범주에 대한 액세스 레벨을 사용자에게 지정합니다.
10. **작업** 열의  을 눌러 변경사항을 저장하거나  을 눌러 변경사항을 삭제합니다.

11. 을 누릅니다.

## 등록정보 범주 편집

등록정보 범주를 편집하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. 등록정보 범주를 선택하고 을 누릅니다.
3. 등록정보 탭의 **사용가능** 목록에서 등록정보 범주에 지정할 등록정보를 선택하고 화살표를 사용하여 등록정보를 **선택됨** 목록으로 이동합니다.

 주:

**Ctrl+누르기** 또는 **Shift+누르기**를 사용하면 여러 개의 등록정보를 선택할 수 있습니다. 등록정보를 두 번 눌러 선택하거나 선택 취소합니다.

4. 화살표를 사용하여 선택한 등록정보를 다시 정렬하거나 을 눌러 선택한 등록정보를 사전순으로 표시합니다.
5. 사용자 탭의 **사용가능** 목록에서 등록정보 범주에 지정할 사용자를 선택하고 화살표를 사용하여 사용자를 **선택됨** 목록으로 이동합니다.
6. [선택됨] 목록에서 사용자 행을 선택하고 **작업** 열의 을 누릅니다.
7. 액세스 열에서 [읽기] 또는 [편집]을 선택하여 등록정보 범주에 대한 액세스 레벨을 사용자에게 지정합니다.
8. 작업 열의 을 눌러 변경사항을 저장하거나 을 눌러 변경사항을 삭제합니다.
9. 을 누릅니다.

## 등록정보 범주 삭제

등록정보 범주를 삭제하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **메타데이터**에서 **등록정보 범주**를 확장합니다.
3. 등록정보 범주를 선택하고 을 누릅니다.
4. **이 항목 삭제**를 선택하여 삭제를 확인합니다.

 주:

등록정보 범주를 삭제해도 범주와 연결된 등록정보가 삭제되지는 않습니다. 애플리케이션 내에서 이 등록정보를 계속 사용할 수 있습니다.

# 9

## 등록정보 정의 관리

등록정보 정의는 Oracle Data Relationship Management에서 버전, 계층 구조 및 노드의 속성을 관리하는 데 사용됩니다. 등록정보는 텍스트, 숫자, 날짜, 다른 데이터 객체에 대한 참조 등 다양한 데이터 유형을 저장할 수 있습니다. 등록정보는 명시적 값을 저장하거나, 상속을 사용하여 하위 멤버 노드에 자동으로 값을 지정하거나, 공식 또는 룩업 테이블을 기준으로 계산될 수 있습니다. 사용을 간소화하고 사용자 액세스를 제어하기 위해 등록정보 범주를 사용하여 등록정보를 관련 세트로 그룹화 및 구성할 수 있습니다.

기본적으로 사용할 수 있는 시스템 정의 등록정보는 표준 제품 기능에 사용됩니다. 애플리케이션 관리자는 사용자정의 등록정보 정의를 생성하여 비즈니스 또는 시스템 통합 요구사항을 지원하는 데 필요한 추가 속성을 관리할 수 있습니다.

Data Relationship Management의 등록정보 정의는 다양한 소스에서 제공될 수 있습니다. 예를 들어 등록정보는 다음과 같을 수 있습니다.

- Data Relationship Management에서 시스템 정의됨
- 애플리케이션 관리자가 생성한 사용자정의 등록정보
- 다른 Oracle 제품에 사용되는 애플리케이션 템플릿에서 로드됨
- 마이그레이션 유틸리티를 사용하여 다른 Data Relationship Management 애플리케이션 또는 환경에서 로드됨

### 네임스페이스

여러 소스의 등록정보가 유사한 이름을 가지며 데이터 무결성을 위해 분리된 상태로 유지되어야 하는 경우 충돌을 방지하기 위해 등록정보 정의에 네임스페이스를 사용합니다. 등록정보 이름은 네임스페이스 접두어 규칙을 사용하여 차별화됩니다.

**표 9-1** 네임스페이스를 사용한 등록정보 정의 예

필드	예
전체 이름	Custom.AccountType
네임스페이스	사용자정의
이름	AccountType
레이블	AccountType

Data Relationship Management에는 충돌이 발생하지 않도록 하기 위해 네임스페이스에 적용되는 다음 특수 규칙이 있습니다.

- 시스템 정의 등록정보는 "Core" 네임스페이스를 사용합니다.
- 사용자정의 등록정보는 "Custom" 네임스페이스를 사용합니다.
- 다른 네임스페이스는 Data Relationship Management 애플리케이션에서 다른 Oracle 제품에 사용하기 위해 예약되었습니다.

## 데이터 유형

등록정보 데이터 유형은 다음 표에 설명되어 있습니다.

**표 9-2** 등록정보 데이터 유형

등록정보 데이터 유형	설명
연계된 그룹	연계된 노드 그룹입니다. 여러 노드를 가리킵니다. 노드가 [연결된 그룹] 노드 및 노드 간을 다시 가리킵니다. 비유: 협회 <b>참고:</b> 이 데이터 유형은 글로벌 노드 레벨 등록정보에만 사용해야 합니다. <b>주의:</b> 노드를 가져오는 순서에 따라 연계된 노드 등록정보가 버전에 아직 존재하지 않아서 임포트를 통해 로드되는 해당 등록정보가 다른 모든 노드를 올바르게 가리키지 않을 수 있습니다.
연계된 노드	연계된 노드입니다. 다른 단일 노드를 가리킵니다. 가리킨 노드가 [연결된 단일 노드] 노드를 다시 가리킵니다. 비유: 결혼 <b>참고:</b> 이 데이터 유형은 글로벌 노드 레벨 등록정보에만 사용해야 합니다. <b>주의:</b> 노드를 가져오는 순서에 따라 연계된 노드 등록정보가 버전에 아직 존재하지 않아서 임포트를 통해 로드되는 해당 등록정보가 다른 모든 노드를 올바르게 가리키지 않을 수 있습니다.
연계된 노드	연계된 노드 목록입니다. 여러 노드를 가리킵니다. 가리킨 노드가 [연결된 노드]를 다시 가리키지만 노드 간은 가리키지 않습니다. 비유: 친구 <b>참고:</b> 이 데이터 유형은 글로벌 노드 레벨 등록정보에만 사용해야 합니다. <b>주의:</b> 노드를 가져오는 순서에 따라 연계된 노드 등록정보가 버전에 아직 존재하지 않아서 임포트를 통해 로드되는 해당 등록정보가 다른 모든 노드를 올바르게 가리키지 않을 수 있습니다.
Boolean	True 또는 False입니다.
Date	날짜 값은 고정 문화권에서 형식이 지정됩니다. 이를 통해 예측 가능한 응답 및 조치를 수행하여 원하는 경우 결과 형식을 다시 지정할 수 있습니다. <b>주의:</b> 영어(미국) 형식으로 기본값, 최대값 및 최소값을 입력해야 합니다.
날짜/시간	날짜 및 시간 값은 고정 문화권에서 형식이 지정됩니다. 이를 통해 예측 가능한 응답 및 조치를 수행하여 원하는 경우 결과 형식을 다시 지정할 수 있습니다. <b>주의:</b> 영어(미국) 형식으로 기본값, 최대값 및 최소값을 입력해야 합니다.

표 9-2 (계속) 등록정보 데이터 유형

등록정보 데이터 유형	설명
Float	사용자 세션과 연결된 국가별 설정에 따라 부동 소수점 값 형식이 지정됩니다. <b>참고:</b> 기본값이 정의되지 않은 경우 익스포트할 때 0 값이 출력됩니다.
형식 지정 메모	형식 지정 메모 - 모든 형식(공백, 탭, 새 라인 등)을 텍스트에 유지합니다. 하이퍼링크 텍스트도 형식 지정 메모에 포함할 수 있습니다. 하이퍼링크 URL의 형식 지정에 대한 자세한 내용은 하이퍼링크 데이터 유형을 참조하십시오. <b>참고:</b> 등록정보 값에 텍스트와 하이퍼링크를 모두 사용하는 경우 URL이 아닌 텍스트가 제외되지 않습니다.
글로벌 노드	버전의 한 노드를 가리킵니다. 값을 지정하면 등록정보 그리드의 값 필드에 노드 이름만 표시됩니다.
Group	심표로 구분된 항목 목록입니다.
계층	계층을 가리킵니다.
Hierarchy Group	계층 그룹을 가리킵니다. 계층 그룹 등록정보를 사용하면 계층을 보려는 컨텍스트에 따라 계층을 여러 가지 방법으로 그룹화할 수 있습니다. 사용에 따라 동일한 버전 내의 계층 구조를 다양한 방법으로 그룹화할 수 있습니다.
Hyperlink	URL 텍스트에 대해 하이퍼링크 기능을 허용합니다. 여러 개의 URL 입력은 공백 없이 캐리지 리턴-라인 피드(CRLF 또는 0x0D0A)로 구분됩니다. 입력한 URL이 탐색 가능한 하이퍼링크로 표시됩니다. 구문분석된 구분된 URL 또는 형식이 지정된 URL만 표시됩니다. URL은 다음 형식을 따라야 합니다. <code>[url=http_URL]URL_Title[/url]</code> 여기서 <code>http_URL</code> 은 하이퍼링크 텍스트를 지정하고 <code>URL_Title</code> 은 사용자에게 표시되는 텍스트를 지정합니다. 예를 들어 <code>[url=http://support.oracle.com]</code> 오라클 고객지원센터[/url] 태그 예제는 등록정보 그리드에 <a href="http://support.oracle.com">오라클 고객지원센터</a> 으로 렌더링됩니다.
Integer	정수 값입니다. 기본값이 정의되지 않은 경우 익스포트할 때 0 값이 출력됩니다.
리프 노드	계층의 리프 노드를 가리킵니다. 값을 지정하면 등록정보 그리드의 값 필드에 계층 이름과 노드 이름이 표시됩니다.

표 9-2 (계속) 등록정보 데이터 유형

등록정보 데이터 유형	설명
림 노드	계층의 림 노드를 가리킵니다. 값을 지정하면 등록정보 그리드의 값 필드에 계층 이름과 노드 이름이 표시됩니다.
목록 그룹	항목의 확인 목록입니다. 목록에서 여러 항목을 선택할 수 있습니다.
Memo	메모 필드 - 형식 지정이 저장되지 않고 단일 텍스트 라인에 데이터가 병합됩니다. 메모에 하이퍼링크도 포함됩니다. 하이퍼링크 URL의 형식 지정에 대한 자세한 내용은 하이퍼링크 데이터 유형을 참조하십시오. <b>참고:</b> 등록정보 값에 텍스트와 하이퍼링크를 모두 사용하는 경우 URL이 아닌 텍스트가 제외되지 않습니다.
여러 노드	여러 노드를 가리킵니다.
Node	계층의 한 노드를 가리킵니다. 값을 지정하면 등록정보 그리드의 값 필드에 계층 이름과 노드 이름이 표시됩니다.
노드 등록정보	노드의 등록정보를 가리킵니다.
Property	등록정보를 가리킵니다.
범위 목록	값 범위를 정의합니다. 정수 값만 수락합니다.
Sort	정렬에 사용되는 정수 값입니다.
정렬 등록정보	정렬 등록정보를 가리킵니다.
표준 쿼리	표준 쿼리를 가리킵니다.
String	문자열 값
Time	시간 값은 고정 문화권에서 형식이 지정됩니다. 이를 통해 예측 가능한 응답 및 조치를 수행하여 원하는 경우 결과 형식을 다시 지정할 수 있습니다. <b>주의:</b> 영어(미국) 형식으로 기본값, 최대값 및 최소값을 입력해야 합니다.
Version	버전을 가리킵니다.

## 외부 룩업

외부 룩업 등록정보는 선택 가능한 값 목록을 위해 외부 데이터 소스에 액세스하는 등록정보입니다. 외부 데이터 소스는 외부 작업을 사용하여 액세스합니다. 외부 룩업 등록정보 유형은 Oracle 또는 SQL Server 데이터베이스에서 레코드 세트를 반환할 수 있습니다. 외부 룩업 결과를 통해 외부 값 목록에서 등록정보 값으로 사용할 항목을 선택하거나 외부 소스의 데이터를 사용하여 요청 항목 등록정보 값을 계산할 수 있습니다. 등록정보 목록에 대한 외부 룩업은 Data Relationship Management 및 Data Relationship Governance에서 액세스할 수 있습니다.

## 등록정보 생성

등록정보 정의를 생성하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **새로 작성**에서 **등록정보 정의**를 선택합니다.
3. 등록정보의 이름을 입력합니다.

### 주:

등록정보는 Custom 네임스페이스에 지정됩니다. 이름을 입력하면 [전체 이름] 및 [레이블] 필드가 자동으로 채워집니다. 등록정보의 전체 이름은 고유해야 합니다. 등록정보 레이블은 애플리케이션 관리 옆의 모든 기능 등록정보 정의에 대해 표시되는, 사용자에게 친숙한 기술자입니다. 동일한 네임스페이스에만 없으면 여러 등록정보가 동일한 레이블을 사용할 수 있습니다. 등록정보 설명은 등록정보 편집기 맨아래에 표시되는 선택적인 긴 기술자입니다.

4. 등록정보의 매개변수를 정의합니다.

### 주:

아래 매개변수가 모두 표시되지는 않습니다. 선택한 데이터 유형에 따라 표시되는 매개변수가 달라집니다.

- **데이터 유형** - [등록정보 데이터 유형](#)을 참조하십시오.

연결된 그룹, 연결된 단일 노드, 연결된 복수 노드, 글로벌 노드, 리프 노드, 림 노드, 여러 노드 또는 노드 데이터 유형을 선택하여 사용자에게 표시되는 노드 목록을 제한할 수 있습니다. 데이터 유형을 선택하면 **제약 조건** 탭이 표시됩니다.

- **등록정보 레벨** - 다음과 같은 등록정보 정의 레벨입니다.

- **로컬 노드** - 특정 계층의 노드에 대해 등록정보 값이 관리되고 이 레벨에서만 액세스할 수 있습니다.
- **글로벌 노드** - 버전의 노드에 대해 등록정보 값이 관리되지만 로컬 노드 레벨에서도 액세스할 수 있습니다.
- **계층** - 계층에 대해 등록정보 값이 관리되지만 로컬 노드 레벨에서도 액세스할 수 있습니다.
- **버전** - 버전에 대해 등록정보 값이 관리되지만 글로벌 또는 로컬 노드 레벨에서도 액세스할 수 있습니다.

### 주:

글로벌 노드 상속 등록정보를 정의하는 경우 글로벌 등록정보에 대한 제어 계층을 정의해야 합니다. 이렇게 하려면 홈 페이지의 [계층] 탭에서 계층에 제어된 등록정보를 지정합니다.

- **등록정보 유형**
  - 정의됨 - 값이 사용자에게 의해 정의되고 저장됩니다.
  - 록업 - 다른 등록정보와 록업 테이블에 따라 록업됩니다.
  - 파생됨 - Deriver 클래스를 사용하여 계산됩니다.

 **주:**

스크립트 deriver 클래스를 사용한 파생 등록정보는 버전, 계층 및 노드 등록정보에 사용할 수 있습니다. 공식 deriver 클래스는 글로벌 또는 로컬 노드 등록정보에만 사용할 수 있습니다.

- 외부 록업 - 외부 데이터 소스를 사용한 록업입니다.

 **주:**

외부 데이터 소스에서 실시간으로 값을 검색합니다. 여러 개의 값이 반환되는 경우 등록정보에 대한 특정 값을 선택해야 합니다.

- **기본값** - 등록정보의 기본값입니다.
  - **도메인** - 데이터 유형이 노드, 림 노드, 리프 노드, 여러 노드, 연결된 단일 노드, 연결된 복수 노드 또는 연결된 그룹인 등록정보의 경우(모두 값으로 저장된 노드를 나타냄) [도메인] 드롭다운을 사용할 수 있습니다. 드롭다운에는 시스템에서 정의된 모든 도메인이 포함되며, 선택적으로 기존 도메인 중 하나를 선택할 수 있습니다.
  - **열 너비** - 등록정보 유형이 [정의됨]인 경우 고정 너비 열의 너비입니다.
  - **최소값/길이** - 데이터 유형에 따른 등록정보의 값 또는 길이입니다.
  - **최대값/길이** - 데이터 유형에 따른 등록정보의 값 또는 길이입니다.
5. 다음 옵션 중에서 선택합니다.
- **상속됨** - 등록정보를 상속으로 정의합니다.

 **주:**

이 옵션은 AncestorProp 또는 DualAncestorProp 등의 등록정보 Deriver가 사용되고 등록정보가 글로벌인 특수한 경우를 제외하고 파생 등록정보 유형에 영향을 주지 않습니다. 이 경우 등록정보가 실제로 값을 상속하지 않아도 [상속됨] 옵션을 통해 제어 계층을 지정할 수 있습니다.

- **재정의 가능** - 등록정보 그리드에서 등록정보를 재정의할 수 있게 합니다.

 **주:**

이 옵션은 파생 등록정보 유형에 대해서만 활성화됩니다.

- **목록** - 사전 정의된 값 목록에서만 등록정보 값을 선택할 수 있습니다.

 주:

EnforceListProps 시스템 환경설정을 사용하여 목록 등록정보에 대해 저장되는 등록정보 값을 목록의 값으로만 제한할 수 있습니다.

 주:

정의된 등록정보 또는 재정의 가능한 파생 등록정보에 대해 값 목록을 사용할 수 있습니다.

- 숨김 - 등록정보 그리드에서 등록정보를 숨깁니다.
- 인덱싱 - 등록정보에 대한 인덱스를 생성하여 검색, 등록정보 쿼리 및 검증 성능을 향상시킵니다. 이 옵션은 정의된 문자열 데이터 유형 등록정보에서만 사용할 수 있습니다.

 주:

인덱스화된 등록정보는 애플리케이션 서버에서 메모리 사용을 증가시킬 수 있으며, 검색, 쿼리 및 고유성을 확인하는 검증에서 사용될 가능성이 큰 등록정보에만 사용해야 합니다.

## 6. 다음 중 하나를 수행합니다.

- 등록정보를 범주에 지정하려면 **사용가능** 목록에서 범주를 선택하고 **선택됨** 목록으로 이동합니다.
- 목록 옵션과 함께 **정의됨** 등록정보 유형을 선택한 경우 **목록 값** 탭에서 다음을 수행합니다.
  - a. **추가**를 누르고 목록에 값을 입력합니다.
  - b. 행의 [작업] 열에서 **저장**을 누릅니다.

 주:

각 행의 [이동] 또는 [삭제]를 사용하여 목록 값의 순서를 조정하거나 삭제합니다. [편집]을 사용하거나 행을 두 번 눌러 편집하고 [취소]를 사용하여 편집을 취소합니다.

- **록업** 등록정보 유형을 선택한 경우 **록업 테이블** 탭을 선택하고 다음을 수행합니다.
  - a. **추가**를 눌러 목록에 새 키-값 쌍을 입력합니다.
  - b. 행의 [작업] 열에서 **저장**을 누릅니다.

 주:

각 행의 [이동] 또는 [삭제]를 사용하여 목록 값의 순서를 조정하거나 삭제합니다. [편집]을 사용하거나 행을 두 번 눌러 편집하고 [취소]를 사용하여 편집을 취소합니다.

- 계층 제약 조건을 허용하는 데이터 유형을 선택한 경우 **제약 조건** 탭을 선택하고 다음을 수행합니다.
  - a. **계층 그룹 등록정보**에서 등록정보를 선택하고 계층 그룹을 선택합니다.  
노드 선택기에서 사용자는 선택된 계층 그룹에 속한 계층의 노드만 볼 수 있습니다.

 주:

Oracle Data Relationship Management Analytics에서는 기본 코어 등록정보 유형만 지원됩니다.

- b. **선택사항: 서버 등록정보 업데이트에 제약 조건 적용**을 선택하여 웹 클라이언트, 임포트, 작업 스크립트 또는 웹 서비스 API를 통해 등록정보가 업데이트될 때 이 제약 조건을 검증합니다.
- **파생됨** 등록정보 유형을 선택한 경우 **매개변수** 탭을 선택하고 파생 등록정보에 대한 공식 또는 스크립트를 정의합니다.  
공식에 대한 자세한 내용은 **공식 생성**을 참조하십시오. 스크립트에 대한 자세한 내용은 **동적 스크립트 생성**을 참조하십시오.
  - **외부 룩업** 등록정보 유형을 선택한 경우 **외부 룩업** 탭을 선택하고 다음 정보를 입력합니다.
    - **외부 연결** - 데이터베이스 또는 웹 서비스 연결을 선택합니다.
    - **작업** - 수행할 외부 작업을 선택합니다.
    - 각 매개변수에 대해 다음을 구성합니다.
      - \* **매개변수 소스 유형** - [리터럴] 또는 [등록정보]를 선택합니다.
      - \* **소스** - 소스 유형에 대해 **리터럴**을 선택한 경우 [매개변수 소스] 열에 리터럴 값을 입력합니다. 이 외부 룩업 등록정보에 대해 외부 작업을 호출하면 현재 매개변수에 대한 리터럴 값이 전달됩니다. 소스 유형에 대해 **등록정보**를 선택한 경우 외부 작업에 대한 매개변수 값을 제공할 등록정보를 선택합니다. 외부 룩업을 실행하면 현재 노드 또는 요청 항목에서 선택한 등록정보의 매개변수 값이 전달됩니다.
    - **열/등록정보 매핑**에서 선택한 룩업 결과 중 선택한 외부 룩업 등록정보의 값을 제공할 결과 열을 선택합니다. **추가**를 눌러 다른 등록정보에 매핑할 수 있는 열을 더 추가하여 외부 룩업 값을 선택할 때 다른 등록정보 값이 자동으로 업데이트되도록 합니다.  
  
첫 번째 열/등록정보 매핑은 자동으로 정의되며 삭제할 수 없습니다. 이 매핑은 현재 등록정보에 적용됩니다. 열을 선택해야 하며, 기본적으로 작업에서 저장된 첫 번째 열로 설정됩니다. 첫 번째 행의 열 값을 수정할 수 있지만 등록정보 값은 수정할 수 없습니다. 추가 매핑을 위해 열 이름 및 결과 열을 선택하고 편집할 수 있습니다.

7. 을 누릅니다.

## 계층 제약 조건 사용

계층 제약 조건은 노드 데이터 유형 등록정보 값을 업데이트할 때 보고 선택할 수 있는 계층 및 노드가 제한될 수 있습니다. 계층 제약 조건은 노드 데이터 유형을 사용하는 등록정보 정의에 대한 선택적 구성입니다. 계층 제약 조건 기능은 계층 제약 조건을 지정하기 전에 구성되어야 하는 계층 그룹 및 계층 그룹 등록정보를 사용합니다.

계층 제약 조건을 다음 데이터 유형과 함께 사용할 수 있습니다.

- 연계된 그룹
- 연계된 노드
- 연계된 노드
- 글로벌 노드
- 리프 노드
- 림 노드
- 여러 노드
- Node

### 주:

연결된 노드에서 상호 참조를 생성하기 때문에 계층 제약 조건을 설정할 때 연결된 그룹, 연결된 단일 노드, 연결된 복수 노드 등의 노드 데이터 유형을 추가로 고려해야 할 수 있습니다. 계층 제약 조건이 정의되어 있는 경우 계층 그룹에 서로 연결될 수 있는 모든 계층이 포함되도록 주의해야 합니다. 한 가지 예는 사원 및 비용 센터 계층 구조에서 노드 간 상호 참조입니다. 계층 제약 조건에 사용할 별도의 계층 그룹 등록정보 및 계층 그룹을 생성해야 할 수 있습니다.

## 등록정보 정의 편집

등록정보 정의가 [정의됨] 등록정보 유형에서 [파생됨] 또는 [록업]와 같은 편집할 수 없는 유형으로 수정된 경우 다음 조건이 적용됩니다.

- 저장되지 않은 등록정보로 전환 시의 확인 메시지가 변경 요청 항목에 대해 대기 중인 업데이트가 영향을 받을 수 있다고 수정됩니다.
- 진행 중인 요청에 대해 대기 중인 등록정보 업데이트는 해당 태스크가 지정된 항목에 대해 더 이상 표시, 검증 또는 커밋되지 않습니다.

등록정보 정의를 편집하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **메타데이터**에서 **등록정보 정의**를 확장합니다.
3. 등록정보 정의 유형에 따라 **Core** 또는 **Custom**을 확장합니다.
4. 등록정보를 두 번 누릅니다.
5. 편집할 수 있는 매개변수를 수정합니다.

**▲ 주의:**

등록정보 유형을 정의된 값(RWDerived 또는 정의됨)에서 저장을 허용하지 않는 값(파생됨 또는 록업)으로 변경하면 정의된 등록정보 값이 삭제되고 이 데이터가 손실됩니다. 이 유형의 변경 작업을 수행하기 전에 잠재적 데이터 손실이 허용되는지 확인해야 합니다.

자세한 내용은 [등록정보 생성](#)을 참조하십시오.

6. 을 누릅니다.

## 등록정보 삭제

Oracle Data Relationship Management에서 등록정보 정의가 삭제되는 경우 다음 조건이 적용됩니다.

- 등록정보 정의에 대한 종속성 확인이 워크플로우 메타데이터 참조를 포함하도록 수정되며 사용자가 삭제를 확인해야 합니다. 워크플로우 메타데이터에 대한 등록정보 정의 종속성은 다음으로 구성됩니다.
  - 워크플로우 태스크 등록정보
  - 워크플로우 태스크 검증 등록정보
  - 변경 요청 항목 세부정보
- 확인 시 등록정보가 삭제된 경우 워크플로우 태스크에 대한 지정, 진행 중인 요청에 대해 대기 중인 업데이트 및 과거 변경 요청을 포함하여 등록정보에 대한 각 종속 참조도 삭제됩니다.
- 등록정보 정의의 대화식 삭제와 마찬가지로 트랜잭션 내역은 항상 유지됩니다.

등록정보를 삭제하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **메타데이터**에서 **등록정보 정의**를 확장합니다.
3. 등록정보를 선택하고 을 누릅니다.
4. **등록정보 정의 삭제**를 선택하여 삭제를 확인합니다.

**▲ 주의:**

등록정보 정의를 삭제하면 등록정보를 사용 중인 모든 메타데이터 객체에서 등록정보가 제거되는 것은 물론 등록정보에 대해 저장된 모든 값이 삭제됩니다.

# 10

## 검증 관리

검증을 사용하면 버전, 계층 구조, 노드 및 등록정보에 비즈니스 규칙을 적용할 수 있습니다. 실시간, 배치 또는 두 모드로 모두 검증을 실행할 수 있습니다. 실시간 검증은 수정 시 실행되며, 작업이 적용되는 규칙을 위반할 경우 변경사항이 저장되지 않습니다. 배치 검증은 부적합하여 수정해야 하는 데이터 조건을 식별하기 위해 편집 이전이나 이후에 명시적으로 실행할 수 있습니다.

### 검증 클래스

검증 클래스를 사용하면 여러 유형의 비즈니스 규칙을 적용할 수 있습니다. 다른 클래스는 특정 목적에 사용되지만 일부 검증 클래스는 일반적으로 사용할 수 있습니다. 기존 검증 클래스 세트에서 검증을 생성할 수 있습니다. 노드의 많은 비즈니스 규칙은 해당 논리에 대한 쿼리를 사용하는 검증 클래스로 적용할 수 있습니다. 이렇게 하면 검증에서 분석 목적으로 생성된 쿼리를 사용하여 데이터 무결성도 관리할 수 있습니다. 다른 검증 규칙을 사용하여 버전 및 계층 구조에 대한 규칙 또는 노드에 대한 특별한 경우를 처리할 수 있습니다. 몇몇 검증 클래스는 제품 테스트 목적으로만 사용되며 프로덕션 환경에서는 사용하면 안 됩니다.

표 10-1 검증 클래스

검증 클래스	레벨	설명	매개변수
BoolNodeInHier	Node	지정된 계층의 지정된 부울 등록정보에 True 값이 없는지 확인합니다.	등록정보, 계층
ContainAllProp	글로벌 노드	지정된 계층에 지정된 등록정보가 True인 모든 노드가 포함되어 있는지 확인합니다.	계층, 등록정보
ContainAllWith	글로벌 노드	지정된 계층에 지정된 등록정보가 지정된 값을 가진 모든 노드가 포함되어 있는지 확인합니다.	계층, 등록정보, 값
CustPropQuery	Node	사전 정의된 쿼리와 예상 결과를 사용하여 확인합니다. 로컬 등록정보 쿼리만 사용할 수 있습니다.	등록정보 쿼리 이름, 실패 값
DateRangeCheck	Node	시작 날짜가 종료 날짜보다 이전이거나 같은지 확인합니다.	시작 날짜 등록정보, 종료 날짜 등록정보
공식	Node	비즈니스 논리를 사용한 노드가 공식에 표현되었는지 확인합니다. 공식 결과가 False이면 검증에 실패합니다.	공식
GlobalPropQuery	글로벌 노드	사전 정의된 쿼리와 예상 결과를 사용하여 확인합니다.	등록정보 쿼리 이름, 실패 값

표 10-1 (계속) 검증 클래스

검증 클래스	레벨	설명	매개변수
HierContainsRef	Node	부울 등록정보가 True이거나 노드가 리프 노드이고 세 번째 부울 등록정보가 True인 경우 계층에 노드에 대한 참조가 포함됩니다.	계층 이름, 모든 노드에 대한 부울 등록정보, 리프 노드에 대한 부울 등록정보
HierFail	계층	테스트 목적으로 계층 레벨에서 자동으로 실패합니다.	없음
InvalidNameLength	Node	노드 이름이 지정된 길이와 같지 않은지 확인합니다.	Length
MaxChildren	Version	노드당 하위 멤버 수가 지정된 제한을 초과하지 않는지 확인합니다.	최대 하위 멤버 수
MaxHierNodes	계층	계층의 노드 수가 지정된 제한을 초과하지 않는지 확인합니다.	최대 노드 수
MaxVersionNodes	Version	버전의 노드 수가 지정된 제한을 초과하지 않는지 확인합니다.	최대 노드 수
MergeEquiv	병합	영향을 받는 노드와 병합 노드에서 지정된 등록정보의 값이 같은지 확인합니다.	글로벌 노드 등록정보
MergePropSet	병합	영향을 받는 노드 등록정보 값이 설정된(재정의됨) 경우 지정된 등록정보에 대해 병합 노드 등록정보 값이 설정되었는지 확인합니다 (등록정보 값이 동일할 필요 없음).	Property
MixedKids	Node	림 및 리프 하위 멤버가 모두 포함된 노드를 확인합니다.	없음
NoBoolBranch	Node	지정된 분기에서 지정된 부울 등록정보가 한 번 이상 True로 설정되었는지 확인합니다.	Property
NodeFail	글로벌 노드	테스트 목적으로 버전 레벨의 노드에서 자동으로 실패합니다.	없음
NodeFailRandom	Node	테스트 목적으로 지정된 노드의 백분율에서 자동으로 실패합니다.	실패 백분율
NoDefaults	Node	지정된 등록정보에 기본값이 사용되지 않는지 확인합니다.	Property

표 10-1 (계속) 검증 클래스

검증 클래스	레벨	설명	매개변수
NoPropBranch	Node	지정된 분기에서 지정된 등록정보가 한 번 이상 설정되었는지 확인합니다.	Property
PropEquivBool	Node	세 번째 부울 등록정보가 True인 경우 등록정보가 동일합니다.	평가할 부울 등록정보, 첫 번째 등록정보, 두 번째 등록정보
PropLength	Node	지정된 등록정보가 최소 길이 이상이고 최대 길이를 넘지 않는지 확인합니다.	등록정보, 최소 길이, 최대 길이
PropRemove	제거	지정된 등록정보(prop1, prop2 및 prop3 매개변수)가 지정된 값(value1, value2, value3 매개변수)과 같은 경우 노드가 제거되지 않도록 합니다.	Property1, Property2, Property3, Value1, Value2, Value3
RequiredField	Node	지정된 등록정보가 지정된 값을 가진 모든 노드에 대해 필수 목록의 각 등록정보에 값이 있는지 확인합니다. <ul style="list-style-type: none"> <li>[기본 레코드 거부] 플래그가 True인 경우 필수 목록의 각 등록정보에 기본값이 아닌 값이 있어야 합니다.</li> <li>[기본 레코드 거부] 플래그가 False인 경우 기본값이 허용됩니다.</li> </ul>	등록정보, 값, 기본 레코드 거부, 필수 등록정보
스크립트	노드, 계층, 버전, 글로벌 노드, 이동, 제거, 병합	동적 스크립트를 사용하여 데이터를 확인합니다. 반환 값이 True이면 검증을 통과합니다. 반환 값이 False이면 검증에 실패합니다.	스크립트
SingleBoolBranch	Node	지정된 부울 등록정보가 분기당 한 번만 True로 설정되었는지 확인합니다.	Property
SinglePropBranch	Node	지정된 등록정보가 분기당 한 번만 설정되었는지 확인합니다.	Property
StrandedParent	Node	모든 림 노드에 하위 멤버가 있는지 확인합니다.	없음
StrPropEqual	Node	지정된 등록정보가 지정된 값과 같은 모든 노드에 대해 실패합니다.	등록정보, 값

표 10-1 (계속) 검증 클래스

검증 클래스	레벨	설명	매개변수
UniqueProp	Node	계층 내에서 지정된 등록정보에 중복된 값이 없는지 확인합니다. 기본값 포함이 False인 경우 기본값이 있는 노드는 포함되지 않습니다. 공유 제외가 True인 경우 등록정보 값이 고유한지 확인할 때 공유 노드는 고려되지 않습니다.	등록정보, 기본값 포함, 공유 제외 UniqueProp 검증에서 인덱싱된 등록정보를 사용하는 것이 좋습니다.
UniquePropBranch	Node	분기 내에서 지정된 등록정보에 고유한 값이 있는지 확인합니다.	Property
VersionFail	Version	테스트 목적으로 버전 레벨에서 자동으로 실패합니다.	없음
VersionUnique2Prop	글로벌 노드	버전 내에서 지정된 등록정보에 중복된 값이 없는지 확인합니다. 기본값 포함이 False인 경우 기본값이 있는 노드는 포함되지 않습니다. 공유 제외가 True인 경우 등록정보 값이 고유한지 확인할 때 공유 노드는 고려되지 않습니다.	첫 번째 등록정보, 두 번째 등록정보, 기본값 포함, 공유 제외
VersionUniqueProp	글로벌 노드	버전 내에서 지정된 등록정보에 중복된 값이 없는지 확인합니다. 기본값 포함이 False인 경우 기본값이 있는 노드는 포함되지 않습니다. 공유 제외가 True인 경우 등록정보 값이 고유한지 확인할 때 공유 노드는 고려되지 않습니다.	등록정보, 기본값 포함, 공유 제외

## 검증 레벨

검증 레벨은 비즈니스 규칙의 범위를 정의합니다. 노드 검증의 경우 검증을 실행하기 위해 수행해야 하는 작업 유형도 레벨에 포함될 수 있습니다. 다음 표에서는 각 검증 레벨을 정의하고 다음을 표시합니다.

- 배치 모드, 실시간 모드 또는 둘 다에서 검증을 실행할 수 있는지 여부
- 검증이 지정되는 위치
- 검증이 수행되는 객체

표 10-2 검증 레벨

검증 레벨	배치 또는 실시간으로 실행	지정되는 위치	수행 대상
<p><b>노드</b> - 노드 관계 및 등록정보를 검토하여 기준이 충족되었는지 확인합니다.</p> <p>노드 레벨 문자열 등록정보 값이 적합한 길이인지 여부를 확인하는 데 사용합니다.</p>	실시간 또는 배치	버전, 계층 또는 노드	로컬 노드
<p><b>계층</b> - 계층의 등록정보를 검토하여 기준이 충족되었는지 확인합니다.</p> <p>계층 또는 버전 레벨에서 지정하고 실행할 수 있습니다.</p> <p>계층에 포함된 노드 수가 10,000개를 넘지 않도록 하는 데 사용합니다.</p>	배치	버전 또는 계층	계층
<p><b>버전</b> - 버전의 등록정보를 검토합니다.</p> <p>버전에 포함된 노드 수가 100,000개를 넘지 않도록 하는 데 사용합니다.</p>	배치	Version	Version
<p><b>글로벌 노드</b> - 버전 레벨에서 지정됩니다. 계층에 관계없이 단독을 비롯한 버전의 모든 노드를 검증합니다. 글로벌로 정의된 등록정보만 검토됩니다.</p> <p>버전 내의 모든 노드에 고유한 등록정보 값이 있도록 하는 데 사용합니다.</p>	배치	Version	글로벌 노드
<p><b>병합</b> - 병합이 필요한 작업 (예: 삭제 또는 비활성화)을 수행할 때 실행됩니다. 버전 레벨에서 지정됩니다.</p> <p>리프 노드가 다른 리프 노드에만 병합되도록 하는 데 사용합니다.</p>	실시간	Version	글로벌 노드
<p><b>이동</b> - 노드를 이동하려고 할 때 트리거되는 검증입니다. 계층 레벨에서 지정됩니다.</p> <p>계층 내에서 비용 센터를 이동할 수 없도록 하는 데 사용합니다.</p>	실시간	계층	로컬 노드

표 10-2 (계속) 검증 레벨

검증 레벨	배치 또는 실시간으로 실행	지정되는 위치	수행 대상
제거 - 이동 레벨과 비슷합니다. 계층에서 노드를 제거하거나 삭제하려고 할 때 실행됩니다. 지정된 노드 유형이 삭제되지 않도록 하는데 사용할 수 있습니다. 계층에서 비용 센터 노드를 삭제할 수 없도록 하는데 사용합니다.	실시간	버전 또는 계층	글로벌 노드

## 검증 생성

검증을 생성하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **새로 작성**에서 **검증**을 선택합니다.
3. 검증의 이름을 입력합니다.

### 주:

검증은 Custom 네임스페이스에 지정됩니다. 검증의 전체 이름은 고유해야 합니다. 이름을 입력하면 [레이블] 필드가 자동으로 채워집니다. 검증 레이블은 애플리케이션 관리 옆의 모든 기능 검증에 대해 표시되는, 사용자에게 친숙한 기술자입니다. 동일한 네임스페이스에만 없으면 여러 검증이 동일한 레이블을 사용할 수 있습니다.

4. 검증에 실패할 경우 사용자에게 표시할 메시지를 입력합니다.
5. 검증 클래스를 선택합니다. **검증 클래스**를 참조하십시오.

### 주:

선택한 클래스에 따라 적합한 레벨이 채워집니다.

6. 노드 레벨에서 실시간으로 실행할 수 있는 클래스의 경우 작업 유형이 포함된 레벨을 선택합니다.
7. 검증에 대한 다음 옵션 중에서 선택합니다.
  - 실시간 - 변경할 때 실행됩니다.
  - 배치 - 명시적으로 요청할 때 실행됩니다.
  - 상속됨 - 선택한 노드 및 하위 멤버에 대해 실행됩니다.

 주:

선택한 검증 클래스에 따라 이러한 옵션 중 일부를 사용할 수 없거나, 값을 편집해야 할 수 있는 옵션의 매개변수가 표시됩니다.

8. 선택한 검증 클래스의 매개변수를 정의합니다.

각 검증 클래스에 대한 매개변수는 [검증 클래스](#)를 참조하십시오. 공식 생성에 대한 자세한 내용은 [공식 생성](#)을 참조하십시오. 스크립트 생성에 대한 자세한 내용은 [동적 스크립트 생성](#)을 참조하십시오.

9. 을 누릅니다.

## 이동 관련 스크립트 검증 생성

이동 관련 스크립트 검증을 생성하려면 다음을 수행하십시오.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **새로 작성**에서 **검증**을 선택합니다.
3. **클래스**에서 **스크립트**를 선택합니다.

기본적으로 검증 레벨은 [노드]이고 실행 모드는 [뱃치]입니다.

4. **이 검증 실행**에서 **실시간**을 선택합니다.

이를 통해 특정 작업(예: 이동)에서 검증을 트리거할 수 있습니다.

5. **레벨**에서 **이동**을 선택합니다.

 주:

[레벨] 옵션은 4단계에서 선택한 [실시간] 옵션 위에 있습니다.

6. 검증을 저장합니다.

## 검증 지정

검증을 생성한 후 버전, 계층 구조, 도메인 및 노드에 지정할 수 있습니다. 동시에 여러 검증을 지정할 수 있습니다.

 주:

도메인 레벨에서 검증을 지정하면 해당 도메인의 멤버인 모든 노드가 검증을 상속합니다. 버전 레벨에서 검증을 지정하면 해당 버전 내의 모든 계층 구조와 노드가 검증을 상속합니다. 계층 레벨에서 검증을 지정하면 계층 내의 모든 노드가 검증을 상속합니다.

도메인에 검증을 할당하는 방법에 대한 자세한 내용은 [도메인 관리](#)를 참조하십시오. 버전, 계층 구조 및 노드에 검증을 할당하는 방법에 대한 자세한 내용은 *Oracle Data Relationship Management 사용자 가이드*를 참조하십시오.

## 검증 편집

검증을 편집하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **메타데이터**에서 **검증**을 확장합니다.
3. 검증을 선택하고 을 누릅니다.
4. 검증을 변경합니다.

 **주:**

검증을 저장한 후에는 Class, Level 및 Mode of Operation 매개변수를 수정할 수 없습니다.

5. **저장**을 누릅니다.

## 검증 삭제

검증을 삭제하면 버전, 계층 구조 및 노드에 대한 검증 지정도 모두 삭제됩니다.

검증을 삭제하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **메타데이터**에서 **검증**을 확장합니다.
3. 검증을 선택하고 을 누릅니다.
4. **이 항목 삭제**를 선택하여 삭제를 확인합니다.

# 11

## 공식 관리

공식을 사용하면 Oracle Data Relationship Management의 기본 공식 언어로 파생 등록정보 및 검증에 대한 복합 논리를 정의할 수 있습니다. 공식은 함수 및 문자열 리터럴로 구성되며 특정 구문 규칙을 따라야 합니다.

자세한 내용은 다음을 참조하십시오.

- [등록정보 생성](#)
- [검증 관리](#)

## 함수 작업

함수 이름은 대소문자를 구분하지 않으며, 매개변수가 필수인지 여부에 관계없이 바로 뒤에 괄호가 와야 합니다.

함수 매개변수는 예상 유형과 개수여야 합니다. 매개변수는 중첩 함수 또는 문자열 리터럴일 수 있습니다. 매개변수가 부정확한 유형인 경우 오류가 보고됩니다. 매개변수가 너무 적은 경우 목록 인덱스가 범위를 벗어남 오류가 보고됩니다. 매개변수가 너무 많은 경우 추가 매개변수가 무시됩니다.

## 특수 문자

매개변수 값에 특수 문자(예: 심표, 공백, 탭)가 포함된 특정 함수에서는 대괄호(`[]`)를 사용합니다. 예를 들어 `FlipList(PropValue(Custom.NodeList), [comma])`는 `PropValue(Custom.NodeList)` 함수 호출에서 반환된 심표로 구분된 목록에 대해 `FlipList` 함수를 수행합니다.

다음 함수는 `Delimiter` 매개변수에 대해 대괄호(`[]`)로 묶인 `comma`, `space` 또는 `tab`을 사용할 수 있습니다. `ArrayCount`, `ArrayIndex`, `ArrayItem`, `FlipList`, `Intersection`, `ListContains`, `PadList`, `RangeListContains`, `IsRangeListSubset`, `MinList`, `MaxList`, `AvgList`, `SumList`, `SortList`, `ListDistinct`, `ListNodePropValues` 및 `ListNodesWith`.

이전 패턴과 새 패턴의 매개변수가 필요한 `ReplaceStr` 함수는 일반 텍스트 문자열과 함께 대괄호(`[]`)로 묶인 `comma`, `space`, `tab`, `crlf`, `cr`, `lf`, `openparen` 또는 `closeparen`을 사용할 수 있습니다.

### 주:

리터럴 심표가 포함된 매개변수 값을 지정하면 "부적합한 매개변수 수입니다." 구문 오류가 발생합니다. 함수 호출의 결과로 전달되는 심표로 구분된 목록은 적합한 사용이며 예상대로 처리됩니다. 예를 들면 다음과 같습니다.

부적합한 구문: `FlipList(a,b,c, [comma])`

적합한 구문: `FlipList(PropValue(Custom.NodeList), [comma])` (여기서 `Custom.NodeList` 값 = `a,b,c`임)

## 리터럴

뒤에 괄호가 오는 적합한 함수 이름이 아닌 값은 리터럴로 간주됩니다. 리터럴은 문자열, 정수, 부동 소수점 또는 부울 리터럴일 수 있습니다. 문자열 리터럴에서 공백은 문자로 간주됩니다. 따라서 적절한 결과를 파생시키는 데 필요 없는 경우 공식에 추가 공백을 사용하지 마십시오. 저장하기 전에 [공백 제거] 옵션을 사용하여 공식에서 공백을 제거할 수 있습니다.

## 형식 문자열 매개변수

문자열 형식 지정 루틴에 전달되는 형식 문자열에는 리터럴 문자와 형식 지정자 등 두 가지 유형의 객체가 포함됩니다. 리터럴 문자는 결과 문자열에 복사됩니다. 형식 지정자는 지정된 등록정보에서 등록정보 값을 가져온 다음 형식 지정을 적용합니다. 형식 문자열에는 하나의 지정자만 있습니다.

형식 지정자는 다음과 같은 형식을 사용합니다.

```
"%"["-"][width]["."prec]type
```

**표 11-1** 형식 문자열 문자

문자	설명
%	형식 지정자의 시작을 나타냅니다.
["—"]	왼쪽 맞춤 표시기입니다(선택 사항). 값 뒤에 공백을 추가하여 결과를 왼쪽에 맞춥니다. 기본값은 값 앞에 공백을 추가하여 결과를 오른쪽에 맞추는 것입니다.
[width]	너비 지정자입니다(선택 사항). 변환의 최소 필드 너비를 설정합니다. 결과 문자열이 최소 필드 너비보다 짧으면 필드 너비를 늘리기 위해 공백이 채워집니다.
["." prec]	정밀도 지정자입니다(선택 사항).
type	변환 유형 문자입니다. 변환 문자는 대문자 또는 소문자로 지정할 수 있습니다. 모든 부동 소수점 형식의 경우 소수 구분자와 천단위 구분자로 사용되는 실제 문자는 <code>DecimalSeparator</code> 및 <code>ThousandSeparator</code> 글로벌 변수 또는 동등한 <code>TFormatSettings</code> 항목에서 가져옵니다. type에 적합한 값은 다음 표에 나와 있습니다.

표 11-2 형식 문자열 유형 값

유형 값	설명
d	10진수 등록정보 값이 정수여야 합니다. 값이 10진수 문자열로 변환됩니다. 형식 문자열에 정밀도 지정자가 포함된 경우 결과 문자열에 지정된 자릿수 이상이 포함되어야 함을 나타냅니다. 값의 자릿수가 더 적으면 결과 문자열의 왼쪽에 0이 채워집니다.
u	부호 없는 10진수 d와 유사하지만 부호가 출력되지 않습니다.
e	지수 등록정보 값이 부동 소수점 값이어야 합니다. 값이 "-d.ddd...E+ddd" 형태의 문자열로 변환됩니다. 숫자가 음수인 경우 결과 문자열이 빼기 기호로 시작됩니다. 소수점 앞에 항상 한 자릿수가 옵니다. 결과 문자열의 총 자릿수(소수점 앞의 한 자릿수 포함)는 형식 문자열의 정밀도 지정자에 의해 지정됩니다. 정밀도 지정자가 없으면 기본 정밀도 15가 사용됩니다. 결과 문자열의 "E" 지수 문자 뒤에는 항상 더하기 또는 빼기 기호와 세 자릿수 이상이 옵니다.
f	고정 등록정보 값이 부동 소수점 값이어야 합니다. 값이 "-ddd.ddd..." 형태의 문자열로 변환됩니다. 숫자가 음수인 경우 결과 문자열이 빼기 기호로 시작됩니다. 소수점 뒤의 자릿수는 형식 문자열의 정밀도 지정자에 의해 지정됩니다. 정밀도 지정자가 없으면 기본값인 두 자릿수가 사용됩니다.
g	일반 등록정보 값이 부동 소수점 값이어야 합니다. 값이 고정 또는 과학적 형식을 사용하여 가능한 가장 짧은 10진수 문자열로 변환됩니다. 결과 문자열의 유효 자릿수는 형식 문자열의 정밀도 지정자에 의해 지정됩니다. 정밀도 지정자가 없으면 기본 정밀도 15수가 사용됩니다. 결과 문자열에서 후행 0이 제거되고 필요한 경우에만 소수점이 나타납니다. 값의 소수점 왼쪽 자릿수가 지정된 정밀도보다 작거나 같은 경우 및 값이 0.00001보다 크거나 같은 경우 결과 문자열이 고정 소수점 형식을 사용합니다. 그렇지 않으면 결과 문자열이 과학적 형식을 사용합니다.
n	Number 등록정보 값이 부동 소수점 값이어야 합니다. 값이 "-d.ddd,ddd.ddd..." 형태의 문자열로 변환됩니다. "n" 형식은 결과 문자열에 천단위 구분자가 포함된다는 점을 제외하고 "f" 형식과 같습니다.

표 11-2 (계속) 형식 문자열 유형 값

유형 값	설명
m	화폐 등록정보 값이 부동 소수점 값이어야 합니다. 값이 통화 금액을 나타내는 문자열로 변환됩니다. 변환은 CurrencyString, CurrencyFormat, NegCurrFormat, ThousandSeparator, DecimalSeparator 및 CurrencyDecimals 글로벌 변수 또는 TFormatSettings 데이터 구조의 동등한 항목에 의해 제어됩니다. 형식 문자열에 정밀도 지정자가 포함된 경우 CurrencyDecimals 글로벌 변수 또는 동등한 TFormatSettings 항목에 의해 지정된 값을 재정의합니다.
s	String 등록정보 값이 문자, 문자열 또는 PChar 값이어야 합니다. 형식 지정자 대신 문자열 또는 문자가 삽입됩니다. 형식 문자열에 있는 경우 정밀도 지정자는 결과 문자열의 최대 길이를 지정합니다. 등록정보 값이 이 최대값보다 긴 문자열이면 문자열이 잘립니다.
x	16진수 등록정보 값이 정수 값이어야 합니다. 값이 16진수 문자열로 변환됩니다. 형식 문자열에 정밀도 지정자가 포함된 경우 결과 문자열에 지정된 자릿수 이상이 포함되어야 함을 나타냅니다. 값의 자릿수가 더 적으면 결과 문자열의 왼쪽에 0이 채워집니다.

## 날짜-시간 형식 문자열

날짜-시간 형식 문자열은 문자열로 변환될 경우 날짜-시간 값(예: TDateTime)의 형식을 지정합니다. 날짜-시간 형식 문자열은 형식 문자열에 삽입될 값을 나타내는 지정자에서 구성됩니다. 일부 지정자(예: "d")는 숫자 또는 문자열의 형식을 지정합니다. 다른 지정자(예: "f")는 글로벌 변수의 로케일별 문자열을 나타냅니다. "am/pm" 및 "a/p" 지정자를 제외한 지정자의 대소문자는 형식에서 무시됩니다.

지정자	표시
c	날짜 뒤에 시간이 옵니다. <b>참고:</b> 날짜-시간 값이 자정 정각을 나타내는 경우에는 시간이 표시되지 않습니다.
d	선행 0 없이 일이 숫자로 표시됩니다(1-31).
dd	선행 0을 포함하여 일이 숫자로 표시됩니다(01-31).
ddd	일이 약어로 표시됩니다(Sun-Sat).
dddd	일이 전체 이름으로 표시됩니다(일요일-토요일).
dddddd	짧은 날짜 형식입니다.
ddddddd	긴 날짜 형식입니다.

지정자	표시
e	선행 0 없이 현재 기간/시기의 연도가 숫자로 표시됩니다(일본어, 한국어 및 대만어 로케일에만 해당).
ee	선행 0을 포함하여 현재 기간/시기의 연도가 숫자로 표시됩니다(일본어, 한국어 및 대만어 로케일에만 해당).
g	기간/시기가 약어로 표시됩니다(일본어 및 대만어 로케일에만 해당).
gg	기간/시기가 전체 이름으로 표시됩니다(일본어 및 대만어 로케일에만 해당).
m	선행 0 없이 월이 숫자로 표시됩니다(1-12). <b>주의:</b> "m" 지정자가 "h" 또는 "hh" 지정자 바로 뒤에 오는 경우에는 월 대신 분이 표시됩니다.
mm	선행 0을 포함하여 월이 숫자로 표시됩니다(01-12). <b>주의:</b> "m" 지정자가 "h" 또는 "hh" 지정자 바로 뒤에 오는 경우에는 월 대신 분이 표시됩니다.
mmm	월이 약어로 표시됩니다(Jan-Dec).
mmmm	월이 전체 이름으로 표시됩니다(1월-12월).
yy	연도가 두 자릿수로 표시됩니다(00-99).
yyyy	연도가 네 자릿수로 표시됩니다(0000-9999).
h	선행 0이 없는 시간입니다(0-23).
hh	선행 0을 포함하는 시간입니다(00-23).
n	선행 0이 없는 분입니다(0-59).
nn	선행 0을 포함하는 분입니다(00-59).
s	선행 0이 없는 초입니다(0-59).
ss	선행 0을 포함하는 초입니다(00-59).
z	선행 0이 없는 밀리초입니다(0-999).
zzz	선행 0을 포함하는 밀리초입니다(000-999).
t	ShortTimeFormat 글로벌 변수로 지정된 형식을 사용한 시간입니다.
tt	LongTimeFormat 글로벌 변수로 지정된 형식을 사용한 시간입니다.
am/pm	선행 "h" 또는 "hh" 지정자에 대해 12시간제를 사용하고 정오 이전의 시간에 대해 "am", 정오 이후의 시간에 대해 "pm"을 표시합니다. am/pm 지정자는 소문자, 대문자 또는 혼합된 대소문자를 사용할 수 있으며 그에 따라 결과가 표시됩니다.
a/p	선행 "h" 또는 "hh" 지정자에 대해 12시간제를 사용하고 정오 이전의 시간에 대해 "a", 정오 이후의 시간에 대해 "p"를 표시합니다. a/p 지정자는 소문자, 대문자 또는 혼합된 대소문자를 사용할 수 있으며 그에 따라 결과가 표시됩니다.

지정자	표시
ampm	선행 "h" 또는 "hh" 지정자에 대해 12시간제를 사용합니다.
/	국가별 설정에 의해 지정된 날짜 구분 문자입니다.
:	국가별 설정에 의해 지정된 시간 구분 문자입니다.
'xx'/'xx'	작은따옴표 또는 큰따옴표로 묶인 문자가 있는 그대로 표시되고 형식에 영향을 주지 않습니다.

## 공식 평가

등록정보 정의 또는 검증을 생성하거나 수정할 때 공식을 테스트할 수 있습니다. 제공된 등록정보 값으로 공식을 평가하여 공식의 결과를 계산합니다. 이 프로세스는 간단한 구문 검증에서 놓쳤을 수 있는 공식의 논리 또는 구현 오류를 찾을 수 있습니다. 공식 결과와 모든 공식 오류 또는 상태 메시지가 표시됩니다.

공식은 왼쪽에서 오른쪽으로 평가되며 함수 및 문자열 리터럴이 발견될 때 해당 평가가 수행됩니다. 이 방법을 사용할 경우 중첩 함수 오른쪽에 표시되는 추가 매개변수보다 중첩 함수가 먼저 평가됩니다. 공식에서 명시적으로 함수를 중첩하거나, 다른 공식 등록정보의 값을 검색하여 암시적으로 함수를 중첩할 수 있습니다. 대부분의 경우 순환 참조(명시적 또는 암시적으로 등록정보 자체를 참조하는 등록정보 공식)를 피해야 합니다. Oracle Data Relationship Management는 유해한 순환 참조를 검색하고 차단하지만, 반드시 필요하고 잘 아는 경우가 아니면 순환 참조를 사용하지 않도록 해야 합니다.

## 공식 구문 확인

공식을 저장하기 전에 공식 구문에서 다음을 확인합니다.

- 함수 이름이 정확합니다.
- 등록정보 이름이 정확합니다.
- 동일한 개수의 여는 괄호와 닫는 괄호가 있습니다.
- 실제 매개변수 개수가 각 함수의 예상 매개변수 개수 이상입니다.

Concat 등의 함수는 개수 제한 없이 매개변수를 사용할 수 있습니다. 매개변수 개수 검증은 실제 매개변수 개수가 예상 매개변수 개수보다 많거나 같은지 확인합니다. 따라서 매개변수 개수가 너무 많다고 오류가 생성되지는 않지만 너무 적으면 오류가 생성됩니다.

구문 검증은 공식을 평가하지 않으므로 부적합한 상수를 입력하면 오류가 발생할 수 있습니다. 예를 들어 `IntToStr(ABC, 3)`은 구문 검증을 통과하지만 Oracle Data Relationship Management 애플리케이션에서 오류를 생성합니다. 이 유형의 오류를 방지하려면 저장하기 전에 각 공식을 평가해야 합니다.

## 구문 확인의 등록정보 이름

등록정보 이름에 대해 구문 검증을 정확하게 수행하기 위해 등록정보 이름이 리터럴이지만 함수의 결과인 드문 경우에 대해 등록정보 이름이 필요한 함수를 부분적으로 평가합니다.

다음 예를 고려합니다.

- `PropValue(Concat(Core.Abbrev))` 공식은 적합하지만 등록정보 이름을 확인하기 위해 구문을 위한 검증뿐 아니라 `Concat` 함수를 평가해야 합니다.

- `PropValue(If(NodeIsLeaf(),Core.Abbrev,Custom.Label))` 공식은 적합하지만 등록정보 이름을 확인하기 위해 `If` 함수를 평가해야 합니다.

해당 등록정보 이름이 공식의 일부만으로 구성된 경우 등록정보 이름을 확인하는 데 필요한 부분만 평가됩니다. 예를 들어 `Add(PropValue(Concat(Core.,I,D)),If(NodeIsLeaf(),0,1))` 공식에서 구문 검증을 위해 평가되는 공식 부분은 `Concat` 함수와 해당 매개변수뿐입니다.

이러한 공식 부분이 평가된다는 사실은 `PropValue(PropValue(NodeType))` 등의 경우에 중요합니다. 이 공식의 경우 `Custom.NodeType` 등록정보의 값을 지정하지 않으면 구문 검증에 실패합니다.

## 공식 사용 시의 고려 사항

### 데이터 유형 변환

일부 함수의 경우 제대로 평가되기 위해 데이터 값이 특정 데이터 유형이어야 합니다. 예를 들어 수학 계산을 수행하는 함수의 경우 입력 인수가 정수 또는 부동 소수점 값이어야 하고, 문자열 조작 함수의 경우 문자열 값이 입력으로 제공되어야 합니다. 성공적으로 파생되기 위해 데이터 값을 한 데이터 유형에서 다른 데이터 유형으로 변환해야 하는 경우도 있습니다. Oracle Data Relationship Management는 공식 내에서 데이터 유형 변환을 처리하기 위한 함수 세트를 제공합니다.

### 등록정보 레벨 제한 사항

일반적으로, 하위 세분화 레벨의 데이터를 관리하기 위해 생성된 등록정보 정의는 상위 세분화 레벨의 데이터를 관리하는 다른 등록정보를 참조할 수 있습니다.

- 로컬 노드 – 다른 로컬 노드, 글로벌 노드, 계층 또는 버전 등록정보를 참조할 수 있습니다.
- 글로벌 노드 – 다른 글로벌 노드 또는 버전 등록정보를 참조할 수 있습니다.
- 계층 – 다른 계층 또는 버전 등록정보를 참조할 수 있습니다(록업 전용).
- 버전 – 다른 버전 등록정보를 참조할 수 있습니다(록업 전용).

### 다른 노드의 등록정보 참조

파생 등록정보 또는 검증에서 공식이 계산되는 현재 노드와는 다른 노드의 등록정보 값을 평가 또는 검색하는 것이 일반적입니다. Data Relationship Management는 동일한 버전 내의 노드 등록정보 값에 액세스할 수 있게 하는 여러 개의 함수를 제공합니다.

- `NodePropValue`
- `ParentPropValue`
- `HierNodePropValue`
- `AncestorProp`
- `DualAncestorProp`
- `AscNodeProp`
- `ReplacePropValue`
- `ListPropValues`
- `ListNodePropValues`

### 글로벌 노드 등록정보에서 로컬 노드 등록정보 참조

글로벌 노드 등록정보는 값을 반환하기 위해 계층 컨텍스트가 필요하지 않지만 로컬 노드 등록정보는 계층을 지정해야 합니다. 글로벌 노드에 대해 계산된 검증 또는 파생 등록정보는 표준

PropValue 또는 NodePropValue 함수를 사용하여 로컬 노드 등록정보 값을 참조할 수 없습니다. 글로벌 노드 등록정보는 HierNodePropValue 함수를 사용하여 로컬 노드 등록정보 값을 참조할 수 있습니다. 이 경우 계층에 있는 특정 로컬 노드의 등록정보 값을 검색하기 위해 특정 계층을 지정해야 합니다.

### 중첩 함수

함수를 동일한 공식으로 결합하는 것을 중첩 함수라고 합니다. 한 함수의 출력이 공식에 있는 다른 함수의 입력 인수로 사용됩니다. 중첩 함수를 평가할 때 Data Relationship Management는 가장 안쪽 함수를 먼저 실행한 다음 바깥쪽으로 평가합니다. 동일한 공식 내에서 명시적으로 함수를 중첩하거나, 다른 공식을 사용하는 등록정보를 참조하는 공식을 사용하여 암시적으로 함수를 중첩할 수 있습니다.

### 등록정보를 다른 등록정보의 변수로 사용

Data Relationship Management에서는 중첩 함수의 조합, 다른 등록정보 또는 노드에 대한 참조 및 리터럴 값을 사용할 수 있으며, 이 경우 긴 공식이나 복합 공식이 생성될 수 있습니다. 개별 등록정보 정의를 사용하여 공식 논리를 모듈화하고 동일한 결과를 얻는 데 필요한 공식 구문을 간소화할 수 있습니다. 이 방법을 사용할 경우 이러한 공식의 유지 관리가 훨씬 쉬워질 수 있습니다.

공식이 동일한 등록정보 정의 내에서 또는 지정된 노드에 대한 여러 등록정보 정의에서 동일한 데이터를 여러 번 평가하거나 동일한 계산을 여러 번 수행할 수도 있습니다. 이 논리가 훨씬 더 큰 공식에 포함되거나 등록정보 정의 내에 구현된 경우 이러한 확인과 계산이 여러 번 수행되며, 이로 인해 등록정보를 계산해야 하는 작업의 성능이 저하될 수 있습니다. 개별 등록정보 내의 중복 공식 논리를 격리하여 중복 처리를 최소화할 수 있습니다.

### 반복을 사용하여 계층 관계 통과

계층의 최하위 레벨에 있는 노드의 비즈니스 규칙에는 상위 멤버 노드의 등록정보 값에 대한 평가가 필요할 수도 있습니다. 하위 레벨 노드가 이러한 등록정보 값을 참조할 수 있게 하는 한 가지 방법은 참조해야 하는 값을 관리하는 등록정보 정의에서 상속을 활성화하는 것입니다. 그러나 대부분의 경우 등록정보 정의에 상속을 사용하는 것은 적절하지 않습니다.

상위 멤버 노드의 등록정보 값을 검색하거나 평가하기 위해 현재 등록정보 정의에 대한 자체 참조가 있는 특정 계층 공식 함수를 사용하여 계층의 분기를 위로 반복할 수 있습니다.

**ParentPropValue** – 이 함수를 사용하여 현재 계층에서 상위 멤버의 분기를 위로 반복합니다. 예: `If (Equals (Integer, PropValue (Core.Level), 1), Label Only, ParentPropValue (Essbase.DataStorage))`

**HierNodePropValue** – 이 함수를 사용하여 다른 계층에서 상위 멤버의 분기를 위로 반복합니다. 예: `If (Equals (Boolean, PropValue (Custom.PlanPoint), True), Abbrev (), HierNodePropValue (Geography, HierNodePropValue (Essbase.DataStorage)))`

## 공식 생성

공식은 파생 등록정보 정의 및 검증을 생성하거나 편집하기 위해 [매개변수] 탭에서 사용할 수 있는 공식 편집기에서 생성됩니다.

공식을 생성하려면 다음을 수행합니다.

1. **매개변수** 탭에서 다음과 같은 방법으로 텍스트 공식을 입력하거나 함수 및 등록정보를 삽입할 수 있습니다.

- 함수를 삽입하려면 공식에 커서를 놓고 **함수 삽입**을 누릅니다. 함수 목록이 표시됩니다. 함수를 확장하여 입력 매개변수를 봅니다. 매개변수 값을 입력하고 **확인**을 누릅니다.
  - 등록정보를 삽입하려면 공식에 커서를 놓고 **등록정보 삽입**을 누릅니다. 등록정보 목록이 표시됩니다. 등록정보를 선택하고 **확인**을 누릅니다.
2. 다음 옵션 중에서 선택합니다.
- **공백 제거** – 기본적으로 선택되어 있습니다. 선택하면 공식을 평가하고 등록정보를 저장할 때 공식의 모든 공백이 제거됩니다. 공식에서 리터럴 값으로 평가할 공백을 보존하려면 이 옵션을 비활성화합니다.
  - 공식을 평가하려면 옵션을 선택합니다.
    - **선택한 노드로 평가** – 을 누르고 노드를 선택합니다. 노드의 현재 등록정보 값이 공식에 사용됩니다. **평가**를 누릅니다. 결과가 공식 디자이너의 맨아래에 표시됩니다.
    - **스크래치 패드로 평가** – 등록정보 값을 수동으로 입력합니다. 노드에서 값을 복사한 다음 평가를 위해 수정할 수도 있습니다. [복사 대상 노드]에서 을 누르고 노드를 선택하여 그리드에 해당 등록정보 값을 표시합니다. 열 제목 아래의 필터 행을 사용하여 등록정보 목록을 필터링합니다. [작업] 열의 [편집] 버튼을 사용하여 공식으로 평가하기 위해 등록정보 값을 수정합니다. [평가]를 누릅니다. 평가 결과가 공식 디자이너의 맨아래에 표시됩니다.
3. 공식을 테스트하려면 **평가**를 누릅니다.

## 함수 정의

다음은 파생된 공식 등록정보 정의에 사용할 수 있는 함수 목록으로, 알파벳순으로 표시됩니다.

### Abbrev

#### 설명

현재 노드의 이름(Abbrev)을 반환합니다.

#### 구문

```
Abbrev(): String
```

#### 예

```
Abbrev()
```

반환 값은 노드의 이름입니다.

### Add

#### 설명

지정된 두 정수 값을 더한 후 결과를 반환합니다.

#### 구문

```
Add(Int1, Int2: Integer): Integer
```

**예**

```
Add(1, 4)
```

반환 값은 5입니다.

**AddedBy****설명**

추가한 사람 변경 추적 등록정보의 값을 반환합니다.

**구문**

```
AddedBy():String
```

**예**

```
AddedBy()
```

현재 노드를 버전에 추가한 사용자의 이름을 반환합니다.

**AddedOn****설명**

추가한 날짜 변경 추적 등록정보의 값을 날짜/시간으로 반환합니다.

**구문**

```
AddedOn():Date/Time
```

**예**

```
AddedOn()
```

현재 노드가 버전에 추가된 날짜 및 시간을 반환합니다.

**AddFloat****설명**

지정된 두 부동 소수점 값을 더한 후 결과를 반환합니다.

**구문**

```
AddFloat(Float1,Float2:Float):Float
```

**예**

```
AddFloat(2.14,3.75)
```

반환 값은 5.89입니다.

### AncestorProp

#### 설명

등록정보가 지정된 값과 같은 첫 번째 상위 멤버의 등록정보 값을 반환합니다.

이 함수는 로컬 범위로 제한되며 글로벌 컨텍스트에서 사용되는 경우 제대로 작동하지 않습니다.

#### 주:

현재 노드가 기준에 적합한 경우 해당 노드가 반환됩니다.

### 구문

```
AncestorProp (Operator:String, Property:String, Value:String, FromTop:Boolean, ReturnProp:String)
```

**Operator**는 등록정보를 값과 비교할 때 사용할 연산자입니다. 적합한 값은 =, <, >, >= 및 <= 입니다.

**Property**는 사용할 등록정보의 이름입니다.

**Value**는 비교할 값입니다.

**FromTop**은 계층의 맨위 노드부터 검색할지 여부를 지정합니다. False인 경우 검색은 현재 노드부터 수행됩니다.

**ReturnProp**는 반환할 등록정보의 이름입니다.

### And

#### 설명

지정된 모든 부울 표현식이 True인 경우 True를 반환합니다.

#### 구문

```
And (Expression1, Expression2, ... ExpressionN:Boolean) : Boolean
```

#### 예

```
And (1, T, True)
```

반환 값은 True입니다.

### ArrayCount

#### 설명

지정된 목록(배열)에 있는 항목 수를 반환합니다.

## 구문

```
ArrayCount(List:String,Delimiter:String):Integer
```

**List**는 검색할 문자열 목록을 지정합니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [심표]
- [공백]
- [탭]



주:

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

## 예

```
ArrayCount(Diet Cola;Root Beer;Cola,[comma])
```

반환 값은 3입니다.

## ArrayIndex

### 설명

목록(배열) 내에서 지정된 항목의 첫 번째 발생 위치를 반환합니다. 항목을 찾을 수 없는 경우 0을 반환합니다.

## 구문

```
ArrayIndex(Item:String,List:String,Delimiter:String):Integer
```

**Item**은 테스트할 문자열 값을 지정합니다.

**List**는 검색할 문자열 목록을 지정합니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [심표]
- [공백]
- [탭]

**주:**

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**예**

```
ArrayIndex(Cola,Diet Cola;Root Beer;Cola,[comma])
```

반환 값은 3입니다.

**ArrayItem****설명**

목록(배열)에서 지정된 인덱스 위치의 항목을 반환합니다.

**구문**

```
ArrayItem(List:String,Delimiter:String,Index:Integer):String
```

**List**는 검색할 문자열 목록을 지정합니다.**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [심표]
- [공백]
- [탭]

**주:**

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**Index**는 목록에서 문자열의 위치를 나타냅니다. 음수 값은 목록에서 마지막 항목을 나타냅니다.**예**

```
ArrayItem(Diet Cola;Root Beer;Cola,;,3)
```

반환 값은 Cola입니다.

**AscNodeProp****설명**

지정된 등록정보가 참조하는 연결된 노드의 등록정보 값을 반환합니다.

## 구문

```
AscNodeProp (LookUpProperty, ReturnProperty)
```

**LookUpProperty**는 노드를 가리키는 등록정보의 이름입니다. 등록정보는 Node 또는 AscNode 데이터 유형이어야 합니다.

**ReturnProperty**는 반환할 연결된 노드의 등록정보 이름입니다. 등록정보는 글로벌이어야 합니다.

## AvgList

### 설명

빈 항목을 무시하고 목록에 있는 항목의 평균을 반환합니다. 지정된 항목 유형이 아닌 항목이 목록에 포함된 경우 빈 문자열을 반환합니다.

## 구문

```
AvgList (InputList:String, Delimiter:String, ItemType:String) :String
```

**InputList**는 사용할 목록을 지정합니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [십표]
- [공백]
- [탭]

### 주:

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**ItemType**은 목록 멤버의 예상 항목 데이터 유형을 나타냅니다. 적합한 값은 integer, float 및 datetime입니다. 기본값은 float입니다.

## 예

```
AvgList (1;2;3, [comma], Integer)
```

반환 값은 2입니다.

## BoolToStr

### 설명

부울 값을 True 또는 False로 변환하여 반환합니다. 입력이 부울 값을 나타내지 않는 경우 False를 반환합니다.

## 구문

```
BoolToStr(Expression:Boolean):String
```

## 예

```
BoolToStr(1)
```

반환 값은 True입니다.

## Changed

### 설명

노드 변경됨 변경 추적 등록정보의 값을 부울로 반환합니다.

## 구문

```
Changed()
```

## ChangedBy

### 설명

버전에서 현재 노드를 최종 업데이트한 사용자의 이름을 반환합니다.

## 구문

```
ChangedBy():String
```

## 예

```
ChangedBy()
```

## ChangedOn

### 설명

변경한 날짜 변경 추적 등록정보의 값을 날짜/시간으로 반환합니다.

## 구문

```
ChangedOn():Date/Time
```

## 예

```
ChangedOn()
```

현재 노드가 버전에서 최종 업데이트된 날짜 및 시간을 반환합니다.

## Concat

### 설명

두 개 이상의 지정된 문자열을 하나로 연결한 후 결과를 반환합니다.

### 구문

```
Concat (Item1, Item2, ... ItemN:String) :String
```

### 예

```
Concat (Abbrev, -, Descr ())
```

현재 노드 이름이 100이고 현재 노드 설명이 Colas이면 반환 값은 100-Colas입니다.

## ConcatWithDelimiter

### 설명

두 개 이상의 지정된 문자열을 하나의 구분된 목록으로 연결한 후 결과를 반환합니다.

### 구문

```
ConcatWithDelimiter (Delimiter:String, SkipBlanks:Boolean, Items:String)
```

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [심표]
- [공백]
- [탭]

### 주:

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**SkipBlanks**는 문자열 목록에서 빈 값을 건너뛰는 여부를 나타냅니다. 적합한 값은 1, 0, T, F, t, f입니다.

**Items**는 연결할 문자열 목록을 지정합니다.

### 예

```
ConcatWithDelimiter ([comma], 1, Item1, Item2, Item3, Item4)
```

반환 값은 Item1; Item2; Item3; Item4입니다.

## Decode

### 설명

[openparen], [closeparen], [comma], [tab], [space], [crlf], [cr] 및 [lf] 인스턴스가 모두 적절한 문자로 바뀐 입력 문자열을 반환합니다.

### 주:

이 함수는 특수 문자를 사용하는 등록정보 정의 이름을 업그레이드하는 데 사용됩니다. 이러한 특수 문자로 인해 파생 등록정보 공식에서 구문분석 이슈가 발생할 수 있습니다. 이 함수는 사용되지 않는 `Deriver` 클래스를 사용한 기존 등록정보를 `Formula Deriver` 클래스로 변환하는 데 주로 사용됩니다.

### 구문

```
Decode (CodedString:String) :String
```

**CodedString**은 함수를 실행할 문자열 값입니다.

### DefaultProp

#### 설명

등록정보의 기본값을 반환합니다.

#### 구문

```
DefaultProp (Property:String)
```

**Property**는 사용할 등록정보의 이름입니다.

### Descr

#### 설명

현재 노드에 대한 설명을 반환합니다.

#### 구문

```
Descr() :String
```

### 예

현재 노드 설명이 `Colas`이면 반환 값은 `Colas`입니다.

### Divide

#### 설명

지정된 두 정수 값을 나눈 후 결과를 반환합니다.

## 구문

```
Divide(Int1, Int2: Integer) : Integer
```

## 예

```
Divide(200, 10)
```

반환 값은 20입니다.

## DivideFloat

### 설명

두 개의 부동 소수점 수(부동)를 나눈 후 결과를 반환합니다.

## 구문

```
Divide(Float1, Float2: Float) : Float
```

## 예

```
DivideFloat(2.535, 1.5)
```

반환 값은 1.69입니다.

## DualAncestorProp

### 설명

두 개의 등록정보가 지정된 값과 같은 첫 번째 상위 멤버의 등록정보 값을 반환합니다.

이 함수는 로컬 범위로 제한되며 글로벌 컨텍스트에서 사용되는 경우 제대로 작동하지 않습니다.

## 구문

```
DualAncestorProp(Operator1: String, Property1: String, Value1: String, Operator2: String, Property2: String, Value2: String) : String
```

**Operator1**은 첫 번째 등록정보와 값을 비교할 때 사용할 연산자입니다. 적합한 값은 =, <, >, >= 및 <=입니다.

**Property1**은 확인할 첫 번째 등록정보의 이름입니다.

**Value1**은 비교할 첫 번째 값입니다.

**Operator2**는 두 번째 등록정보와 값을 비교할 때 사용할 연산자입니다. 적합한 값은 =, <, >, >= 및 <=입니다.

**Property2**는 확인할 두 번째 등록정보의 이름입니다.

**Value2**는 비교할 두 번째 값입니다.

**FromTop**은 계층의 맨위 노드부터 검색할지 여부를 지정합니다. `False`인 경우 검색은 현재 노드부터 수행됩니다.

**ReturnProp**는 반환할 상위 멤버의 등록정보 이름입니다.

## Equals

### 설명

지정된 두 값이 같은 경우 `True`를 반환합니다. 이 함수는 대소문자를 구분합니다.

### 구문

```
Equals (ParamType:String, Param1:String, Param2:String) : Boolean
```

**ParamType**은 값 비교에 사용할 데이터 유형입니다. 적합한 값은 `string`, `integer`, `float` 및 `date`입니다. 기본값은 `integer`입니다.

**Param1**은 비교할 첫 번째 값입니다.

**Param2**는 비교할 두 번째 값입니다.

### 예

```
Equals (integer, 01, 1)
```

반환 값은 `True`입니다.

## FlipList

### 설명

지정된 목록의 반대를 나타내는 문자열을 반환합니다.

### 구문

```
FlipList (List, Delimiter:String) : String
```

**List**는 뒤집을 문자열 목록을 지정합니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [심표]
- [공백]
- [탭]

### 주:

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**예**

```
FlipList(DietCola;Orange Soda;Root Beer;Lemonade,[comma])
```

반환 값은 Lemonade,Root Beer,Orange Soda,Diet Cola입니다.

**FloatToStr****설명**

부동 소수점 값을 문자열로 변환하여 반환합니다. 입력 값이 부동 소수점을 나타내지 않는 경우 0을 반환합니다.

**구문**

```
FloatToStr(Float1:Float):String
```

**예**

```
FloatToStr(1.001)
```

반환 값은 1.001입니다.

**형식****설명**

지정된 형식 문자열 매개변수 유형 식별자와 지정된 유형의 매개변수 값을 사용하여 값의 형식을 지정합니다. 이 함수는 단일 값 매개변수로 제한됩니다.

**구문**

```
Format(Format:String,ParamType:String, ValueToFormat:String):String
```

**Format**은 적용할 형식입니다.

**ParamType**은 값 비교에 사용할 데이터 유형입니다. 적합한 값은 string, integer, float 및 date입니다. 기본값은 integer입니다.

**ValueToFormat**은 함수를 실행할 값입니다.

**예**

```
Format('%8.2f',Float,123.456)
```

반환 값은 123.46입니다.

**FormattedDate****설명**

지정된 형식 문자열을 사용하여 형식이 지정된 날짜 등록정보의 값을 반환합니다.

## 구문

```
FormattedDate (PropertyName:String,FormatString:String) : String
```

**PropertyName**은 사용할 등록정보의 이름입니다.

**FormatString**은 적용할 날짜 형식을 지정합니다.

## GreaterThan

### 설명

두 값을 비교하고 첫 번째 값이 두 번째 값보다 큰 경우 True를 반환합니다.

## 구문

```
GreaterThan (Value1:Integer,Value2:Integer,ParamType:String) :Boolean
```

**Value1**은 비교할 첫 번째 값입니다.

**Value2**는 비교할 두 번째 값입니다.

**ParamType**은 값 비교에 사용할 데이터 유형입니다. 적합한 값은 string, integer, float 및 date입니다. 기본값은 integer입니다.

## 예

```
GreaterThan (1,2)
```

반환 값은 False입니다.

## GreaterThanOrEqual

### 설명

두 값을 비교하고 첫 번째 값이 두 번째 값보다 크거나 같은 경우 True를 반환합니다.

## 구문

```
GreaterThanOrEqual (Value1:Integer,Value2:Integer,ParamType:String) :Boolean
```

**Value1**은 비교할 첫 번째 값입니다.

**Value2**는 비교할 두 번째 값입니다.

**ParamType**은 값 비교에 사용할 데이터 유형입니다. 적합한 값은 string, integer, float 및 date입니다. 기본값은 integer입니다.

## 예

```
GreaterThanOrEqual (2,2)
```

반환 값은 True입니다.

## HasCharacters

### 설명

지정된 입력에 문자 클래스, 특수 문자 또는 CharList에 나열된 문자의 문자가 포함된 경우 True를 반환합니다.

### 구문

```
HasCharacters (Input:String,CharList:String) :Boolean
```

**Input**은 테스트할 문자열 값입니다.

**CharList**는 선택적 특수 값을 포함한 테스트할 문자 목록입니다. 특수 문자 값은 대괄호로 묶이고 쉼표로 구분됩니다. 적합한 값은 [alpha], [numeric], [whitespace], [punctuation], [uppercase], [lowercase], [comma], [space], [tab], [crLf], [cr], [lf], [openparen] 및 [closeparen]입니다.

## HasChildWith

### 설명

지정된 표현식이 현재 노드의 하위 멤버에 대해 True인 경우 True를 반환합니다.

### 구문

```
HasChildWith (Expression:Boolean) :Boolean
```

### 예

```
HasChildWith (GreaterThan (ID (), 200) )
```

현재 노드에 ID가 200보다 큰 하위 멤버가 있으면 반환 값은 True입니다.

## HasParentNode

### 설명

현재 노드에 상위 노드가 있는 경우 True를 반환합니다.

이 함수는 로컬 범위로 제한되며 글로벌 컨텍스트에서 사용되는 경우 제대로 작동하지 않습니다.

### 구문

```
HasParentNode () :Boolean
```

### 예

```
HasParentNode ()
```

노드가 계층 최상위 노드의 하위 멤버이거나 하위 멤버 노드이면 반환 값은 True입니다.

## HasSiblingWith

### 설명

지정된 표현식이 현재 노드의 동위 멤버에 대해 True인 경우 True를 반환합니다.

이 함수는 로컬 범위로 제한되며 글로벌 컨텍스트에서 사용되는 경우 제대로 작동하지 않습니다.

### 구문

```
HasSiblingWith(Expression:Boolean):Boolean
```

### 예

```
HasSiblingWith(PropValue(Leaf))
```

하위 멤버가 하나라도 리프이면 반환 값은 True입니다.

## HierNodePropValue

### 설명

지정된 계층의 지정된 노드에 대해 지정된 등록정보 값을 반환합니다.

### 구문

```
HierNodePropValue(HierAbbrev:String,NodeAbbrev:String,PropAbbrev:String):String
```

**HierAbbrev**는 사용할 계층의 이름입니다.

**NodeAbbrev**는 사용할 노드의 이름입니다.

**PropAbbrev**는 사용할 등록정보의 이름입니다.

### 예

```
HierNodePropValue(Assets,1000,Description)
```

Assets 계층에 있는 노드 1000의 설명이 "Banking"이면 반환 값은 Banking입니다.

## ID

### 설명

현재 노드의 ID를 반환합니다.

### 구문

```
ID():Integer
```

**예**

```
ID()
```

현재 노드 ID가 2000이면 반환 값은 2000입니다.

**If****설명**

지정된 표현식이 True인 경우 TrueResult 매개변수의 값을 반환합니다. 그렇지 않으면 FalseResult 매개변수의 값을 반환합니다.

**구문**

```
If(Expression:Boolean, TrueResult:String, FalseResult:String):String
```

**Expression**은 평가할 부울 표현식입니다.

**TrueResult**는 조건이 True인 경우 반환할 문자열 값입니다.

**FalseResult**는 조건이 False인 경우 반환할 문자열 값입니다.

**예**

```
If(Equals(String, Descr(), ), Abbrev(), Concat(Abbrev, -, Descr()))
```

노드 이름이 Colas이고 현재 노드 설명이 비어 있으면 반환 값은 Colas입니다.

노드 이름이 100이고 현재 노드 설명이 Colas이면 반환 값은 100-Colas입니다.

**InheritedPropOrigin****설명**

상속된 등록정보 값의 원본 노드 이름을 반환합니다. 지정된 등록정보가 글로벌이면 원본 계층도 반환됩니다. 지정된 등록정보가 상속되지 않거나 노드 등록정보를 찾을 수 없는 경우 False를 반환합니다.

로컬 등록정보가 매개변수로 전달되는 경우 이 함수는 로컬 범위로 제한될 수 있습니다.

**구문**

```
InheritedPropOrigin(PropAbbrev:String, Node:String):String
```

**예**

```
InheritedPropOrigin(Custom.AccountType, Abbrev())
```

**PropAbbrev**는 사용할 등록정보의 이름입니다.

**Node**는 사용할 노드의 이름입니다.

## InRange

### 설명

지정된 값이 지정된 값 범위 내에 해당하는 경우 `True`를 반환합니다. 입력 매개변수가 문자열이면 `Min` 및 `Max` 매개변수가 확인할 문자열 길이 범위를 지정합니다. 다른 유형의 경우 `Min` 및 `Max`가 확인할 숫자 또는 날짜 값 범위를 지정합니다.

### 주:

`MinExclusive/MaxExclusive`가 `True`이면 `Min/Max`와 같은 값이 범위에 포함되고, 그렇지 않으면 제외됩니다.

### 구문

```
InRange (DataType:String, Input:String, Min:String, Max:String, MinExclusive:String, MaxExclusive:String)
```

**DataType**은 사용할 데이터 유형입니다. 적합한 값은 `string`, `integer`, `float` 및 `datetime`입니다.

**Input**은 테스트할 문자열 값입니다.

**Min**은 길이 또는 범위 확인의 최소값입니다.

**Max**는 길이 또는 범위 확인의 최대 값입니다.

**MinExclusive**는 확인할 범위에서 `Min` 값을 제외할지 여부를 지정합니다.

**MaxExclusive**는 확인할 범위에서 `Max` 값을 제외할지 여부를 지정합니다.

### 예

```
InRange (Integer, 5, 1, 10, False, False)
```

반환 값은 `True`입니다.

## InternalPrefix

### 설명

현재 노드의 이름에서 숫자가 아닌 접두어를 반환합니다.

### 구문

```
InternalPrefix ()
```

## Intersection

### 설명

두 개의 지정된 값 목록에서 공통된 항목 세트를 반환합니다. 지정된 첫 번째 목록에 항목이 표시되는 방식에 따라 결과가 정렬됩니다.

## 구문

```
Intersection(List1:String, List2:String, Delimiter:String) :String
```

**List1**은 검색할 문자열 목록을 지정합니다.

**List2**는 검색할 문자열 목록을 지정합니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [심표]
- [공백]
- [탭]

### 주:

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

## 예

```
Intersection(A;B;C;D;E,C;E;F;A,[comma])
```

반환 값은 A,C,E입니다.

## IntToStr

### 설명

지정된 정수 값을 문자열 데이터 유형으로 변환하여 반환합니다. 입력 값이 정수를 나타내지 않는 경우 0을 반환합니다.

## 구문

```
IntToStr(Int1:Integer) :String
```

## 예

```
IntToStr(12345)
```

반환 값은 12345입니다.

## InvertedLevel

### 설명

현재 노드 아래에 있는 하위 멤버의 최대 깊이를 반환합니다.

## 구문

```
InvertedLevel ()
```

## IsAlpha

### 설명

지정된 문자열에 영문자(대소문자 구분 안 함)만 포함된 경우 True를 반환합니다.

### 구문

```
IsAlpha (String:String) :Boolean
```

### 예

```
IsAlpha (A23D)
```

반환 값은 False입니다.

## IsAlphaNumeric

### 설명

지정된 문자열에 영문자 또는 숫자 문자(대소문자 구분 안 함)만 포함된 경우 True를 반환합니다.

### 구문

```
IsAlphaNumeric (String:String, AllowBlanks:Boolean) :Boolean
```

**String**은 테스트할 문자열 값입니다.

**AllowBlanks**는 빈 문자열을 숫자로 처리할지 여부를 지정합니다. 기본값은 False입니다.

### 예

```
IsAlphaNumeric (ABC123, True)
```

True를 반환합니다.

## IsBlank

### 설명

지정된 입력 값이 빈 문자열(길이 0)인 경우 True를 반환합니다.

### 구문

```
IsBlank (Input:String) :Boolean
```

**예**

```
IsBlank (Descr ())
```

노드 설명이 비어 있는 경우 True를 반환합니다.

**IsBottomNode****설명**

지정된 노드에 하위 노드가 없는 경우 True를 반환합니다. 노드를 찾을 수 없는 경우 False를 반환합니다.

**구문**

```
IsBottomNode (Node:String) :Boolean
```

**Node**는 사용할 노드의 이름입니다.

**예**

```
IsBottomNode (Abbrev)
```

노드에 하위 멤버가 없는 경우 True를 반환합니다.

**IsDataType****설명**

입력 값이 지정된 데이터 유형과 일치하는 경우 True를 반환합니다.

**구문**

```
IsDataType (DataType:String, Input:String) :Boolean
```

**DataType**은 사용할 데이터 유형입니다. 적합한 값은 boolean, string, integer, float 및 datetime입니다.

**Input**은 테스트할 문자열 값입니다.

**예**

```
IsDataType (123, Integer)
```

True를 반환합니다.

**IsDefinedPropVal****설명**

지정된 노드의 지정된 등록정보에 정의된(재정의된) 값이 있는 경우 True를 반환합니다. 노드 또는 등록정보를 찾을 수 없는 경우 False를 반환합니다.

로컬 등록정보가 매개변수로 전달되는 경우 이 함수는 로컬 범위로 제한될 수 있습니다.

### 구문

```
IsDefinedPropVal (PropAbbrev:String,Node:String) :Boolean
```

**PropAbbrev**는 사용할 등록정보의 이름입니다.

**Node**는 사용할 노드의 이름입니다.

### 예

```
IsDefinedPropVal (Custom.AccountType,Abbrev())
```

계정 유형 등록정보에 정의된(재정의된) 값이 있는 경우 **True**를 반환합니다.

### IsNodeAbove

#### 설명

첫 번째 노드가 현재 계층에서 두 번째 노드의 상위 멤버인 경우 **True**를 반환합니다. **Node1** 또는 **Node2**를 찾을 수 없는 경우 **False**를 반환합니다.

이 함수는 로컬 범위로 제한되며 글로벌 컨텍스트에서 사용되는 경우 제대로 작동하지 않습니다.

### 구문

```
IsNodeAbove (Node1:String,Node2:String) :Boolean
```

**Node1**은 사용할 첫 번째 노드의 이름입니다.

**Node2**는 사용할 두 번째 노드의 이름입니다.

### 예

```
IsNodeAbove (Parent,Child)
```

상위 노드가 하위 노드의 상위 멤버인 경우 **True**를 반환합니다.

### IsNodeBelow

#### 설명

첫 번째 노드가 현재 계층에서 두 번째 노드의 하위 멤버인 경우 **True**를 반환합니다. **Node1** 또는 **Node2**를 찾을 수 없는 경우 **False**를 반환합니다.

### 구문

```
IsNodeBelow (Node1:String,Node2:String) :Boolean
```

**Node1**은 사용할 첫 번째 노드의 이름입니다.

**Node2**는 사용할 두 번째 노드의 이름입니다.

## 예

```
IsNodeBelow(Child, Parent)
```

하위 노드가 상위 노드의 하위 멤버인 경우 `True`를 반환합니다.

## IsNumeric

### 설명

지정된 값에 숫자 문자(0-9)만 포함된 경우 `True`를 반환합니다.

### 구문

```
IsNumeric(String: String, AllowBlanksAsNumeric: Boolean): Boolean
```

**String**은 테스트할 문자열 값입니다.

**AllowBlanksAsNumeric**은 빈 값을 문자열로 간주할지 여부를 지정합니다. 기본값은 `False`입니다.

## 예

```
IsNumeric(12345)
```

반환 값은 `True`입니다.

## IsRangeListSubset

### 설명

지정된 값이 지정된 범위 목록의 서브세트인 경우 `True`를 반환합니다.

### 구문

```
IsRangeListSubset(RangeList: Range List, SubsetRangeList: Range List, Delimiter: String): Boolean
```

**RangeList**는 지정된 구분자로 구분되는 검색할 정수 범위 목록입니다.

**SubsetRangeList**는 지정된 구분자로 구분되는 검색할 정수 범위의 수퍼 세트 목록입니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [침표]
- [공백]
- [탭]

**주:**

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

## Length

### 설명

지정된 문자열 값에 있는 문자 수를 반환합니다.

### 구문

```
Length (String:String) :Integer
```

### 예

```
Length (Desc ())
```

현재 노드의 설명이 Colas이면 반환 값은 5입니다.

## LessThan

### 설명

두 값을 비교하고 첫 번째 값이 두 번째 값보다 작은 경우 True를 반환합니다.

### 구문

```
LessThan (Value1:Integer, Value2:Integer, ParamType:String) :Boolean
```

**Value1**은 비교할 첫 번째 값입니다.

**Value2**는 비교할 두 번째 값입니다.

**ParamType**은 값 비교에 사용할 데이터 유형입니다. 적합한 값은 string, integer, float 및 date입니다. 기본값은 integer입니다.

### 예

```
LessThan (1, 2)
```

반환 값은 True입니다.

## LessThanOrEqual

### 설명

두 값을 비교하고 첫 번째 값이 두 번째 값보다 작거나 같은 경우 True를 반환합니다.

## 구문

```
LessThanOrEqual (Value1: Integer, Value2: Integer, ParamType: String) : Boolean
```

**Value1**은 비교할 첫 번째 값입니다.

**Value2**는 비교할 두 번째 값입니다.

**ParamType**은 값 비교에 사용할 데이터 유형입니다. 적합한 값은 string, integer, float 및 date입니다. 기본값은 integer입니다.

## 예

```
LessThanOrEqual (3, 3)
```

반환 값은 True입니다.

## ListAncestors

### 설명

최상위 노드부터 시작하여 현재 노드의 상위 멤버 이름을 심표로 구분된 목록으로 반환합니다. 현재 노드가 로컬 노드가 아닌 경우 빈 문자열을 반환합니다.

이 함수는 로컬 범위로 제한되며 글로벌 컨텍스트에서 사용되는 경우 제대로 작동하지 않습니다.

## 구문

```
ListAncestors (SortOrder: String) : String
```

**SortOrder**는 노드 반환 목록의 정렬 순서를 지정합니다. 지원되는 정렬 순서 값은 다음과 같습니다.

- [hier] – 로컬 컨텍스트의 기본값입니다. 노드 목록이 현재 계층의 표준 계층 정렬 순서로 반환됩니다.
- [alpha] – 노드 목록이 노드 이름별로 정렬되어 반환됩니다.
- [nodeid] – 레거시 호환성을 위해 제한적으로 사용됩니다. 노드 목록이 반환 목록에 있는 각 노드의 노드 ID별로 숫자로 정렬되어 반환됩니다.

### 주:

SortOrder 매개변수를 대괄호로 묶어야 합니다.

## 예

```
ListAncestors ([alpha])
```

A, B, C 및 D가 Z의 하위 멤버이고 Z가 Y의 하위 멤버이고 현재 노드가 B이면 반환 값은 Z,Y입니다.

### ListChildren

#### 설명

현재 노드의 하위 멤버를 심표로 구분된 목록으로 반환합니다.

#### 구문

```
ListChildren(SortOrder:String):String
```

**SortOrder**는 노드 반환 목록의 정렬 순서를 지정합니다. 지원되는 정렬 순서 값은 다음과 같습니다.

- [hier] – 로컬 컨텍스트의 기본값입니다. 노드 목록이 현재 계층의 표준 계층 정렬 순서로 반환됩니다.
- [alpha] – 노드 목록이 노드 이름별로 정렬되어 반환됩니다.
- [nodeid] – 레거시 호환성을 위해 제한적으로 사용됩니다. 노드 목록이 반환 목록에 있는 각 노드의 노드 ID별로 숫자로 정렬되어 반환됩니다.



#### 주:

SortOrder 매개변수를 대괄호로 묶어야 합니다.

#### 예

```
ListChildren([alpha])
```

A, B, C 및 D가 Z의 하위 멤버이고 현재 노드가 Z이면 반환 값은 A, B, C, D입니다.

### ListContains

#### 설명

지정된 목록에 지정된 값이 포함된 경우 True를 반환합니다.

#### 구문

```
ListContains(List:String,Item:String,Delimiter:String):Boolean
```

**List**는 검색할 문자열 목록을 지정합니다.

**Item**은 함수를 실행할 문자열 값을 지정합니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [심표]
- [공백]

## • [탭]

**주:**

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**예**

```
ListContains(PropValue(NodeList), Colas, [comma])
```

반환 값은 True입니다.

**ListDescendants****설명**

현재 노드의 하위 멤버를 심표로 구분된 목록으로 반환합니다.

**구문**

```
ListDescendants(SortOrder:String):String
```

**SortOrder**는 노드 반환 목록의 정렬 순서를 지정합니다. 지원되는 정렬 순서 값은 다음과 같습니다.

- [hier] – 로컬 컨텍스트의 기본값입니다. 노드 목록이 현재 계층의 표준 계층 정렬 순서로 반환됩니다.
- [alpha] – 노드 목록이 노드 이름별로 정렬되어 반환됩니다.
- [nodeid] – 레거시 호환성을 위해 제한적으로 사용됩니다. 노드 목록이 반환 목록에 있는 각 노드의 노드 ID별로 숫자로 정렬되어 반환됩니다.

**주:**

SortOrder 매개변수를 대괄호로 묶어야 합니다.

**예**

```
ListDescendants([hier])
```

A, B, C 및 D가 Z의 하위 멤버이고 Z가 Y의 하위 멤버이고 현재 노드가 Y이면 반환 값은 Z, A, B, C, D입니다.

**ListDistinct****설명**

중복 항목을 제거하고 지정된 목록에서 고유한 항목 목록을 반환합니다.

## 구문

```
ListDistinct (InputList:String,Delimiter:String):String
```

**InputList**는 사용할 목록을 지정합니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [쉼표]
- [공백]
- [탭]



### 주:

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

## 예

```
ListDistinct (A;B:C;A;D, [comma])
```

반환 값은 A,B,C,D입니다.

## ListNodePropValues

### 설명

지정된 노드 목록의 지정된 등록정보에 대해 등록정보 값 목록을 반환합니다. 찾을 수 없는 모든 노드에 대해 목록에 빈 문자열을 반환합니다.

로컬 등록정보가 매개변수로 전달되는 경우 이 함수는 로컬 범위로 제한될 수 있습니다.

## 구문

```
ListNodePropValues (NodeList:String,Delimiter:String,PropAbbrev:String):String
```

**NodeList**는 쉼표로 구분된 노드 이름 목록입니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [쉼표]
- [공백]
- [탭]

**주:**

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**PropAbbrev**는 사용할 등록정보의 이름입니다.

예

```
ListNodePropValues (100;200;300, [comma], Core.Leaf)
```

노드 100, 200 및 300이 리프 노드인 경우 `True, True, True`를 반환합니다.**ListNodesWith**

설명

지정된 노드 목록에서 지정된 표현식이 `True`인 노드 목록을 반환합니다.

구문

```
ListNodesWith (NodeList:String, Delimiter:String, Expression:String) :String
```

**NodeList**는 쉼표로 구분된 노드 이름 목록입니다.**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [쉼표]
- [공백]
- [탭]

**주:**

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**Expression**은 평가할 부울 표현식입니다.

예

```
ListNodesWith (100;200;300, [comma], NodeIsLeaf ())
```

노드 100, 200 및 300이 리프 노드인 경우 `True, True, True`를 반환합니다.**ListRelatedNodesWith**

설명

현재 노드와 관련된, 지정된 표현식이 `True`인 노드 목록을 반환합니다.

관계 매개변수가 Ancestors 또는 Siblings인 경우 이 함수는 로컬 범위로 제한됩니다.

## 구문

```
ListRelatedNodesWith (Relation:String, Expression:String, SortOrder:String, Max:Integer) :String
```

**Relation**은 다음과 같을 수 있습니다.

- 상위 멤버 – 지정된 표현식에서 로컬 등록정보를 참조할 수 있습니다.
- 동위 멤버 – 지정된 표현식에서 로컬 등록정보를 참조할 수 있습니다.
- 하위 멤버 – 지정된 표현식에서 로컬 및 글로벌 등록정보를 참조할 수 있습니다.
- 하위 멤버 – 지정된 표현식에서 로컬 및 글로벌 등록정보를 참조할 수 있습니다.

**Expression**은 평가할 부울 표현식입니다.

**SortOrder**는 노드 반환 목록의 정렬 순서를 지정합니다. 지원되는 정렬 순서 값은 다음과 같습니다.

- [hier] – 로컬 컨텍스트의 기본값입니다. 노드 목록이 현재 계층의 표준 계층 정렬 순서로 반환됩니다.
- [alpha] – 노드 목록이 노드 이름별로 정렬되어 반환됩니다.
- [nodeid] – 레거시 호환성을 위해 제한적으로 사용됩니다. 노드 목록이 반환 목록에 있는 각 노드의 노드 ID별로 숫자로 정렬되어 반환됩니다.

### 주:

SortOrder 매개변수를 대괄호로 묶어야 합니다.

**Max**는 반환할 최대 노드 수를 나타내는 정수 값입니다. 0 값이나 값이 없으면 제한 없음을 나타내고 모든 노드가 반환됩니다.

## 예

```
ListRelatedNodesWith (children, True, [alpha], 1000)
```

노드가 현재 노드의 1차 하위 구성요소인 경우 100,200,300을 반환합니다.

## ListSiblings

### 설명

현재 노드의 동위 멤버(피어)를 심표로 구분된 목록으로 반환합니다.

이 함수는 로컬 범위로 제한되며 글로벌 컨텍스트에서 사용되는 경우 제대로 작동하지 않습니다.

## 구문

```
ListSiblings (SortOrder:String) :String
```

**SortOrder**는 노드 반환 목록의 정렬 순서를 지정합니다. 지원되는 정렬 순서 값은 다음과 같습니다.

- [hier] – 로컬 컨텍스트의 기본값입니다. 노드 목록이 현재 계층의 표준 계층 정렬 순서로 반환됩니다.
- [alpha] – 노드 목록이 노드 이름별로 정렬되어 반환됩니다.
- [nodeid] – 레거시 호환성을 위해 제한적으로 사용됩니다. 노드 목록이 반환 목록에 있는 각 노드의 노드 ID별로 숫자로 정렬되어 반환됩니다.

#### 예

```
ListSiblings([alpha])
```

A, B, C 및 D가 Z의 하위 멤버이고 현재 노드가 B이면 반환 값은 A, C, D입니다.

### LowerCase

#### 설명

지정된 문자열 값을 소문자로 변환하여 반환합니다.

#### 구문

```
LowerCase(String:String):String
```

#### 예

```
LowerCase(HOBBS)
```

반환 값은 hobbes입니다.

### LTrim

#### 설명

문자열 시작에서 모든 공백을 지우고 지정된 값을 반환합니다.

#### 구문

```
LTrim(String: String): String
```

#### 예

```
LTrim(" 101203")
```

반환 값은 101203입니다.

## MaxList

### 설명

빈 항목을 무시하고 지정된 목록에서 최대 항목을 반환합니다. 지정된 유형이 아닌 항목이 목록에 포함된 경우 빈 문자열을 반환합니다.

### 구문

```
MaxList(InputList: String, Delimiter: String, ItemType: String)
```

**InputList**는 사용할 목록을 지정합니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [십표]
- [공백]
- [탭]



### 주:

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**ItemType**은 목록 멤버의 예상 항목 데이터 유형을 나타냅니다. 적합한 값은 integer, float 및 datetime입니다. 기본값은 float입니다.

### 예

```
MaxList(1;2;3,[comma],Integer)
```

반환 값은 3입니다.

## MinList

### 설명

빈 항목을 무시하고 지정된 목록에서 최소 항목을 반환합니다. 지정된 유형이 아닌 항목이 목록에 포함된 경우 빈 문자열을 반환합니다.

### 구문

```
MinList(InputList:String, Delimiter:String, ItemType:String)
```

**InputList**는 사용할 목록을 지정합니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [십표]

- [공백]
- [탭]



주:

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**ItemType**은 목록 멤버의 예상 항목 데이터 유형을 나타냅니다. 적합한 값은 `integer`, `float` 및 `datetime`입니다. 기본값은 `float`입니다.

예

```
MinList(1;2;3,[comma],Integer)
```

반환 값은 1입니다.

### Modulus

설명

지정된 두 정수를 나눈 값의 나머지를 반환합니다.

구문

```
Modulus(Dividend: Integer, Divisor: Integer): Integer
```

**Dividend**는 나뉘는 분수의 분자입니다.

**Divisor**는 나뉘는 분수의 분모입니다.

예

```
Modulus(5,2)
```

반환 값은 1입니다.

### Multiply

설명

지정된 두 정수를 곱한 후 결과를 반환합니다.

구문

```
Multiply(Int1: Integer, Int2: Integer): Integer
```

예

```
Multiply(2,5)
```

반환 값은 10입니다.

### **MultiplyFloat**

#### **설명**

지정된 두 개의 부동 소수점 수(부동)를 곱한 후 결과를 반환합니다.

#### **구문**

```
Multiply(Float1: Float, Float2: Float): Float
```

#### **예**

```
MultiplyFloat(4.76, 2.3)
```

반환 값은 10.948입니다.

### **NextSibling**

#### **설명**

현재 계층에 사용된 정렬 순서에 따라 현재 노드의 다음 동위 멤버를 반환됩니다.

이 함수는 로컬 범위로 제한되며 글로벌 컨텍스트에서 사용되는 경우 제대로 작동하지 않습니다.

#### **구문**

```
NextSibling(): String
```

#### **예**

```
NextSibling()
```

A, B, C 및 D가 Z의 하위 멤버이고 현재 노드가 B이면 반환 값은 C입니다.

### **NodeAccessGroups**

#### **설명**

현재 노드에 대한 현재 사용자의 노드 액세스 그룹을 심표로 구분된 목록으로 반환합니다.

이 함수는 로컬 범위로 제한되며 글로벌 컨텍스트에서 사용되는 경우 제대로 작동하지 않습니다.

#### **구문**

```
NodeAccessGroups(): String
```

#### **예**

```
NodeAccessGroups()
```

반환 값은 Accounts, Finance입니다.

### NodeExists

#### 설명

지정된 노드가 있는 경우 True를 반환합니다.

#### 구문

```
NodeExists(NodeAbbrev: string): Boolean
```

**NodeAbbrev**는 사용할 노드의 이름입니다.

#### 예

```
NodeExists(2000)
```

노드 2000이 있으면 반환 값은 True입니다.

### NodeInHier

#### 설명

지정된 계층에 지정된 노드가 있는 경우 True를 반환합니다.

#### 구문

```
NodeInHier(NodeAbbrev, HierAbbrev: string): Boolean
```

**NodeAbbrev**는 사용할 노드의 이름입니다.

**HierAbbrev**는 사용할 계층의 이름입니다.

#### 예

```
NodeInHier(2000, Assets)
```

Assets 계층에 노드 2000이 있으면 반환 값은 True입니다.

### NodeIsLeaf

#### 설명

현재 노드가 리프 노드인 경우 True를 반환합니다.

#### 구문

```
NodeIsLeaf(): Boolean
```

#### 예

```
NodeIsLeaf()
```

현재 노드가 리프이면 반환 값은 True입니다.

### **NodeIsValidForPropertyHiers**

#### **설명**

지정된 노드가 지정된 등록정보에 대한 계층 제약 조건을 충족하는 경우 True를 반환합니다. 등록정보가 노드 값을 저장하지 않거나 등록정보에 대한 제약 조건이 정의되지 않은 경우에도 True를 반환합니다.

로컬 등록정보가 매개변수로 전달되는 경우 이 함수는 로컬 범위로 제한될 수 있습니다.

#### **구문**

```
NodeIsValidForPropertyHiers(NodeAbbrev: String, PropAbbrev: String): Boolean
```

**NodeAbbrev**는 사용할 노드의 이름입니다.

**PropAbbrev**는 사용할 등록정보의 이름입니다.

### **NodePropValue**

#### **설명**

로컬 노드에 대한 현재 계층 또는 글로벌 노드에 대한 현재 버전의 지정된 노드에 대해 지정된 등록정보 값을 반환합니다.

로컬 등록정보가 매개변수로 전달되는 경우 이 함수는 로컬 범위로 제한될 수 있습니다.

#### **구문**

```
NodePropValue(NodeAbbrev: String, PropAbbrev: String): String
```

**NodeAbbrev**는 사용할 노드의 이름입니다.

**PropAbbrev**는 사용할 등록정보의 이름입니다.

#### **예**

```
NodePropValue(2000, Abbrev())
```

반환 값은 2000입니다.

### **Not**

#### **설명**

지정된 부울 표현식의 반대 부울을 반환합니다.

#### **구문**

```
Not(Expression: Boolean): Boolean
```

**예**

```
Not(NodeIsLeaf())
```

노드가 리프이면 반환 값은 True입니다.

**Now****설명**

현재 시스템 날짜 및/또는 시간을 반환합니다.

**구문**

```
Now([DateTimeType: String]): DateTime
```

**DateTimeType**은 반환할 데이터 부분을 지정하는 선택 사항입니다. 적합한 값은 Date, Time, Datetime입니다. 기본값은 Date입니다.

**예**

```
Now()
```

현재 날짜 및 시간을 반환합니다(예: 3/25/2010 9:20:44 AM).

```
Now(Time)
```

현재 시간만 반환합니다(예: 9:20:44 AM).

```
Now(Date)
```

현재 날짜만 반환합니다(예: 3/25/2010).

**NumChildWith****설명**

지정된 표현식이 True인 현재 노드의 하위 멤버 수를 반환합니다.

**구문**

```
NumChildWith(Expression: Boolean): Integer
```

**예**

```
NumChildWith(NodeIsLeaf())
```

노드에 두 개의 리프 하위 멤버가 있으면 반환 값은 2입니다.

## NumDescendantsWith

### 설명

지정된 표현식이 True인 현재 노드의 하위 멤버 수를 반환합니다.

### 구문

```
NumDescendantsWith(Expression: Boolean): Integer
```

### 예

```
NumDescendantsWith(NodeIsLeaf())
```

노드에 두 개의 하위 멤버가 있고 각 하위 멤버에 10개의 리프 하위 멤버가 있으면 반환 값은 20입니다.

## Or

### 설명

지정된 부울 표현식이 하나라도 True인 경우 True를 반환합니다.

### 구문

```
Or(Expression1, Expression2, ... ExpressionN: Boolean): Boolean
```

### 예

```
Or(NodeIsLeaf(), Equals(Integer, PropValue(Level), 3))
```

현재 노드가 리프이거나 계층의 레벨 3에 있으면 반환 값은 True입니다.

## OrigPropValue

### 설명

HasSiblingWith 또는 NumDescendantsWith 함수를 사용하는 경우 시작 노드에 대해 지정된 등록정보 값을 반환합니다.

로컬 등록정보가 매개변수로 전달되는 경우 이 함수는 로컬 범위로 제한될 수 있습니다.

### 구문

```
OrigPropValue(PropAbbrev: String): String
```

**PropAbbrev**는 사용할 등록정보의 이름입니다.

### 예

```
HasSiblingWith(GreaterThan(OrigPropValue(ID), ID()))
```

현재 노드의 ID가 200이고 노드 ID가 200보다 큰 동위 멤버가 있으면 반환 값은 True입니다.

## PadChar

### 설명

지정된 문자열을 지정된 채움 문자로 연장하여 반환합니다. 원본 문자열의 왼쪽이나 오른쪽에 채울 수 있습니다. 결과 문자열은 지정된 자릿수 이상입니다. 원본 문자열이 지정된 자릿수보다 긴 경우 원본 목록이 반환됩니다.

### 구문

```
PadChar(String: String, PadChar: String; PadLeft: Boolean; NewLength: Integer): String
```

**String**은 함수를 실행할 문자열 값입니다.

**PadChar**은 문자열 패딩에 사용할 문자입니다.

**PadLeft**는 왼쪽에서 문자열을 패딩할지 여부를 지정합니다. 적합한 값은 1, 0, T, F, t 또는 f입니다.

**NewLength**는 결과 길이를 지정하는 정수입니다.

### 예

```
PadChar(102,0,1,6)
```

반환 값은 000102입니다.

## PadList

### 설명

지정된 목록을 지정된 채움 문자로 연장하여 반환합니다. 원본 목록의 왼쪽이나 오른쪽에 채울 수 있습니다. 결과 목록은 지정된 자릿수 이상입니다. 원본 목록이 지정된 자릿수보다 긴 경우 원본 목록이 반환됩니다.

### 구문

```
PadList(String, DelimChar, PadChr:String, PadLeft: Boolean, NewLength:Integer): String
```

**StringList**는 지정된 구분자로 구분되는 패딩을 적용할 문자열 목록입니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [싹표]
- [공백]
- [탭]

 주:

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**PadChar**은 문자열 패딩에 사용할 문자입니다.

**PadLeft**는 왼쪽에서 문자열을 패딩할지 여부를 지정합니다. 적합한 값은 1, 0, T, F, t 또는 f입니다.

**NewLength**는 결과 길이를 지정하는 정수입니다.

## 예

```
PadList(1;2;3;4,;,T,3)
```

반환 값은 001;002;003,004입니다.

**ParentPropValue**

## 설명

현재 노드의 상위 노드에 대해 지정된 등록정보 값을 반환합니다. 노드에 상위 멤버가 없거나 현재 노드가 로컬 노드가 아닌 경우 빈 문자열을 반환합니다.

이 함수는 로컬 범위로 제한되며 글로벌 컨텍스트에서 사용되는 경우 제대로 작동하지 않습니다.

## 구문

```
ParentPropValue(PropAbbrev: String): String
```

**PropAbbrev**는 사용할 등록정보의 이름입니다.

## 예

```
ParentPropValue(Abbrev)
```

상위 노드 이름이 Colas이면 반환 값은 Colas입니다.

**Pos**

## 설명

대소문자 구분 검색을 사용하여 지정된 문자열 내에서 지정된 하위 문자열의 첫 번째 문자 위치 (인덱스)를 반환합니다. 문자열 값 내에서 하위 문자열을 찾을 수 없는 경우 0 값이 반환됩니다.

## 구문

```
Pos(SubString: String, String: String): Integer
```

**Substring**은 검색할 문자열 값입니다.

**String**은 함수를 실행할 문자열 값입니다.

**예**

```
Pos (D, ABCDEFG)
```

반환 값은 4입니다.

**PreviousSibling****설명**

현재 계층에 사용된 정렬 순서에 따라 현재 노드의 이전 동위 멤버를 반환됩니다.

이 함수는 로컬 범위로 제한되며 글로벌 컨텍스트에서 사용되는 경우 제대로 작동하지 않습니다.

**구문**

```
PreviousSibling(): String
```

**예**

```
PreviousSibling()
```

A, B, C 및 D가 Z의 하위 멤버이고 현재 노드가 B이면 반환 값은 A입니다.

**PropControllingHier****설명**

현재 버전에서 지정된 등록정보의 제어 계층 이름을 반환합니다.

**구문**

```
PropControllingHier(PropAbbrev: String): String
```

**PropAbbrev**는 사용할 등록정보의 이름입니다.

**예**

```
PropControllingHier(TimeBalance)
```

반환 값은 Accounts입니다.

**PropDefaultValue****설명**

지정된 등록정보 정의의 기본값을 반환합니다.

**구문**

```
PropDefaultValue(PropAbbrev: String): String
```

**PropAbbrev**는 사용할 등록정보의 이름입니다.

예

```
PropDefaultValue (Currency)
```

반환 값은 USD입니다.

### PropertyCategories

설명

현재 사용자에게 대한 등록정보 범주를 심표로 구분된 목록으로 반환합니다.

구문

```
PropertyCategories (AccessType: String) :String
```

**AccessType**은 등록정보 범주에 대한 액세스 레벨입니다. 적합한 값은 `ReadOnly`, `ReadWrite` 또는 `Both`입니다.

예

```
PropertyCategories (Both)
```

반환 값은 `System`, `All`, `Essbase`, `Enterprise`, `HFM`, `Planning`입니다.

### PropMaxValue

설명

지정된 등록정보 정의의 최대값을 반환합니다.

구문

```
PropMaxValue (PropAbbrev: String): Integer
```

**PropAbbrev**는 사용할 등록정보의 이름입니다.

예

```
PropMaxValue (Volume)
```

반환 값은 10입니다.

### PropMinValue

설명

지정된 등록정보 정의의 최소값을 반환합니다.

## 구문

```
PropMinValue(PropAbbrev: String): Integer
```

**PropAbbrev**는 사용할 등록정보의 이름입니다.

## 예

```
PropMinValue(Volume)
```

반환 값은 1입니다.

## PropValue

### 설명

현재 노드에 대해 지정된 등록정보 값을 반환합니다.

로컬 등록정보가 매개변수로 전달되는 경우 이 함수는 로컬 범위로 제한될 수 있습니다.

## 구문

```
PropValue(PropAbbrev: String): String
```

**PropAbbrev**는 사용할 등록정보의 이름입니다.

## 예

```
PropValue(Volume)
```

반환 값은 2입니다.

## RangeListContains

### 설명

지정된 범위 목록에 지정된 값이 포함된 경우 True를 반환합니다.

## 구문

```
RangeListContains(RangeList: String, Value: Integer, Delimiter:  
String): Boolean
```

**RangeList**는 지정된 구분자로 구분되는 검색할 정수 범위 목록입니다. 예: 1-100, 201-300

**Value**는 범위 목록에서 검색할 정수 값입니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [침표]
- [공백]

- [탭]



주:

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

예

```
RangeListContains(PropValue(MyRangeList), 1, [Comma])
```

MyRangeList 등록정보의 값이 1-10, 101-10000이면 지정된 범위에 1이 포함되므로 반환 값은 True입니다. 그러나 RangeListContains(PropValue(MyRangeList), 11, [Comma])는 11이 지정된 범위에 포함되지 않으므로 False를 반환합니다.



주:

MyRangeList를 "1-5,6-10,101-1000"으로 변경하면 Data Relationship Management는 RangeList를 확인하고 연속 경계의 범위를 결합하므로 이 값을 "1-10,101-1000"으로 바꿉니다.

## ReplacementAbbrev

설명

노드가 비활성 상태이고 병합 노드가 지정된 경우 현재 노드에 대한 대체(병합) 노드 이름을 반환합니다.

구문

```
ReplacementAbbrev(): String
```

예

```
ReplacementAbbrev()
```

## ReplacePropValue

설명

노드가 비활성 상태이고 병합 노드가 지정된 경우 현재 노드의 대체(병합) 노드에 대해 지정된 등록정보 값을 반환합니다.

로컬 등록정보가 매개변수로 전달되는 경우 이 함수는 로컬 범위로 제한될 수 있습니다.

구문

```
ReplacePropValue(PropAbbrev: String): String
```

**PropAbbrev**는 사용할 등록정보의 이름입니다.

예

```
ReplacePropValue (Description)
```

### ReplaceStr

설명

이전 패턴의 인스턴스를 새 패턴으로 바꾼 문자열을 반환합니다.

구문

```
ReplaceStr (String: String, OldPattern: String, NewPattern:  
String, ReplaceAll: Boolean): String
```

**String**은 함수를 실행할 문자열 값입니다.

**NewPattern**은 찾을 문자열을 대체할 문자열 값입니다.

**OldPattern**은 검색할 문자열 값입니다.

**ReplaceAll**은 발견된 모든 검색 문자열을 바꾸기 문자열로 바꿀지 여부를 지정합니다.  
적합한 값은 1, 0, T, F, t 또는 f입니다.

예

```
ReplaceStr (A1;A2;A3,A,B,T)
```

반환 값은 B1;B2;B3입니다.

### RTrim

설명

문자열 끝에서 모든 공백을 지우고 지정된 값을 반환합니다.

구문

```
RTrim (String: String): String
```

**String**은 함수를 실행할 문자열 값입니다.

예

```
RTrim ("100  ")
```

반환 값은 100입니다.

## SortList

### 설명

지정된 목록을 정렬된 순서로 반환합니다.

### 구문

```
SortList(InputList: String, Delimiter: String, IgnoreCase: Boolean, ItemType: String)
```

**InputList**는 사용할 목록을 지정합니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [십표]
- [공백]
- [탭]



주:

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**IgnoreCase**는 정렬할 때 대소문자를 무시할지 여부를 지정합니다. 기본값은 `False`입니다.

**ItemType**은 결과 목록 항목의 대상 데이터 유형을 나타냅니다. 적합한 값은 `string`, `integer`, `float`, `date`, `time` 및 `datetime`입니다. 기본값은 `string`입니다. 항목을 지정된 유형으로 변환할 수 없는 경우에는 함수에서 빈 문자열이 반환됩니다.

## StripPadChar

### 설명

지정된 문자열의 시작에서 지정된 채움 문자를 제거하고 수정된 값을 반환합니다. 원본 문자열에 포함된 채움 문자 수가 `StripCount`에 대해 지정된 개수보다 적은 경우 원본 문자열 값이 반환됩니다.

### 구문

```
StripPadChar(String: String, PadChar: String, StripCount: Integer): String
```

**String**은 함수를 실행할 문자열 값입니다.

**PadChar**은 문자열 패딩에 사용할 문자입니다.

**StripCount**는 문자열에서 제거할 문자 수를 지정하는 정수입니다. 0은 모든 패딩된 문자를 제거합니다.

**예**

```
StripPadChar(0003333,0,6)
```

반환 값은 3333입니다.

**StrToBool****설명**

지정된 문자열을 기준으로 부울 값을 반환합니다. 대소문자 또는 다음 문자에 관계없이 문자열이 Y, T 또는 1로 시작하는 경우 True 값이 반환됩니다. 대소문자 또는 다음 문자에 관계없이 문자열이 N, F 또는 0으로 시작하는 경우 False 값이 반환됩니다.

**구문**

```
StrToBool(String: String): Boolean
```

**String**은 함수를 실행할 문자열 값입니다.

**예**

```
StrToBool(0)
```

반환 값은 False입니다.

**StrToFloat****설명**

지정된 문자열의 부동 소수점 값을 반환합니다. 공백 또는 빈 문자열의 경우 0을 반환합니다. 지정된 문자열이 부동 소수점 수를 나타내지 않는 경우 오류가 반환됩니다.

**구문**

```
StrToFloat(String: String): Float
```

**String**은 함수를 실행할 문자열 값입니다.

**예**

```
StrToFloat(11.101)
```

반환 값은 11.101입니다.

**StrToInt****설명**

지정된 문자열의 정수 값을 반환합니다. 공백 또는 빈 문자열의 경우 0을 반환합니다.

지정된 문자열이 정수를 나타내지 않는 경우 오류가 반환됩니다.

### 구문

```
StrToInt(String: String): Integer
```

**String**은 함수를 실행할 문자열 값입니다.

### 예

```
StrToInt(101)
```

반환 값은 101입니다.

### Stuff

#### 설명

지정된 문자를 지정된 문자열로 바꾸어 지정된 값을 반환합니다.

### 구문

```
Stuff(PropAbbrev: String, CharsToReplace: String, ReplacementChars: String):  
String
```

**PropAbbrev**는 사용할 등록정보의 이름입니다.

**CharsToReplace**는 검색할 문자열 값입니다.

**ReplacementChars**는 찾을 문자열을 대체할 문자열 값입니다.

### 예

```
Stuff(Abbrev(), GEO, RIO)
```

Abbrev가 GEO101이면 반환 값은 RIO101입니다.

### SubString

#### 설명

지정된 인덱스부터 시작하여 지정된 문자 수를 포함하는 지정된 문자열 부분을 반환합니다.

### 구문

```
SubString(String: String, Index: Integer, Count: Integer): String
```

**SubString**은 함수를 실행할 문자열 값입니다.

**Index**는 하위 문자열 검색을 시작할 색인 위치를 나타내는 정수입니다. 0은 문자열의 첫 번째 문자 위치를 나타냅니다.

**Count**는 시작 색인부터 검색할 문자 수를 나타내는 숫자입니다.

**예**

```
SubString(Colas,1,2)
```

반환 값은 Co입니다.

**Subtract****설명**

첫 번째 값에서 두 번째 정수 값을 뺀 후 결과를 반환합니다.

**구문**

```
Subtract(Minuend: Integer,Subtrahend: Integer): Integer
```

**Minuend**는 정수 값입니다.

**Subtrahend**는 정수 값입니다.

**예**

```
Subtract(10,2)
```

반환 값은 8입니다.

**SubtractFloat****설명**

첫 번째 값에서 두 번째 부동 소수점 값을 뺀 후 결과를 반환합니다.

**구문**

```
SubtractFloat(Minuend,Subtrahend: Float): Float
```

**Minuend**는 부동 소수점 값입니다.

**Subtrahend**는 부동 소수점 값입니다.

**예**

```
SubtractFloat(8.09,3.76)
```

반환 값은 4.33입니다.

**SumList****설명**

빈 항목을 무시하고 목록에 있는 항목의 합계를 반환합니다. 지정된 항목 유형이 아닌 항목이 목록에 포함된 경우 빈 문자열을 반환합니다.

## 구문

```
SumList(InputList: String,Delimiter: String,ItemType: String):Integer
```

**InputList**는 사용할 목록을 지정합니다.

**Delimiter**는 문자열 목록에서 항목을 기술하는 데 사용할 문자입니다. 지원되는 특수 문자는 다음과 같습니다.

- [심표]
- [공백]
- [탭]



주:

문자가 아니라 구분자의 이름을 사용하고 대괄호로 이름을 묶어야 합니다.

**ItemType**은 목록 멤버의 예상 항목 데이터 유형을 나타냅니다. 적합한 값은 `integer` 및 `float`입니다. 기본값은 `float`입니다.

## 예

```
SumList(1;2;3,;,Integer)
```

반환 값은 6입니다.

## Trim

### 설명

문자열 시작 및 끝에서 모든 공백을 지우고 지정된 값을 반환합니다.

## 구문

```
Trim(String: String): String
```

**String**은 함수를 실행할 문자열 값입니다.

## 예

```
Trim(" 101 ")
```

반환 값은 101입니다.

## UpperCase

### 설명

문자열 값을 대문자로 변환하여 반환합니다.

### 구문

```
UpperCase(String: String): String
```

**String**은 함수를 실행할 문자열 값입니다.

### 예

```
UpperCase(smaller)
```

반환 값은 SMALLER입니다.

### UserName

#### 설명

현재 사용자의 사용자 이름을 반환합니다.

### 구문

```
UserName(): String
```

### 예

```
UserName()
```

반환 값은 사용자 이름입니다.

### XOr

#### 설명

지정된 부울 표현식 중 하나만 True인 경우 True를 반환합니다.

### 구문

```
XOr(Expression1:Boolean, Expression2: Boolean): Boolean
```

### 예

```
XOr(NodeIsLeaf(), Equals(Integer, PropValue(Level), 3))
```

노드가 리프이거나 계층의 레벨 3에 있으면 반환 값은 True입니다.

## 함수 그룹

다음 표에는 함수가 사용별로 그룹화되어 있습니다.

표 11-3 함수 그룹

함수 그룹	함수
집계	<ul style="list-style-type: none"> <li>• AvgList</li> <li>• MaxList</li> <li>• MinList</li> <li>• SumList</li> </ul>
변경 추적	<ul style="list-style-type: none"> <li>• AddedBy</li> <li>• AddedOn</li> <li>• Changed</li> <li>• ChangedBy</li> <li>• ChangedOn</li> <li>• Now</li> </ul>
비교	<ul style="list-style-type: none"> <li>• Equals</li> <li>• GreaterThan</li> <li>• GreaterThanOrEqual</li> <li>• InRange</li> <li>• IsBlank</li> <li>• IsRangeListSubset</li> <li>• LessThan</li> <li>• LessThanOrEqual</li> <li>• RangeListContains</li> </ul>
조건	<ul style="list-style-type: none"> <li>• And</li> <li>• If</li> <li>• Not</li> <li>• Or</li> <li>• XOr</li> </ul>
데이터 유형	<ul style="list-style-type: none"> <li>• BoolToStr</li> <li>• FloatToStr</li> <li>• IntToStr</li> <li>• IsDataType</li> <li>• IsNumeric</li> <li>• StrToBool</li> <li>• StrToFloat</li> <li>• StrToInt</li> </ul>
목록	<ul style="list-style-type: none"> <li>• ArrayCount</li> <li>• ArrayIndex</li> <li>• ArrayItem</li> <li>• Intersection</li> <li>• ListContains</li> <li>• ListDistinct</li> <li>• ListNodePropValues</li> <li>• ListNodesWith</li> <li>• SortList</li> </ul>

표 11-3 (계속) 함수 그룹

함수 그룹	함수
수치	<ul style="list-style-type: none"> <li>• Add</li> <li>• AddFloat</li> <li>• Divide</li> <li>• DivideFloat</li> <li>• Modulus</li> <li>• Multiply</li> <li>• MultiplyFloat</li> <li>• Subtract</li> <li>• SubtractFloat</li> </ul>
Node	<ul style="list-style-type: none"> <li>• Abbrev</li> <li>• ID</li> <li>• InternalPrefix</li> <li>• NodeExists</li> <li>• NodeInHier</li> <li>• NodeIsLeaf</li> </ul>
Property	<ul style="list-style-type: none"> <li>• AncestorProp</li> <li>• AscNodeProp</li> <li>• DefaultProp</li> <li>• Descr</li> <li>• DualAncestorProp</li> <li>• HierNodePropValue</li> <li>• InheritedPropOrigin</li> <li>• IsDefinedPropVal</li> <li>• NodePropValue</li> <li>• OrigPropValue</li> <li>• ParentPropValue</li> <li>• PropControllingHier</li> <li>• PropDefaultValue</li> <li>• PropMaxValue</li> <li>• PropMinValue</li> <li>• PropValue</li> <li>• ReplacePropValue</li> </ul>

표 11-3 (계속) 함수 그룹

함수 그룹	함수
관계	<ul style="list-style-type: none"> <li>• Children</li> <li>• HasChildWith</li> <li>• HasParentNode</li> <li>• HasSiblingWith</li> <li>• InvertedLevel</li> <li>• IsBottomNode</li> <li>• IsNodeAbove</li> <li>• IsNodeBelow</li> <li>• ListAncestors</li> <li>• ListChildren</li> <li>• ListDescendants</li> <li>• ListRelatedNodesWith</li> <li>• ListSiblings</li> <li>• NextSibling</li> <li>• NumChildWith</li> <li>• NumDescendantsWith</li> <li>• PreviousSibling</li> <li>• ReplacementAbbrev</li> </ul>
문자열 조작	<ul style="list-style-type: none"> <li>• Concat</li> <li>• ConcatWithDelimiter</li> <li>• Decode</li> <li>• FlipList</li> <li>• 형식</li> <li>• FormattedDate</li> <li>• HasCharacters</li> <li>• IsAlpha</li> <li>• IsAlphaNumeric</li> <li>• Length</li> <li>• LowerCase</li> <li>• LTrim</li> <li>• PadChar</li> <li>• PadList</li> <li>• Pos</li> <li>• ReplaceStr</li> <li>• RTrim</li> <li>• StripPadChar</li> <li>• Stuff</li> <li>• SubString</li> <li>• Trim</li> <li>• UpperCase</li> </ul>
사용자	<ul style="list-style-type: none"> <li>• NodeAccessGroups</li> <li>• PropertyCategories</li> <li>• UserName</li> </ul>

# 12

## 동적 스크립트 관리

동적 스크립트를 사용하면 JavaScript로 파생 등록정보 및 검증에 대한 비즈니스 논리를 개발할 수 있습니다. 동적 스크립트는 공식 대신에 사용할 수 있는 보다 강력하고 효과적인 방법으로 표준 스크립팅 언어를 사용합니다. 스크립트를 통해 여러 명령문, 변수 및 인라인 설명을 사용하면 보다 조직적이고 덜 복잡한 논리 구성이 가능합니다. 동적 스크립트는 루프, 정규 표현식 등의 고급 개념도 지원합니다.

### 실행 컨텍스트

여러 스크립트 실행 컨텍스트(등록정보 컨텍스트, 검증 컨텍스트 및 요청 항목 등록정보 컨텍스트)가 있습니다. 각 컨텍스트에서 다른 초기 매개변수를 정의하고 다른 유형의 결과를 반환합니다.

### 스크립트를 사용한 파생 등록정보

스크립트 `deriver` 클래스를 통해 파생 등록정보에서 동적 스크립트를 사용할 수 있습니다. 스크립트를 사용한 파생 등록정보는 버전, 계층 구조 및 노드에 사용할 수 있습니다.

표 12-1 등록정보 레벨 설명

등록정보 레벨	매개변수	객체
Version	version	VersionObject
계층	hierarchy	HierarchyObject
글로벌 노드	node	NodeObject
로컬 노드	node	LocalNodeObject

자세한 내용은 다음을 참조하십시오.

- [노드 파생 등록정보](#)
- [버전 및 계층 등록정보](#)

#### 노드 파생 등록정보

이 컨텍스트에서는 `node`라는 매개변수가 전달됩니다. 글로벌 등록정보의 경우 `node`는 `NodeObject`입니다. 로컬 등록정보의 경우 `node`는 `LocalNodeObject`입니다. 파생 등록정보의 스크립트에서 값을 반환해야 하고, 값이 평가 또는 실행되는 등록정보의 데이터 유형에 적합해야 합니다. 스크립트에서 반환된 값이 등록정보 데이터 유형과 일치하지 않는 경우 강제됩니다. 예를 들어 부울 등록정보에 대해 반환되는 `null` 값은 `false`로 처리됩니다.

#### 주:

모든 Oracle Data Relationship Management 등록정보 데이터 유형에 JavaScript 표현이 있는 것은 아닙니다. [데이터 유형 변환](#)을 참조하십시오.

### 버전 및 계층 등록정보

이 컨텍스트에서는 VersionObject를 참조하는 version 매개변수나 HierObject를 참조하는 hierarchy 매개변수를 사용합니다. 스크립트를 정의하는 경우 스크립트를 평가 또는 실행할 때 버전을 로드할 수 없습니다. version 또는 hierarchy 파생 등록정보가 다른 버전 및 계층 레벨 등록정보에만 액세스하는 경우 버전 로드 상태에 관계없이 등록정보가 계산됩니다. version 또는 hierarchy 파생 등록정보가 노드 레벨 정보에 액세스하는 경우 버전이 로드되어 있어야 합니다. 그렇지 않으면 등록정보를 계산할 때 오류 값이 생성됩니다. 예를 들어 버전 레벨 등록정보가 단독 목록을 가져오는 경우 버전이 로드되어 있지 않으면 해당 등록정보에서 오류 값이 생성됩니다. 버전이 로드된 후에는 동일한 등록정보에서 정확한 값이 생성됩니다.

## 스크립트를 사용한 검증

스크립트 검증 클래스를 통해 동적 스크립트를 검증과 함께 사용할 수 있습니다. 여러 가지 검증 레벨이 있으며 일부 레벨에는 다른 매개변수가 있습니다. 다음은 검증 레벨 및 매개변수입니다.

표 12-2 검증 레벨 및 매개변수

레벨	매개변수	설명
임의 레벨	validation	현재 실행 중인 검증에 대한 정보를 제공합니다.
계층	hierarchy	검증되는 계층의 HierarchyObject
GlobalNode	node	검증되는 글로벌 노드의 NodeObject
Node	node	검증되는 노드의 LocalNodeObject
제거	node	검증되는 노드의 NodeObject
이동	node	이동되는 노드의 LocalNodeObject
	move	이동에 대한 정보가 포함된 객체: OldParent - 원본 상위 멤버의 LocalNodeObject NewParent - 대상 상위 멤버의 LocalNodeObject IsPost/IsPre - 이 스크립트가 이동 직전에 실행되는지, 아니면 이동이 완료된 직후에 실행되는지를 나타냅니다. 일반적으로 스크립트는 두 번(이동 전에 한 번, 이동 후에 한 번) 실행됩니다. Values - 이동 전 단계에서 단순 키-값 쌍을 이 객체에 저장할 수 있습니다(예: Values["key"] = "value"). 이동 후 단계에는 이러한 값이 있으므로 이동 전 상태에 대한 정보를 저장하고 이동 후 상태와 비교할 수 있습니다. 모든 값이 String, Number 또는 Date 객체로 변환됩니다. 복합 객체는 현재 지원되지 않습니다.
병합	node	삭제 또는 비활성화되는 노드

표 12-2 (계속) 검증 레벨 및 매개변수

레벨	매개변수	설명
	merge	병합에 대한 정보가 포함된 객체: Target - 병합 대상의 NodeObject IsInactivate - 비활성 작업인 경우 True IsDelete - 삭제 작업인 경우 True
Version	version	검증되는 버전의 VersionObject

## 스크립트를 사용한 관리 요청

동적 스크립트는 관리 요청의 워크플로우 태스크에 사용할 수 있습니다. 스크립트는 현재 요청 항목의 컨텍스트에서 실행되며 업데이트되는 노드의 이름 또는 상위 멤버 같이 항목에서 사용되는 값을 계산하는 데 사용됩니다.

표 12-3 관리 매개변수

매개변수	설명
requestitem	계산되는 요청에 대한 현재 RequestItemObject

## 열거 상수

특정 등록정보는 지정된 상수에 해당하는 숫자이며 코드를 더 쉽게 이해하고 유지 관리할 수 있게 만듭니다. 예를 들어

```
if(nodeProp.PropOrigin == 2)를 사용하는 대신 if(nodeProp.PropOrigin == PropOrigin.Overridden)을 사용할 수 있습니다.
```

### 등록정보 열거 상수

- DataType - Boolean, LeafNode, Date, Time, Float, Integer, Sort, Group, Node, LimbNode, String, Hier, Version, ListGroup, MultiNode, AscNode, AscNodes, AscGroup, Memo, FormatMemo, SortProp, Property, Query, StdQuery, GlobalNode, NodeProps, RangeList, DateTime, Hyperlink, HierarchyGroup
- PropLevel - Node, Hier, Version
- PropOrigin - Default, Inherited, Overridden, InheritedHier, InheritedVer, Derived, InheritedDomain, Unknown
- PropType - Invalid, System, Defined, Lookup, Derived, Stats, Validation, Verification, LimbAccessGroup, LeafAccessGroup, UserSpecific, RWDerived, SharedInfo

### 검증 열거 상수

- ValidationLevel - Node, Hier, Version, GlobalNodes, Merge, Move, Remove
- ValidationType - None, RealTime, Batch, Both

### 요청 열거 상수

- WorkflowAction - AddLeaf, AddLimb, 업데이트, 비활성, 삽입, 이동, 제거, 삭제
- WorkflowStageType - 제출, 강화, 승인, 커밋
- WorkflowStatus - 없음, 초안, 제출됨, 계산됨, 검증됨, 반송됨, 대기 중, 지정됨, 클레임됨, 상향 조정됨, 하향 조정됨, 거부됨, 커밋됨

#### 주:

WorkflowStatus 열거는 현재 요청 값인 RequestObject.Status를 반환하는데 사용됩니다. 하지만 일부 값은 내부적으로만 사용됩니다. RequestObject.Status에 대한 적합한 값은 초안, 제출됨, 대기 중, 클레임됨, 상향 조정됨, 거부됨 또는 커밋됨입니다.

## 지원되는 JavaScript 데이터 유형

표준 JavaScript 데이터 유형을 사용할 수 있으며, 가능한 경우 Oracle Data Relationship Management는 항상 표준 JavaScript 데이터 유형을 사용합니다. 예를 들어 날짜는 Date 객체를 사용하여 나타냅니다. 함수 자체는 객체이며, *new*로 호출된 함수는 프로토타입이 ECMA 규격 JavaScript 환경과 마찬가지로 함수의 구성자 프로토타입을 가리키는 객체를 생성합니다.

#### 주:

JavaScript DOM(문서 객체 모델) 객체는 Data Relationship Management 스크립트에서 지원되지 않습니다.

사용 가능한 메소드를 포함하여 JavaScript 구문과 기본 제공 객체에 대해 잘 알고 있어야 합니다. 사용 가능한 데이터 유형 중 일부는 다음과 같습니다.

- Array - length, pop, push, concat, join, reverse, slice, shift, sort 등을 포함합니다.

#### 주:

캐싱 메커니즘에 따라 항목의 JavaScript boxing이 변경되었으므로 일부 배열 함수가 예상대로 작동하지 않거나 이전 릴리스에서처럼 작동하지 않습니다. 예를 들어 JavaScript의 indexOf는 항목의 문자열이나 텍스트 값이 아닌 메모리 위치를 기반으로 객체를 비교합니다. 따라서 배열을 검사할 때 다른 방법을 고려해야 합니다. indexOf()는 JavaScript에서 "===" 비교를 사용하며 사용 가능한 단일 정의의 "=="가 없습니다. JavaScript 디자인 패턴을 사용하여 사용자 고유의 specialIndexOf()를 구현함으로써 "==" 형식의 비교를 제공할 수 있습니다.

- Boolean - True 및 False를 나타냅니다.
- Date—Date.parse(), month, day, year 등을 포함합니다.

- Error - try/catch 오류 처리를 사용하고 error.message에 액세스합니다.
- Function - 표준 호출 및 적용 기능을 지원합니다.
- Math - random, max, pow, round, sin, cos, floor, sqrt, log 등을 포함합니다.
- Number - JavaScript의 모든 숫자는 부동 소수점 유형 숫자입니다.
- RegExp - 정규 표현식에 대한 언어 지원을 사용하거나 명시적으로 액세스할 수 있습니다.
- String - concat, indexOf, lastIndexOf, substr, split, splice, search, replace, toUpperCase, toLowerCase 등을 포함합니다.

parseInt, parseFloat, isNaN, decodeURI, encodeURI 등의 글로벌로 사용 가능한 함수를 사용할 수도 있습니다.

### Print 함수

print 함수를 사용하면 스크립트를 생성하는 동안 디버그 정보를 출력할 수 있습니다. 결과는 스크립트 편집기의 [경고] 섹션에 표시됩니다. print 함수는 테스트 컨텍스트에서만 출력을 생성하지만 엔진에서 인수를 구성해야 합니다. 따라서 프로덕션 용도로 스크립트를 저장하기 전에 print 문을 모두 주석 처리합니다.

### Format 함수

Format 함수는 표준 JavaScript보다 훨씬 풍부한 문자열 형식 지정 메커니즘을 제공합니다. 첫 번째 매개변수는 중괄호 안에 형식 지정자를 포함하는 문자열입니다. 중괄호를 이스케이프하려면 두 개를 사용합니다. 예를 들어 "{{"는 출력에서 "{"가 됩니다. 형식 지정자는 0부터 시작되며 증분 방식으로 증가합니다. 순서에서 지정자를 생략하면 Format 함수의 동등한 매개변수가 무시됩니다. 예를 들어 "{1}"은(는) Format의 첫 번째 값 매개변수를 무시하고 두 번째 값 매개변수를 사용합니다.

한 가지 손쉬운 방법이 있습니다. Format을 호출하고 중괄호 없이 형식 지정자를 전달할 수 있습니다. 하나의 인수만 전달할 수 있습니다. 결과는 Format("{0:<specifier>", <argument>)와 같습니다.

형식 지정자는 Java 또는 C# 같은 다른 언어와 유사하게 작동합니다. 구문은 {<paramnum>} 또는 {<paramnum>;<format>}입니다. 여기서 paramnum은 0부터 시작해서 순차적으로 증가하는 양의 정수입니다. format 매개변수는 해당 매개변수로 전달된 객체의 유형에 따라 달라집니다.

일반적으로 format 매개변수는 사용자 로케일에 적합한 값을 반환합니다. 예를 들어 미국에서는 "{0:0.00}"이 "1.23"을 반환하지만 유럽에서는 "1,23"을 반환합니다. 또는 이스케이프 지원을 사용하여 로케일을 명시적으로 재정의하고 모든 사용자에게 대해 동일한 값을 출력할 수 있습니다. 예를 들어 "#,###,##0"은 culture 설정에 관계없이 모든 지역에서 십표를 천단위 구분자로 사용하여 숫자 형식을 지정합니다.

## 데이터 유형 변환

모든 Oracle Data Relationship Management 등록정보 정의 데이터 유형에 해당하는 표현이 JavaScript에 있는 것은 아닙니다. 해당 표현이 없는 경우 StringValue와 Value는 같으며, 문자열 값을 구문분석하는 방법을 이해해야 합니다. 이러한 데이터 유형 중 하나의 등록정보에 대해 값을 반환하는 경우 해당 데이터 유형의 적절한 문자열 표현도 반환해야 합니다. 저장된 값에 등록정보 데이터 유형에 적합한 변환이 없는 경우 값이 정의되지 않습니다.

목록 등록정보는 데이터 유형에 적합한 유형의 객체를 포함하는 배열의 각 요소가 포함된 배열을 반환합니다. 예를 들어 목록으로 표시된 Date 등록정보는 Date 객체가 포함된 배열을 반환합니다.

lookup 대상이 부적합하거나, lookup 테이블에 키가 없거나, lookup 테이블의 키가 데이터 유형에 적합하지 않을 경우 lookup 등록정보가 필요한 데이터 유형을 반환하지 않을 수도 있습니다. 예를 들어 키-값 쌍의 값이 "TEST"이지만 데이터 유형이 Date인 경우 결과는 정의되지 않음이 됩니다.

다음은 Data Relationship Management 데이터 유형과 해당 JavaScript 표현입니다.

**표 12-4 데이터 유형 비교**

등록정보 정의 데이터 유형	JavaScript 데이터 유형
AscGroup	NodeObject 배열
AscNode	NodeObject
AscNodes	NodeObject 배열
Boolean	Boolean
Date	Date
DateTime	DateTime
Float	Number
FormatMemo	String
GlobalNode	NodeObject
Group	String 배열
Hier	HierObject
Hierarchy Group	String(계층 그룹 이름)
Hyperlink	String(URL을 나타냄)
Integer	Number
LeafNode	LocalNodeObject
LimbNode	LocalNodeObject
ListGroup	String 배열
Memo	String
MultiNode	LocalNodeObject 배열
Node	LocalNodeObject
NodeProps	PropDefObject 배열
Query	String(쿼리 이름)
Property	PropDefObject
Sort	Number
SortProp	PropDefObject
StdQuery	String(쿼리 이름)
String	String
Time	String
Version	String(버전 이름)

다른 JavaScript 파생 등록정보(또는 다른 노드의 파생 등록정보)를 호출하는 경우 `deriver`에서 반환된 값이 문자열 표현으로 즉시 변환되지 않으므로 배열 등에서 기본 제공 변환이 명시된 경우를 제외하고 복합 객체에서 `toString()`을 호출하여 최종 결과가 반환될 때까지 `deriver`와 지연 강제 간에 복합 객체를 전달할 수 있습니다.

## 숫자 형식 지정

단일 바로가기 문자(예: "G") 또는 지정자 조합(예: "##0,000.0")으로만 숫자 형식을 지정할 수 있습니다. 형식 지정자에 1자보다 긴 바로가기 문자를 사용하면 출력에 변경되지 않은 상태로 복사됩니다(리터럴 문자로 처리됨).

출력 형식이 정확하게 지정되려면 적절한 culture를 선택하여 프로덕션 익스포트를 실행합니다.

**표 12-5** 단일 문자 바로가기 숫자 형식

형식	설명
D	정수(음수의 경우 로케일 인식 음수 기호 포함)
D<precision>	필요한 경우 0으로 채워 <precision> 자릿수 이상으로 형식이 지정된 정수. 예를 들어 "{0:D5}"를 사용할 경우 123이 12300으로 출력됩니다.
E	지수(과학적) 표기법 "1.234E+10"
F	부동 소수점 수 "123.456"(로케일 인식 소수 구분자 및 음수의 경우 음수 기호 포함)
F<precision>	소수점 뒤 <precision> 유효 자릿수로 반올림된 부동 소수점 수
G	일반 숫자 형식
N	일반 숫자 형식 "123,456.789"(로케일 인식 그룹/소수 구분자 및 음수의 경우 음수 기호 포함)
N<precision>	소수점 뒤 <precision> 자릿수로 반올림된 일반화된 숫자
P	퍼센트. 0.20146의 경우 "20.14%"로 출력됩니다 (로케일 인식 그룹/소수 구분자 및 음수의 경우 음수 기호 포함).
P<precision>	<precision> 유효 자릿수로 반올림된 퍼센트(0.205 "{0:P0}"의 경우 "21%"로 출력됨)
X	16진수(base-16) 출력 "4D2"

**표 12-6** 숫자 형식 지정자

형식	설명
0	0 자리 표시자. 있으면 숫자를 출력하고 그렇지 않으면 0을 출력합니다.
#	숫자 자리 표시자. 있으면 숫자를 출력하고 그렇지 않으면 출력을 생성하지 않습니다.
.	로케일 특정 소수 구분자
,	자리 표시자 사이에 배치하면 로케일 특정 그룹 구분자를 출력합니다(123456789 "{0:#,}"의 경우 "123,456,789"가 출력됨). 소수점(또는 암시적 소수점) 바로 왼쪽에 하나 이상 배치하면 숫자가 십표마다 1000으로 나뉩니다(123456789 "{0:#,##0,}"의 경우 "1,235"가 출력됨).

표 12-6 (계속) 숫자 형식 지정자

형식	설명
%	숫자에 100을 곱하고 지정된 위치에 로케일 특정 백분율 기호를 출력합니다.
E<sign>0	지수 표기법. 하나 이상의 0이 필요하며, 0의 개수는 지수의 최소 숫자를 지정합니다. <sign>은 선택 사항이며 다음과 같을 수 있습니다. <ul style="list-style-type: none"> <li>• + (항상 기호 +/-를 필수로 출력함)</li> <li>• - (음수의 경우에만 - 기호를 출력함)</li> </ul>
\<char>	이스케이프 문자(<char>이 리터럴 출력으로 처리됨)
;	섹션 구분자. 있으면 양수, 음수 및 0에 대해 다른 형식의 정의가 가능합니다. <ul style="list-style-type: none"> <li>• 1개 섹션 "{0:##,}" - 섹션 없음과 같습니다.</li> <li>• 2개 섹션 "{0:##;#;0}" - 첫 번째 섹션은 양수와 0에 적용되고, 두 번째 섹션은 음수에 적용됩니다.</li> <li>• 3개 섹션 "{0:##;#;0}" - 첫 번째 섹션은 양수와 0에 적용되고, 두 번째 섹션은 음수에 적용되고 (비워 두면 첫 번째 섹션이 음수에도 사용됨), 세 번째 섹션은 0에 적용됩니다.</li> </ul>
다른 임의 문자	변경되지 않고 출력에 복사됩니다.

## 날짜 형식 지정

단일 바로가기 문자(예: "G") 또는 지정자 조합(예: "HH:mm")으로 날짜 형식을 지정할 수 있습니다. 단일 문자를 바로가기가 아니라 일반 지정자로 사용하려는 경우 문자열 앞에 %를 추가합니다. 예를 들어 "%m"은 월+일이 아니라 채워지지 않은 분을 출력합니다.

표 12-7 단일 문자 바로가기 날짜 형식

형식	설명
t	간단한 시간 "4:05 PM"
T	자세한 시간 "4:05:07 PM"
d	간단한 날짜 "3/9/2013"
D	자세한 날짜 "Friday, March 09, 2013"
f	자세한 날짜 + 간단한 시간 "Friday, March 09, 2013 4:05 PM"
F	자세한 날짜 + 자세한 시간 "Friday, March 09, 2013 4:05:07 PM"
g	간단한 날짜 + 간단한 시간 "3/9/2013 4:05 PM"
G	간단한 날짜 + 자세한 시간 "3/9/2013 4:05:07 PM"(기본값)
m	월 + 일 "March 09"
y	월 + 연도 "March, 2013"

표 12-7 (계속) 단일 문자 바로가기 날짜 형식

형식	설명
r	RFC 1123 "Fri, 09 Mar 2013 16:05:07 GMT"
s	정렬 가능한 날짜/시간 "2013-03-09T16:05:07"
u	범용 정렬 가능한 날짜/시간 "2013-03-09 16:05:07Z"

표 12-8 날짜 형식 지정자(2자 이상)

형식	설명 예제는 2013-04-05 04:07:09 PM CST의 경우입니다.
yy	연도 "13"
yyyy	연도 "2013"
M	월 "4"
MM	월 "04"
MMM	월 "Apr"
MMMM	월 "April"
d	일 "5"
dd	일 "05"
ddd	요일 "Sun"
dddd	요일 "Sunday"
h	12시간 "4"
hh	12시간 "04"
H	24시간 "16"(4 AM의 경우 "4")
HH	24시간 "16"(4 AM의 경우 "04")
m	분 "7"
MM	분 "07"
s	초 "9"
ss	초 "09"
f	1초의 분수(추가 정밀도를 위해 1-4회 반복할 수 있음)
F	후행 0이 없는 1초의 분수(1-4회 반복할 수 있음)
t	AM 또는 PM 지정자 "P"(24시간 전용 culture의 경우 비어 있음)
tt	AM 또는 PM 지정자 "PM"(24시간 전용 culture의 경우 비어 있음)
Z	GMT 오프셋 "-6"
ZZ	GMT 오프셋 "-06"
ZZZ	GMT 오프셋 "-06:00"

표 12-8 (계속) 날짜 형식 지정자(2자 이상)

형식	설명 예제는 2013-04-05 04:07:09 PM CST의 경우입니다.
:	시간 구분자(로케일 특정)
/	날짜 구분자(로케일 특정)
\<char>	이스케이프 문자(<char>이 리터럴 출력으로 처리됨). 예를 들어 "{0:HH\h}"는 "16h"를 출력합니다.
다른 임의 문자	변경되지 않고 출력에 복사됩니다.

## Data Relationship Management 객체

다음은 Oracle Data Relationship Management 객체와 메서드 및 등록정보 설명입니다.

### SysObject

Sys라는 SysObject 한 개가 자동으로 생성됩니다. 이 객체는 모든 컨텍스트에서 사용할 수 있으며 일반 함수와 Data Relationship Management 애플리케이션에 대한 정보를 제공합니다. 이 객체에 대한 등록정보는 없습니다.

표 12-9 SysObject 메서드

이름	설명
FormattedDate (value, formatString)	Formula 시스템 규칙에 따라 날짜 형식을 지정합니다. 이전 Formula 등록정보와 정확하게 일치시키기 위한 이전 버전과의 호환성에 유용합니다. <ul style="list-style-type: none"> <li>value는 날짜 객체 또는 적합한 날짜/시간 문자열이어야 합니다.</li> <li>formatString은 적합한 형식 지정 문자열이어야 합니다(<a href="#">FormattedDate</a> 함수 참조).</li> </ul>
GetNextID(key)	지정된 문자열 키 값에 사용 가능한 다음 정수 ID를 반환합니다.
GetPropDef(abbrev)	지정된 등록정보 이름에 대한 PropDefObject를 반환합니다. 이름은 전체 이름이어야 합니다.
GetRequestByID(int)	ID별 워크플로우 요청을 반환합니다.
GetSysPrefValue(abbrev)	지정된 시스템 환경설정(예: HierNodeSeparator)의 값을 반환합니다.
InRange(dataType, input, min, max, minExclusive, maxExclusive)	공식 함수 <a href="#">InRange</a> 에 해당합니다. 필수 매개변수는 dataType, input 및 min입니다.
IsNodeAbove(ancestor, child)	계층에서 상위 멤버가 하위 멤버 위에 있는 경우 True를 반환합니다. 매개변수가 LocalNodeObjects가 아니거나 동일한 계층에 없는 경우 False를 반환합니다.

표 12-9 (계속) SysObject 메소드

이름	설명
IsNodeBelow(descendant, parent)	계층에서 하위 멤버가 상위 멤버 아래에 있는 경우 True를 반환합니다. 매개변수가 LocalNodeObjects가 아니거나 동일한 계층에 없는 경우 False를 반환합니다.
RunFormula(node, propDef, formulaString)	<p>Data Relationship Management 공식을 실행하고 문자열 결과를 반환합니다.</p> <ul style="list-style-type: none"> <li>node는 NodeObject 또는 LocalNodeObject입니다. NodeObject를 전달할 때 공식 문자열에서 로컬 등록정보를 참조하면 안 됩니다. 그렇지 않으면 오류가 발생합니다. LocalNodeObject를 전달할 때 사용 가능한 글로벌 및 로컬 등록정보를 모두 참조할 수 있습니다.</li> <li>propDef - 일부 공식 함수의 경우 정확하게 구문분석하거나 실행하려면 등록정보 정의가 필요합니다. 이러한 함수를 사용할 때는 등록정보 정의를 제공해야 합니다. 일반적으로 레벨, 글로벌 및 로컬, 유형 등의 등록정보 정의 특성이 일치해야 하지만 formulaString의 실제 등록정보일 필요는 없습니다. 관련이 없어도 됩니다. 대부분의 공식에서 이 매개변수에 대해 null을 전달할 수 있습니다. 구문은 <code>Sys.GetPropDef (abbrev)</code> 입니다. 예를 들면 다음과 같습니다. <pre> Sys.RunFormula (node, Sys.GetPropDef ("Custom.MyProp1"), "Concat (Prop value ', PropValue (Custom.MyProp2), ' , is, ,valid)"); </pre> </li> <li>formulaString은 레거시 Data Relationship Management 공식입니다. 공백은 공식의 리터럴 부분으로 간주되므로 필요한 경우 제거해야 합니다. <p><b>참고:</b> 모범 사례로 간주되지는 않으며, 레거시 동작과 정확히 일치시키기 위해 필요한 경우에만 사용해야 합니다. 이 메소드를 사용하면 성능이 저하됩니다.</p> </li> </ul>

### PropDefObject

이 객체에 대한 메소드는 없습니다.

표 12-10 PropDefObject 등록정보

이름	설명
Abbrev	등록정보 정의 이름(전체 네임스페이스 포함)

**표 12-10 (계속) PropDefObject 등록정보**

이름	설명
Cascade	등록정보 값이 상속되는 경우 True
ColumnWidth	기본 익스포트 열 너비
DataType	DataType 열거 값. 예: DataType.String( <a href="#">열거 상수 참조</a> )
Descr	설명
DefaultValue	등록정보 정의의 기본값 유형은 등록정보 정의의 데이터 유형에 따라 달라집니다.
EditorLabel	레이블
Global	등록정보가 글로벌 노드 등록정보인 경우 True
Hidden	등록정보가 등록정보 그리드에서 숨겨지는 경우 True
ID	ID
레벨	PropLevel 열거 값. 예: PropLevel.Node( <a href="#">열거 상수 참조</a> )
목록	등록 정보에서 사용자가 값 목록에서 선택하도록 허용하는 경우 True
ListValues	사용자가 선택할 수 있는 값 배열
LookupValues	lookup 등록정보에 대한 lookup 키-값 쌍. 이 배열에는 객체의 Key 및 Value 등록정보를 사용합니다.
MaxValue	최대값
MinValue	최소값
네임스페이스	등록정보 정의의 네임스페이스
PropType	PropType 열거 값. 예: PropType.Defined( <a href="#">열거 상수 참조</a> )
PropClass	Deriver 클래스(공식 또는 스크립트)
ReadOnly	등록정보가 읽기 전용인 경우 True(예: 핵심 통계 등록정보)

**VersionObject**

**표 12-11 VersionObject 등록정보**

이름	설명
Abbrev	이름
Descr	설명
HierCount	계층 구조 수
ID	ID
NodeCount	노드 수

**표 12-12 VersionObject 메소드**

이름	설명
GetHierarchies()	버전에서 현재 사용자가 사용할 수 있는 모든 계층 구조 배열을 가져옵니다.
GetGlobalNodes()	버전에서 모든 글로벌 노드(NodeObjects) 배열을 가져옵니다.
GetOrphans()	버전에서 모든 단독(NodeObjects) 배열을 가져옵니다.
HierByAbbrev(abbrev)	이름으로 HierarchyObject를 가져옵니다.
HierByID(id)	ID로 HierarchyObject를 가져옵니다.
NodeByAbbrev(abbrev)	이름으로 NodeObject를 가져옵니다.
NodeByID(id)	ID로 NodeObject를 가져옵니다.
NodeExists(abbrev)	지정된 이름을 가진 글로벌 노드가 있는 경우 True를 반환합니다.
Prop(abbrev)	버전의 지정된 등록정보에 대한 NodePropObject를 가져옵니다.
PropValue(abbrev)	버전의 지정된 등록정보 값을 가져옵니다. 반환 유형은 등록정보 정의의 데이터 유형에 따라 달라집니다.

### HierarchyObject

**표 12-13 HierarchyObject 등록정보**

이름	설명
Abbrev	이름
Descr	설명
HierarchyUrl	계층 URL
ID	ID
NodeCount	계층의 노드 수
SharedNodesEnabled	공유 노드가 활성화된 경우 True
TopNode	LocalNodeObject 최상위 노드
Version	VersionObject
VersionAbbrev	버전 이름
VersionID	버전 ID

**표 12-14 HierarchyObject 메소드**

이름	설명
NodeByAbbrev(abbrev)	이름으로 NodeObject를 가져옵니다.
NodeByID(id)	ID로 NodeObject를 가져옵니다.

표 12-14 (계속) HierarchyObject 메소드

이름	설명
NodeExists(abbrev)	지정된 이름을 가진 로컬 노드가 있는 경우 True를 반환합니다.
Prop(abbrev)	버전의 지정된 등록정보에 대한 NodePropObject를 가져옵니다.
PropValue(abbrev)	버전의 지정된 등록정보 값을 가져옵니다. 반환 유형은 등록정보 정의의 데이터 유형에 따라 달라집니다.

공통 노드 등록정보 및 메소드

일부 등록정보와 메소드는 NodeObject 및 LocalNodeObject에 공통적으로 적용되지만 두 객체는 프로토타입 체인을 공유하지 않습니다.

글로벌 또는 로컬 컨텍스트 때문에 값이 다를 수 있는 모든 경우에서 해당 컨텍스트에 맞는 정확한 값이 반환됩니다. 예를 들어 NodeObject에서 GetChildren()을 호출하는 경우 결과 배열에 NodeObjects가 포함됩니다. LocalNodeObject에서 동일한 호출을 수행할 때는 결과 배열에 LocalNodeObjects가 포함됩니다.

표 12-15 NodeObject 및 LocalNodeObject의 공통 등록정보

이름	설명
Abbrev	Core.Abbrev
AddedBy	Core.AddedBy
AddedOn	Core.AddedOn
Changed	Core.Changed
ChangedBy	Core.ChangedBy
ChangedOn	Core.ChangedOn
ChildNodeCount	하위 노드 수
Descr	Core.Descr
DomainAbbrev	Core.DomainAbbrev
DomainNodeAbbrev	Core.DomainNodeAbbrev
ID	Core.ID
Inactive	Core.Inactive
IsPrimary	노드가 공유 노드의 기본 노드인 경우 True, 노드가 공유되지 않거나 기본 노드가 아닌 경우 False
IsShared	노드가 공유 노드인 경우 True
Leaf	Core.Leaf
NodeApproved	Core.NodeApproved
Version	노드의 소유자 VersionObject
VersionAbbrev	노드의 버전 이름
VersionID	노드의 버전 ID

표 12-16 NodeObject 및 LocalNodeObject의 공통 메소드

이름	설명
GetChildren(sorted)	선택적으로 정렬된 순서로 이 노드의 직접 하위 멤버 배열을 가져옵니다. sorted의 기본값은 False입니다.
GetDescendants(inclusive, sorted)	선택적으로 이 노드를 포함하여 정렬된 순서로 이 노드의 하위 멤버 배열을 가져옵니다. inclusive의 기본값은 True입니다. sorted의 기본값은 False입니다.
NodeByAbbrev(abbrev)	이름으로 NodeObject를 가져옵니다.
NodeByID(id)	ID로 NodeObject를 가져옵니다.
NodeExists(abbrev)	지정된 이름을 가진 글로벌 노드가 있는 경우 True를 반환합니다.
Prop(abbrev)	버전의 지정된 등록정보에 대한 NodePropObject를 가져옵니다.
PropValue(abbrev)	버전의 지정된 등록정보 값을 가져옵니다. 반환 유형은 등록정보 정의의 데이터 유형에 따라 달라집니다.

### LocalNodeObject

다양한 *xxxWith* 함수를 사용하여 계층에서 다른 노드를 찾는 것이 좋습니다. 예를 들어 *ChildrenWith*는 *GetChildren()*을 호출하고 결과를 반복하는 것보다 훨씬 빨리 실행됩니다. 마찬가지로, *GetReferenceInHier*은 *GetReferences()*를 호출하고 결과를 반복하는 것보다 훨씬 더 빠르고 사용이 용이합니다.

표 12-17 LocalNodeObject 등록정보

이름	설명
GlobalNode	현재 노드의 글로벌 NodeObject
Hier	노드가 있는 계층의 HierarchyObject
HierAbbrev	Core.HierAbbrev
HierID	Core.HierID
레벨	계층의 노드 레벨을 나타내는 숫자
MissingPrimary	기본 노드를 찾을 수 없는 경우 True
NodeUrl	노드 URL
Parent	이 노드의 상위 노드에 대한 LocalNodeObject. 계층의 최상위 노드에 대해서는 Null이 반환됩니다.
ParentNodeAbbrev	상위 노드 이름
Primary	이 공유 노드의 기본 노드. 기본 노드가 이 계층에 없는 경우 발생하는 첫 번째 계층의 기본 노드를 반환합니다. 기본 노드가 나타나는 계층 구조 목록이 필요한 경우 반환된 기본 노드에 대해 <i>GetReferences()</i> 를 호출합니다. 공유 노드나 기본 노드를 찾을 수 없는 경우 null을 반환합니다.

표 12-17 (계속) LocalNodeObject 등록정보

이름	설명
PrimaryNotInHier	기본 노드가 있지만 이 계층에 없는 경우 True

표 12-18 LocalNodeObject 메소드

이름	설명
AncestorsWith(func, maxResults, searchFromTop, inclusive)	<p>상위 멤버 체인에서 지정된 함수를 충족하는 노드를 검색합니다. 상위 멤버를 찾는 가장 빠른 방법입니다. LocalNodeObject 결과 배열을 반환합니다.</p> <ul style="list-style-type: none"> <li>• Func는 단일 노드 인수를 사용하는 함수여야 하며 노드가 결과에 포함되어야 하는 경우 True, 테스트에 실패하는 경우 False를 반환합니다.</li> <li>• maxResults는 선택 사항이며 기본값은 1입니다. 제한 없음(조건을 통과하는 모든 노드)의 경우 0을 사용합니다.</li> <li>• searchFromTop은 선택 사항이며 기본값은 False입니다. 계층의 맨위에서 시작하려면 True를 사용합니다.</li> <li>• inclusive는 선택 사항이며 기본값은 False입니다. 잠재적 일치에 현재 노드를 포함하려면 True를 사용합니다(테스트를 통과해야 함).</li> </ul>
ChildrenWith(func, maxResults)	<p>노드의 하위 멤버 목록에서 지정된 함수를 충족하는 노드를 검색합니다. 하위 멤버를 찾는 가장 빠른 방법입니다. LocalNodeObject 결과 배열을 반환합니다.</p> <ul style="list-style-type: none"> <li>• func는 단일 노드 인수를 사용하는 함수여야 하며 노드가 결과에 포함되어야 하는 경우 True, 테스트에 실패하는 경우 False를 반환합니다.</li> <li>• maxResults는 선택 사항이며 기본값은 1입니다. 제한 없음(조건을 통과하는 모든 하위 멤버)의 경우 0을 사용합니다.</li> </ul>

표 12-18 (계속) LocalNodeObject 메소드

이름	설명
DescendantsWith(func, maxResults, inclusive, depthFirst)	<p>하위 멤버 체인에서 지정된 함수를 충족하는 노드를 검색합니다. 하위 멤버를 찾는 가장 빠른 방법입니다. LocalNodeObject 결과 배열을 반환합니다.</p> <ul style="list-style-type: none"> <li>func는 단일 노드 인수를 사용하는 함수여야 하며 노드가 결과에 포함되어야 하는 경우 True, 테스트에 실패하는 경우 False를 반환합니다.</li> <li>maxResults는 선택 사항이며 기본값은 1입니다. 제한 없음(조건을 통과하는 모든 노드)의 경우 0을 사용합니다.</li> <li>inclusive는 선택 사항이며 기본값은 False입니다. 잠재적 일치에 현재 노드를 포함하려면 True를 사용합니다(테스트를 통과해야 함).</li> <li>depthFirst는 선택 사항이며 기본값은 True입니다. True이면 트리를 백업하고 다음 분기로 이동하기 전에 각 분기가 끝까지 검사됩니다. False이면 노드의 모든 하위 멤버가 먼저 검사된 후 각 하위 멤버의 노드가 검사됩니다. 트리에서 노드의 위치를 알고 있는 경우 여기서 정확한 값을 선택하면 검색 속도가 훨씬 빨라질 수 있습니다.</li> </ul>
GetAncestorEnumerator()	상위 멤버 노드를 열거하는 NodeEnumeratorObject를 가져옵니다.
GetAncestors(inclusive)	LocalNodeObject 상위 멤버 배열을 가져옵니다.
GetChildEnumerator(sorted)	하위 멤버 노드를 열거하는 NodeEnumeratorObject를 가져옵니다. sorted가 True이면 하위 멤버가 정렬된 순서가 됩니다.
GetDescendantEnumerator()	하위 멤버 노드를 열거하는 NodeEnumeratorObject를 가져옵니다.
GetImplicitly SharedDescendants(inclusive)	이 공유 노드가 관련된 기본 노드의 하위 노드를 가져옵니다.
GetInvertedLevel()	공식 InvertedLevel 함수와 같습니다.
GetReferences()	이 노드에 대한 참조(이 노드가 나타나는 모든 계층 구조)인 LocalNodeObjects 배열을 가져옵니다.
GetReferenceInHier(hierAbbrev)	지정된 계층에서 이 노드에 대한 참조를 가져옵니다. 계층에 액세스할 수 없거나 이 노드가 계층에 없는 경우 결과는 null입니다.
NextSibling()	이 노드의 다음 동위 멤버를 정렬된 순서로 가져옵니다.
PreviousSibling()	이 노드의 이전 동위 멤버를 정렬된 순서로 가져옵니다.

표 12-18 (계속) LocalNodeObject 메소드

이름	설명
SiblingsWith(func, maxResults, inclusive)	<p>노드의 동위 멤버에서 지정된 함수를 충족하는 노드를 검색합니다. LocalNodeObject 결과 배열을 반환합니다.</p> <ul style="list-style-type: none"> <li>func는 단일 노드 인수를 사용하는 함수여야 하며 노드가 결과에 포함되어야 하는 경우 True, 테스트에 실패하는 경우 False를 반환합니다.</li> <li>maxResults는 선택 사항이며 기본값은 1입니다. 제한 없음(조건을 통과하는 모든 상위 멤버)의 경우 0을 사용합니다.</li> <li>inclusive는 선택 사항이며 기본값은 False입니다. 잠재적 일치에 현재 노드를 포함하려면 True를 사용합니다(테스트를 통과해야 함).</li> </ul>

NodePropObject

표 12-19 NodePropObject 등록정보

이름	설명
Abbrev	등록정보 정의의 이름
ControllingHierarchy	이 버전에서 등록정보 정의의 제어 계층에 대한 HierarchyObject. 등록정보가 글로벌 노드 등록정보가 아니거나, 제어 계층이 없거나, 제어 계층을 찾을 수 없는 경우 반환 값은 null이 됩니다.
Locked	값이 잠겨 있는 경우 True
Origin	PropOrigin 열거 값. 예: PropOrigin.Overridden(열거 상수 참조)
Owner	이 값이 연결된 객체(VersionObject, HierarchyObject, NodeObject 또는 LocalNodeObject)
PropType	PropType 열거 값. 예: PropType.Defined(열거 상수 참조)
StringValue	이 등록정보의 원시 문자열 값. Derived 또는 RWDerived 등록정보의 경우 등록정보 정의 기본값 또는 재정의된 값일 수 있습니다.
Value	이 등록정보의 해석된 값(예: DataType.Float 및 DataType.Integer의 경우 이 값은 Number 객체임). 모든 DataTypes에 문자열이 아닌 표현이 있는 것은 아닙니다.

표 12-20 NodePropObject 메소드

이름	설명
GetPropDef()	노드 등록 정보의 PropDefObject를 가져옵니다.

### RangeListObject

RangeListObject는 값의 RangeList를 나타내며, 수동으로 문자열을 구문분석할 필요 없이 RangeList 등록정보를 검사하는 데 사용될 수 있습니다. 적절한 데이터 유형의 파생 등록정보에서 반환할 새 RangeListObject를 구성할 수도 있습니다.

#### 구성자 예

```
var x = new RangeListObject();
```

```
var y = new RangeListObject("1-10,20-25");
```

```
var z = new RangeListObject([{start:1, end:10},{start:20, end:25}]);
```

표 12-21 RangeListObject 구성자 매개변수

매개변수	선택사항	설명
ranges	True	초기화의 범위 값입니다. 이 매개변수는 선택 사항입니다. 다음 두 가지 형식이 허용됩니다. <ul style="list-style-type: none"> <li>Array - 배열의 각 요소가 범위를 포함하여 start 및 end 등록정보가 있는 객체인 배열입니다. 이러한 등록정보가 없는 배열의 객체는 모두 무시됩니다.</li> <li>String - 쉼표로 구분된 문자열 항목 목록입니다. 각 항목에는 시작 및 끝 값이 대시(-) 또는 등호(=) 기호로 구분되어 포함됩니다.</li> </ul>

표 12-22 RangeListObject 등록정보

이름	설명
Ranges	객체 배열입니다. 각 객체에는 다음 두 개의 등록정보가 있습니다. <ul style="list-style-type: none"> <li>start - 범위 항목의 시작입니다.</li> <li>end - 범위 항목의 끝입니다.</li> </ul> 이 등록정보는 읽기 전용입니다. 범위를 수정하려면 아래 메소드를 사용합니다.

표 12-23 RangeListObject 메소드

이름	설명
AddRange(start, end)	범위 목록에 새 범위를 추가합니다. 이로 인해 기존 범위 항목이 확장되거나 새 범위 항목이 생성될 수 있습니다. 목록에 단일 숫자를 추가하려면 start 및 end 매개변수 둘 다에 사용합니다. 필요한 경우 두 매개변수는 정수로 강제됩니다.
Contains(value)	값이 범위 목록에 있으면 True를 반환하고, 그렇지 않으면 False를 반환합니다. 필요한 경우 value는 정수로 강제됩니다.
IsSupersetOf(range)	현재 RangeListObject가 지정된 RangeListObject의 수퍼 세트인 경우 True를 반환합니다. 다른 유형의 객체를 전달하면 오류가 발생합니다.
RemoveRange(start, end)	목록에서 범위를 제거합니다. 이 제거로 인해 기존 범위 항목이 둘로 분할되거나 항목이 완전히 제거될 수 있습니다. 목록에서 단일 숫자를 제거하려면 start 및 end 매개변수 둘 다에 사용합니다. 필요한 경우 두 매개변수는 정수로 강제됩니다.

### NodeEnumeratorObject

NodeEnumeratorObject를 사용하면 좀 더 효율적으로 노드 목록에 대해 작업할 수 있습니다. 전체 목록을 동시에 빌드하는 대신 열거자는 필요에 따라 한 번에 하나씩 노드를 잡습니다. 목록의 중간 부분에서 원하는 항목을 찾은 경우 열거자를 버릴 수 있습니다. 노드 객체의 배열을 반환하는 등록정보와 메소드는 배열 끝의 항목에 액세스하는지 여부에 관계없이 전체 배열을 즉시 빌드해야 합니다.

열거자는 null Current 값으로 시작합니다. 열거자를 목록의 첫 번째 노드로 이동하려면 MoveNext()를 호출해야 합니다.

 주:

가능한 모든 일치 항목에서 몇 개의 노드만 찾아야 하고 한 번만 목록을 반복해야 하는 경우 모범 사례는 *AncestorsWith* 또는 *SiblingsWith* 메소드 같은 *With* 메소드를 사용하는 것입니다. 상위 멤버 노드 목록을 여러 번 반복해야 하거나 상위 멤버가 대부분 또는 모두 필요한 것을 알고 있는 경우 열거자가 더 빠를 수 있습니다.

표 12-24 NodeEnumeratorObject 메소드

이름	설명
GetCurrent()	현재 노드입니다(컨텍스트에 따라 NodeObject 또는 LocalNodeObject).

표 12-24 (계속) NodeEnumeratorObject 메소드

이름	설명
MoveNext()	열거자를 다음 노드로 이동합니다. 열거할 노드가 더 이상 없는 경우 False를 반환합니다.

### ValidationObject

표 12-25 ValidationObject 등록정보

이름	설명
Abbrev	검증 이름(전체 네임스페이스 포함)
Descr	설명
EditorLabel	레이블
Cascade	검증 지정이 상속된 경우 True
ValidationClass	검증 클래스 이름
ValidationLevel	ValidationLevel 열거 값. 예: ValidationLevel.Node(열거 상수 참조)
ValidationType	ValidationType 열거 값. 예: ValidationType.Batch(열거 상수 참조)

### 검증 스크립트

- 검증 스크립트는 "success"라는 등록정보가 포함된 JavaScript 객체를 반환합니다. 스크립트가 부울 값이나 부울이 아닌 객체(예: 숫자 또는 문자열)를 반환하는 경우 해당 값이 표준 JavaScript 변환 규칙을 사용하여 부울로 변환된 다음 success 등록정보에 지정됩니다. 선택적으로 스크립트는 parameters라는 등록정보에 JavaScript 값 배열을 반환할 수 있습니다. 배열 값은 문자열 대체를 사용하여 검증 오류 메시지에 대체됩니다.
- 부울 값(True 또는 False)을 반환할 수 있습니다. True를 반환하면 검증에 성공하고, 그렇지 않으면 실패합니다. 값을 반환하지 않으면 False를 반환하는 것과 동일하게 간주됩니다.
- 부울이 아닌 객체(예: Number 또는 String)를 반환하면 부울로 변환된 후 반환됩니다. 표준 JavaScript 변환이 적용됩니다. 0과 같은 숫자, 빈 문자열 및 null 또는 정의되지 않은 객체는 false로 해석됩니다. 다른 모든 값은 true입니다.
- "success"라는 등록정보가 포함된 복합 객체를 반환하면 success 등록정보가 부울로 변환되며 검증의 반환 값으로 사용됩니다. 선택적으로 "parameters"라는 등록정보에 값의 배열을 반환할 수 있습니다. 이 JavaScript 배열 객체를 채운 다음 매개변수가 있는 실패 메시지에 사용해야 합니다. 매개변수는 문자열 대체를 사용하여 검증 오류 메시지에 대체됩니다. 오류 메시지의 자리 표시자에 해당하는 정확한 개수의 값을 반환해야 합니다. 추가 매개변수를 반환하면 무시됩니다. 충분한 매개변수를 반환하지 않을 경우 누락된 매개변수는 빈 문자열로 간주됩니다.

### RequestObject

RequestObject는 요청 헤더 및 항목을 포함한 관리 요청을 나타냅니다. Items 등록정보는 요청에 추가된 요청 항목 목록을 나타냅니다. 키 속성은 Version 등록정보, 해당 계층 구조 및 노드를 포함한 요청의 대상 버전, 관련된 스크립트 객체를 통해 액세스할 수 있는 모든 항목입니다.

표 12-26 RequestObject 등록정보

이름	설명
ID	ID
Title	요청 제목
Version	요청의 대상 버전
ModelName	요청에 대한 워크플로우 모델
StageName	현재 요청 단계
StageType	WorkflowStageType 열거 값. 예: WorkflowStageType.Submit(참조: 열거 상수)
Status	WorkflowStatus 열거 값. 예: WorkflowStatus.Submitted(참조: 열거 상수)
Items	요청에 추가된 RequestItemObject 목록

### RequestItemObject

RequestItemObject는 항목 세부정보(등록정보 값)와 함께 현재 태스크 및 업데이트되는 노드에 대한 정보를 포함한 관리 요청에 대한 개별 요청 항목을 나타냅니다. 요청 등록정보는 헤더 등록정보 및 기타 항목을 비롯한 전체 요청 항목 객체에 대한 액세스 권한을 제공합니다.

NodeNamePendingInRequest 메소드는 대상 버전에 대해 진행 중인 다른 요청과의 잠재적인 노드 이름 충돌을 식별하는 데 사용되며, 대기 중인 다른 요청 항목에 동일한 노드 이름의 추가 항목이 포함되어 있는 경우 True를 반환합니다.

표 12-27 RequestItemObject 등록정보

이름	설명
ItemID	항목 ID
RequestID	요청 ID
요청	항목이 속한 요청 객체
NodeName	업데이트되는 노드의 Core.Abbrev
설명	업데이트되는 노드의 Core.Descr
HierarchyName	업데이트되는 노드의 계층
ParentName	업데이트되는 노드의 Core.Parent
TaskName	요청 항목의 워크플로우 태스크 이름
TaskAction	WorkflowAction 열거 값. 예: WorkflowAction.AddLimb(참조: 열거 상수)
TaskDomain	워크플로우 태스크의 도메인 이름(있는 경우)
ItemDetails	요청 항목에 대한 RequestItemDetailObject 목록

표 12-28 RequestItemObject 메소드

이름	설명
NodeNamePendingInRequest(name)	테스트할 특정 노드 이름의 매개변수를 허용합니다. 버전의 현재 요청이 아닌 진행 중인 다른 요청에 지정된 이름의 AddLimb/Leaf 항목이 포함된 경우 True를 반환합니다.

**RequestItemDetailObject**

RequestItemDetailObject는 단일 등록정보 값에 해당하는 관리 요청에 대한 개별 요청 항목 세부정보를 나타냅니다.

표 12-29 RequestItemDetailObject 등록정보

이름	설명
CalcValue	계산된 등록정보 값
HasCalcValue	값이 계산된 경우 True를 반환합니다.
Modified	요청에서 값이 수정된 경우 True를 반환합니다.
PropertyName	등록정보 이름
Value	등록정보 값

## 실행 환경

Oracle Data Relationship Management 엔진은 다중 스레드 다중 컴퓨터 환경이며 스크립트가 여러 스레드와 컴퓨터에서 동시에 실행될 수 있습니다. 값을 생성하고 글로벌 범위에 저장할 수 있지만 스크립트가 다른 스레드에서 실행되는 경우 해당 글로벌 값이 없기 때문에 이 동작에 의존해서는 안 됩니다. 마찬가지로, 글로벌 값은 Data Relationship Management 엔진 인스턴스나 컴퓨터 전체에서 업데이트되지 않습니다. 또한 Data Relationship Management는 여러 라이브 버전을 지원하므로 한 노드의 값을 계산하여 글로벌 범위에 해당 값을 저장하는 경우 다른 스크립트가 다른 노드의 등록정보에 액세스하면 부정확한 값이 생성될 수 있습니다.

### 주:

변수를 글로벌 범위에 저장하면 안 되는 것과 동일한 이유로, 수정 내용이 모든 엔진 인스턴스와 스레드에 적용되었는지 확인할 수 없으므로 기본 제공 Data Relationship Management 객체 프로토타입은 수정하면 안 됩니다.

**스크립트 시간 초과 설정**

과도한 엔진 잠금을 방지하기 위해 값을 반환하지 않고 너무 오래 실행되는 스크립트는 시간 초과 설정에 따라 종료됩니다. 각 등록정보 정의 및 검증에 대해 스크립트 시간 초과를 설정할 수 있습니다.

시간 초과는 실행 컨텍스트별로 적용되므로 익스포트에서 100개 노드의 스크립트 등록정보를 내보내는 중이며 등록정보의 시간 초과가 30초로 설정된 경우 각 노드에서 해당 등록정보를 평가하는 데 30초가 걸릴 수 있으므로 익스포트는 최대 5분이 걸릴 수 있습니다. 그러나 스크립트

등록정보가 다른 스크립트 등록정보를 호출하는 경우에는 시간 초과가 증가하지 않습니다. 예를 들어 PropA에 10초 시간 초과가 있고 PropB에 20초 시간 초과가 있으며 PropA가 장기 실행 계산을 시작하는 PropB를 호출하는 경우 10초가 경과하면 원래 시간 초과를 초과했으므로 PropA의 평가가 종료됩니다.

### 무한 루프 방지

무한 루프(스택 오버플로우라고도 함)를 발생시키는 스크립트는 심각한 오류이며, 서버 프로세스가 예기치 않게 종료될 수 있습니다. Data Relationship Management는 이러한 스크립트가 실행되지 않도록 하지만 자체 참조 스크립트나 반복적인 스크립트를 작성할 때는 주의해야 합니다. 프로덕션에 배포하기 전에 항상 개발 환경에서 새 스크립트를 테스트합니다.

무한 루프를 생성하는 스크립트의 간단한 예제는 다음과 같습니다. 이 스크립트에는 자체 호출이 포함되어 있지만 실행을 종료하지 않으므로 함수를 실행하는 엔진은 결국 리소스 부족 때문에 종료됩니다. 마지막으로, 이 스크립트는 Data Relationship Management 엔진을 호출하지 않으므로 오버플로우를 파악해서 스크립트를 중지할 수 없습니다.

```
function badFunc(a) { badFunc(a); }
```

```
badFunc("oops");
```

### 성능 고려 사항

최상의 성능을 얻으려면 공식으로 파생된 등록정보를 스크립트에서 참조하거나 그 반대로 참조하지 마십시오. 일반적으로 스크립트는 64비트 프로세서를 비롯한 기본 하드웨어에 대한 JIT(Just-In-Time) 컴파일 등의 고려 사항 때문에 공식에 비해 성능 조정 최적화를 위한 최상의 기회를 제공합니다. 또한 스크립트는 JIT 컴파일러에서 실제 실행 특성에 맞게 조정되며 시간 경과에 따라 더 빨리 실행됩니다.

## 동적 스크립트 생성

동적 스크립트는 파생 등록정보 정의 및 검증을 위해 [매개변수] 탭에서 사용할 수 있는 스크립트 편집기에서 생성됩니다.

관리 워크플로우 태스크에서 이름 또는 상위 멤버를 계산할 때 스크립트 편집기도 사용할 수 있습니다.

#### 주:

상위 이름을 계산할 때 특수 문자가 사용된 경우 표준 JavaScript 규칙에 따라 특수 문자를 이스케이프해야 합니다. 자세한 내용은 *Oracle Data Relationship Management 사용자 가이드*의 "노드 이름 바꾸기"를 참조하십시오.

동적 스크립트를 생성하려면 다음을 수행합니다.

1. 텍스트 영역에 스크립트를 입력합니다.

 주:

등록정보를 삽입하려면 스크립트에 커서를 놓고 **등록정보 삽입**을 누릅니다.  
등록정보 목록이 표시됩니다. 등록정보를 선택하고 **확인**을 누릅니다.

2. 다음 옵션 중에서 선택합니다.
  - **스크립트 시간 초과**—스크립트가 시간 초과될 때까지의 초 수입니다.
  - 선택한 노드로 스크립트를 평가하려면 을 누르고 노드를 선택합니다. 노드의 현재 등록정보 값이 스크립트에 사용됩니다. **평가**를 누릅니다. 결과가 스크립트 디자이너의 맨아래에 표시됩니다.
3. 스크립트를 테스트하려면 **평가**를 누릅니다.

# 13

## 노드 유형 관리

노드 유형을 사용하면 관계와 속성에 따라 다르게 계층 노드를 보고 관리할 수 있습니다. 특정 노드 유형의 노드는 다음과 같은 동일한 특성을 공유합니다.

- 등록정보
- 검증
- 그래픽 기호

한 계층에 여러 노드 유형의 노드가 포함될 수 있고 동일한 노드가 다른 계층에서는 다른 노드 유형일 수 있습니다. 노드 유형 사용 예로 GL 계정, 비용 센터, 통합 엔티티, 제품 그룹, 예측 지점 등이 있습니다.

노드 유형별로 노드를 범주화하려면 다음을 수행합니다.

1. 계층 내의 노드를 범주화하는 데 필요한 노드 유형을 확인합니다.
2. 각 노드 유형과 관련이 있거나 관련이 없는 등록정보를 식별합니다.
3. 각 노드 유형과 관련이 있거나 관련이 없는 검증을 식별합니다.
4. 선택적으로, 각 노드 유형에 그래픽 기호를 지정합니다.

## 노드 유형 정의

노드 유형을 정의하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **새로 작성**에서 **노드 유형**을 선택합니다.
3. 노드 유형의 이름과 설명을 입력합니다.
4. **선택 사항**: 노드 유형에 사용할 그래픽 기호를 선택합니다.
5. **등록정보** 탭의 **[사용 가능]** 목록에서 노드 유형과 연결할 등록정보를 선택합니다. 화살표를 사용하여 등록정보를 **선택됨** 목록으로 이동합니다.
6. **검증** 탭의 **[사용 가능]** 목록에서 노드 유형과 연결할 검증을 선택합니다. 화살표를 사용하여 검증을 **선택됨** 목록으로 이동합니다.
7. **저장**을 누릅니다.

## 노드 유형 편집

노드 유형을 편집하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **메타데이터**에서 **노드 유형**을 확장합니다.
3. 노드 유형을 선택하고  을 누릅니다.
4. 다음 중 하나를 수행합니다.

- 설명 편집
  - 노드 유형에 사용할 그래픽 기호 변경
  - 등록정보 추가 또는 제거
  - 검증 추가 또는 제거
5. **저장**을 누릅니다.

## 노드 유형 삭제

노드 유형을 삭제하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **메타데이터**에서 **노드 유형**을 확장합니다.
3. 노드 유형을 선택하고 **✖**을 누릅니다.
4. **이 항목 삭제**를 눌러 삭제를 확인합니다.

## 노드 그래픽 기호 작업

그래픽 기호는 노드 유형에 연계되고 Oracle Data Relationship Management 사용자 인터페이스에서 노드의 아이콘으로 표시되는 이미지입니다. 새 그래픽 기호를 생성하고 기존 그래픽 기호를 수정할 수 있습니다. 더 이상 사용하지 않을 그래픽 기호를 삭제할 수도 있습니다. 그래픽 기호는 PNG 형식으로 제공해야 합니다.

노드 그래픽 기호를 추가하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **새로 작성**에서 **그래픽 기호**를 선택합니다.
3. 그래픽 기호의 이름을 입력하고 설명을 추가합니다.
4. **찾아보기**를 누르고 PNG 파일을 선택합니다.
5. **업로드**를 누릅니다.
6. **저장**을 누릅니다.

노드 그래픽 기호를 수정하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **메타데이터**에서 **그래픽 기호**를 확장합니다.
3. 그래픽 기호를 선택하고 **✎**을 누릅니다.
4. **찾아보기**를 누르고 다른 PNG 파일을 선택합니다.
5. **업로드**를 누릅니다.
6. **저장**을 누릅니다.

그래픽 기호를 삭제하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **메타데이터**에서 **그래픽 기호**를 확장합니다.
3. 그래픽 기호를 선택하고 **✖**을 누릅니다.

4. 이 항목 삭제를 눌러 삭제를 확인합니다.

# 14

## 시스템 환경설정 작업

시스템 환경설정을 통해 관리자는 Oracle Data Relationship Management의 동작을 제어하는 설정을 편집할 수 있습니다.

### 시스템 환경설정

다음 테이블에서는 Oracle Data Relationship Management 시스템 환경설정에 대해 설명합니다.

표 14-1 시스템 환경설정

시스템 환경설정	유형	설명
AllowAsOf	Boolean	True이면 핵심 작업이 강제로 캡처되고 기준선 버전이 생성되어 As-Of 버전을 생성할 수 있게 합니다. 이 환경설정이 False로 설정된 경우 As-Of 버전을 생성할 수 없습니다. 기본값은 True입니다. 참고: 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.
AllowNextIDGeneration	Boolean	True이면 다음 ID가 자동으로 생성됩니다. 기본값은 False입니다.
AllowNextIDKeyCreation	역할	NextID 기능에 새 키를 생성할 수 있는 역할 목록입니다. 기본값은 대화식 사용자, 데이터 생성자, 데이터 관리자입니다.
AllowPru	Boolean	True이면 관리자가 아닌 사용자가 하위 멤버를 포함하는 노드를 제거할 수 있게 하는 정리 옵션이 활성화됩니다. False이면 관리자가 아닌 사용자가 하위 멤버를 포함하는 노드를 제거할 수 없습니다. 기본값은 True입니다.
AllowRelaxedMove	Boolean	노드를 이동할 때 True이면 새 상위 멤버가 다른 계층 구조에 있는 노드의 충돌하는 상위 관계보다 우선합니다. 기본값은 False입니다.
AllowSpac	Boolean	True이면 노드 이름에 공백을 사용할 수 있습니다. 기본값은 True입니다.

표 14-1 (계속) 시스템 환경설정

시스템 환경설정	유형	설명
AnalyticsNodeCountUpdateTime	String	로드된 모든 일반 버전에 대한 버전 및 계층 구조의 노드 수가 업데이트되어야 하는 시간을 24시간 형식의 지역 시간으로 지정합니다. 예를 들어 오후 2:15는 "1415"로 입력합니다. 기본 시간은 오전 3:00입니다.
ApprovalGroups	String	심표로 구분된 승인 그룹 목록입니다.
ApprovalGroupTrackProperties	String	그룹별로 추적된 구분된 승인 등록정보 목록입니다.
ApprovalPropertyByApprovalGroup	String	승인 그룹별 글로벌 부울 승인 등록정보입니다.
AuthMethod	String	<p>사용자 인증 방법입니다.</p> <ul style="list-style-type: none"> <li>내부 – 사용자가 Data Relationship Management 내에서만 인증됩니다.</li> <li>CSS(외부) – 사용자가 외부에서만 인증됩니다. Shared Services에 액세스할 수 있어야 합니다.</li> <li>혼합 – 사용자가 개별 사용자에 대한 설정에 따라 내부 또는 외부에서 인증됩니다.</li> </ul> <p>기본값은 내부입니다.</p> <p>참고: 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.</p>
CopyLcl	Boolean	<p>True이면 노드를 복사할 때 로컬 값이 복사됩니다.</p> <p>기본값은 True입니다.</p>
DefaultCurrentVersion	Version	기본 현재 버전입니다. 버전에 대한 [기본값으로 설정] 옵션을 사용하여 이 환경설정을 설정할 수 있습니다.
DefaultPreviousVersion	Version	기본 이전 버전입니다. 버전에 대한 [기본값으로 설정] 옵션을 사용하여 이 환경설정을 설정할 수 있습니다.
DefaultPropCopyMode	String	<p>기본 등록정보 복사 모드입니다.</p> <p>적합한 값은 재정의됨, 선택됨 및 모두 강제 적용입니다.</p> <p>기본값은 재정의됨입니다.</p>
EnablePropCopyOptions	역할	<p>등록정보 복사 옵션에 대한 액세스가 허용되는 역할 목록입니다.</p> <p>기본값은 대화식 사용자, 데이터 생성자, 데이터 관리자입니다.</p>

표 14-1 (계속) 시스템 환경설정

시스템 환경설정	유형	설명
EnforceListProps	Boolean	True이면 사전 정의된 목록의 값으로만 목록 등록정보를 업데이트할 수 있습니다. 기본값은 True입니다.
FiltrChr	String	익스포트 [출력 옵션] 화면의 Replace 함수에 대한 문자 세트입니다.
FindByProperties	Property	계층을 찾을 때 검색할 수 있는 등록정보 목록입니다. 사용자에게 액세스 권한이 있는 등록정보가 표시됩니다. 표시되는 등록정보가 일부 계층 구조에 적용되지 않을 수도 있습니다. <b>참고:</b> Data Relationship Management에서는 사용자정의 등록정보 범주에 관리 사용자를 추가할 수 없습니다. 따라서 FindByProperties 시스템 환경설정에 나열된 등록정보가 ADMIN이 이미 멤버로 포함된 등록정보 범주에 추가되어 있지 않으면 ADMIN은 계층 구조 찾아보기 창에서 해당 등록정보를 사용하여 찾기를 수행할 수 없습니다.
FindWildCardAppend	Boolean	True이면 정확한 일치를 선택하지 않은 경우 [찾기] 기준 뒤에 별표(*)가 추가됩니다. 기본값은 False입니다.
FindWildCardPrepend	Boolean	True이면 정확한 일치를 선택하지 않은 경우 [찾기] 기준 앞에 별표(*)가 추가됩니다. 기본값은 False입니다.
GlobalPropLocalOverride	Property	글로벌 등록정보에 대한 로컬 확인에서 제외할 등록정보 목록입니다. GlobalPropLocalSecurity를 활성화한 경우에 사용됩니다. <b>참고:</b> 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.
GlobalPropLocalSecurity	Boolean	True이면 글로벌 등록정보에 로컬 보안이 적용됩니다. 노드가 있는 모든 계층 구조에 대해 사용자의 로컬 보안 (노드 액세스 레벨)을 기준으로 글로벌 등록정보의 변경사항을 확인합니다. 기본값은 False입니다. <b>참고:</b> 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.

표 14-1 (계속) 시스템 환경설정

시스템 환경설정	유형	설명
HierSep	String	계층 및 노드 구분 문자입니다. 기본값은 물결표(~)입니다.
IdleTime	Integer	애플리케이션 서버에서 세션 시간 초과까지 남은 분 수입니다. 기본값은 60입니다. <b>참고:</b> 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.
비활성화	역할	비활성 노드에 허용되는 사용자 역할 목록입니다. 기본값은 모든 역할입니다.
InactiveChanges	역할	비활성 노드를 변경할 수 있는 역할 목록입니다. 기본값은 데이터 관리자, 애플리케이션 관리자, 액세스 관리자입니다.
InvDescr	String	노드 설명 등록정보에 부적합한 문자 목록입니다.
InvName	String	노드 이름에 부적합한 문자 목록입니다. <b>참고:</b> 이 목록의 문자는 공유 노드에서 구분자로 사용할 수 없습니다.
JobResultsMaxSize	Integer	[클라이언트 파일] 옵션을 사용하여 실행된 작업의 경우 작업 내역에 저장된 결과의 최대 크기(바이트)입니다. 이 크기를 초과하는 작업 결과는 작업 내역에 저장되지 않습니다. 기본값은 10,000,000바이트입니다. 음수 값은 크기에 관계없이 모든 결과가 작업 내역에 저장됨을 나타냅니다. <b>주의:</b> 큰 작업의 성능이 저하될 수 있으므로 음수 값으로 설정하여 JobResultsMaxSize를 사용 안함으로 설정하지 않는 것이 좋습니다. <b>참고:</b> [서버 파일] 또는 [데이터베이스 테이블] 옵션을 사용하여 실행된 익스포트에는 JobResultsMaxSize가 적용되지 않습니다.
JobResultsRetentionAge	Integer	내역에서 아카이브된 작업 결과 세부정보를 유지할 일수입니다. 0 값은 작업 결과가 내역에서 제거되지 않음을 나타냅니다. <b>참고:</b> 데이터베이스 크기를 관리하기 위해 작업 결과가 제거됩니다. 제거를 비활성화하면 시간 경과에 따라 데이터베이스가 상당히 커질 수 있습니다.

표 14-1 (계속) 시스템 환경설정

시스템 환경설정	유형	설명
LeafEdit	역할	리프 등록정보를 변경할 수 있는 역할 목록입니다. 기본값은 데이터 관리자, 데이터 생성자, 애플리케이션 관리자, 액세스 관리자입니다.
LockoutInactivity	Integer	사용자가 잠길 때까지 허용되는 최대 비활동 일수입니다. 기본값은 30입니다. 0이면 최대값이 없습니다.
LockoutInvalidLogins	Integer	사용자가 잠길 때까지 허용되는 부적합한 로그인 최대 횟수입니다. 기본값은 6입니다. 0이면 최대값이 없습니다.
LossLevel	String	캡처할 손실 레벨입니다. 적합한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• 정의됨</li> <li>• 모두</li> </ul> 기본값은 정의됨입니다. [모두]를 선택하면 많은 등록정보 값이 있는 제거 또는 삭제된 노드에 대한 시스템 성능이 크게 저하될 수 있습니다. <b>참고:</b> 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.
LRUPropertyCacheSize	Integer	LRU 등록정보 캐시의 최대 크기입니다. LRU 등록정보 캐시는 여러 번 액세스될 수 있는 계산된 값을 저장합니다. 일반적으로 이 환경설정의 기본값을 사용하고 값을 변경하지 않아야 합니다.
MaxDescr	Integer	노드 설명의 최대 문자 수입니다. 적합한 값은 12에서 255 사이입니다. 기본값은 80입니다.
MaxLeaf	Integer	리프 이름의 최대 문자 수입니다. 적합한 값은 3에서 20 사이입니다. 기본값은 255입니다.
MaxLimb	Integer	림 이름의 최대 문자 수입니다. 적합한 값은 3에서 20 사이입니다. 기본값은 255입니다.
NodeApprovedSecurity	역할	노드에 대한 NodeApproved 시스템 등록정보를 보고 업데이트할 수 있는 역할 목록입니다.
PasswordDuration	Integer	사용자 비밀번호가 적합한 일수입니다. 적합한 값은 1에서 9999 사이입니다. 기본값은 30입니다.

표 14-1 (계속) 시스템 환경설정

시스템 환경설정	유형	설명
PasswordMaxLength	Integer	사용자 비밀번호의 최대 문자 수입니다. 적합한 값은 0에서 255 사이입니다. 0이면 최소값이 없습니다. 기본값은 0입니다.
PasswordMinLength	Integer	사용자 비밀번호의 최소 문자 수입니다. 적합한 값은 0에서 9999 사이입니다. 0이면 최소값이 없습니다. 기본값은 6입니다.
PasswordPolicyEnabled	Boolean	True이면 비밀번호에 다음 요소 중 3개가 포함되어야 합니다. <ul style="list-style-type: none"> <li>• 대문자</li> <li>• 소문자</li> <li>• 숫자</li> <li>• 특수 문자</li> </ul> 기본값은 True입니다.
PasswordWarningPeriod	Integer	사용자가 더 이상 로그인할 수 없게 되기 전에 비밀번호를 변경하도록 경고할 비밀번호 만료 날짜 이전(-) 또는 이후(+) 일수를 나타내는 양수 또는 음수입니다. 적합한 값은 -30에서 30 사이입니다. 기본값은 1입니다.
RenameLeaf	역할	리프 노드의 이름을 바꿀 수 있는 역할 목록입니다. 기본값은 데이터 관리자, 애플리케이션 관리자, 액세스 관리자입니다.
RenameLimb	역할	림 노드의 이름을 바꿀 수 있는 역할 목록입니다. 기본값은 모든 역할입니다.
ReqMerge	Boolean	True이면 UseMerge가 활성화된 경우 비활성화 또는 삭제에 대해 병합이 필요합니다. 기본값은 False입니다.

표 14-1 (계속) 시스템 환경설정

시스템 환경설정	유형	설명
SharedNodeDelimiter	String	<p>노드 이름과 공유 노드 접미어 사이의 구분자를 지정합니다.</p> <p>노드 이름에 영향을 주는 경우 SharedNodeDelimiter 문자를 사용하면 안 됩니다.</p> <p>기본값은 콜론(:)입니다.</p> <p><b>주의:</b> SharedNodeDelimiter 및 SharedNodeSequenceSeparator 시스템 환경설정을 지정할 때는 서로 다른 문자를 사용해야 합니다. 예를 들어, SharedNodeDelimiter가 콜론인 경우 SharedNodeSequenceSeparator 문자는 콜론일 수 없습니다.</p> <p><b>참고:</b> 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.</p>
SharedNodeIdentifier	String	<p>공유 노드 구분자 뒤에 사용할 식별자를 지정합니다.</p> <p>기본값은 공유입니다.</p> <p><b>참고:</b> 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.</p>
SharedNodeMaintenanceEnabled	Boolean	<p>True이면 공유 노드가 활성화됩니다.</p> <p>기본값은 False입니다.</p> <p><b>참고:</b> 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.</p>
SharedNodeNamingType	String	<p>공유 노드의 대체 이름을 지정합니다.</p> <p>적합한 값은 접미어 또는 접두어입니다.</p> <p>기본값은 접미어입니다.</p> <p><b>참고:</b> 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.</p>
SharedNodeSequenceLength	Integer	<p>숫자 순서 유형을 사용하는 경우 고유성 키의 길이를 지정합니다.</p> <p>기본값은 3입니다.</p> <p><b>참고:</b> 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.</p>

표 14-1 (계속) 시스템 환경설정

시스템 환경설정	유형	설명
SharedNodeSequenceSeparator	String	공유 노드 식별자 뒤에 배치할 구분 문자를 지정합니다. 기본값은 대시(-)입니다. <b>주의:</b> SharedNodeDelimiter 및 SharedNodeSequenceSeparator 시스템 환경설정을 지정할 때는 서로 다른 문자를 사용해야 합니다. 예를 들어, SharedNodeDelimiter가 콜론인 경우 SharedNodeSequenceSeparator 문자는 콜론일 수 없습니다. <b>참고:</b> 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.
SharedNodeSequenceType	String	고유성 키의 유형을 지정합니다. 적합한 값은 숫자 또는 상위 멤버입니다. 기본값은 숫자입니다. <b>참고:</b> 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.
SortLimbsFirst	Boolean	True이면 림 노드가 먼저 정렬되고 리프 노드가 정렬됩니다. False이면 림 및 리프 노드를 함께 정렬할 수 있습니다. 이 환경설정은 계층 익스포트, 표시 및 노드 목록에 영향을 줍니다. 기본값은 True입니다.
TopNodeParentString	String	임포트 및 익스포트에서 최상위 노드의 상위 멤버 값을 나타내는 데 사용됩니다. 기본값은 없음입니다.

표 14-1 (계속) 시스템 환경설정

시스템 환경설정	유형	설명
TransactionLevels	String	<p>캡처할 트랜잭션 레벨 목록입니다. As-Of를 설정하거나 결과 또는 손실 작업을 지정하면 핵심 작업이 강제로 캡처됩니다.</p> <p>적합한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• 기록된 작업</li> <li>• 핵심 작업</li> <li>• 결과 작업</li> <li>• 손실 작업</li> </ul> <p><b>참고:</b> 관리 레벨의 트랜잭션은 이 시스템 환경설정에 관계없이 항상 로깅됩니다.</p> <p>기본값은 기록된 작업, 핵심 작업, 결과 작업, 손실 작업입니다.</p> <p><b>참고:</b> 이 환경설정을 변경하는 경우 Data Relationship Management 애플리케이션을 다시 시작해야 합니다.</p>
UpName	Boolean	<p>True이면 노드 이름에 항상 대문자가 사용됩니다.</p> <p>기본값은 False입니다</p>
UseChangeApproval	Boolean	<p>True이면 변경 승인이 활성화됩니다.</p> <p>기본값은 False입니다.</p>
UseMerge	Boolean	<p>True이면 비활성화된 노드와 삭제된 노드에 대해 병합 방법론 사용이 활성화됩니다.</p> <p><b>참고:</b> ReqMerge가 True이면 시스템에서 병합 노드를 지정해야 합니다. ReqMerge가 False이면 노드 승인됨 등록정보가 True가 아닌 경우 병합 노드는 선택 사항입니다. 버전이 종료되었거나 적절한 액세스 권한을 가진 사용자가 구체적으로 True로 설정한 경우 노드 승인됨 등록정보가 True로 설정됩니다.</p> <p>기본값은 False입니다.</p>
ValSec	Boolean	<p>True이면 사용자가 노드에 대한 बै치 검증을 실행할 수 있는지 확인하기 위해 노드 액세스 그룹 보안이 확인됩니다.</p> <p>기본값은 False입니다.</p>
WarnHL	Integer	<p>하위 멤버, 1차 하위 멤버, 쿼리 결과 등의 목록에 대해 표시할 최대 노드 수입니다. 최소값은 1000입니다. 1000보다 작은 값으로 설정하면 1000개 노드가 표시됩니다.</p> <p>기본값은 5000입니다.</p>

자세한 내용은 다음을 참조하십시오.

- [트랜잭션 내역 로깅 레벨 설정](#)
- [변경 승인 설정](#)

## 트랜잭션 내역 로깅 레벨 설정

Oracle Data Relationship Management 트랜잭션 내역 로깅 레벨을 설정하려면 애플리케이션 관리자 권한이 있어야 합니다. TransactionLevels 시스템 환경설정을 설정하여 트랜잭션 내역에 캡처할 작업 유형을 지정합니다.

### 글로벌 등록정보에 대한 로컬 보안

두 개의 시스템 환경설정(GlobalPropLocalSecurity 및 GlobalPropLocalOverride)을 사용하여 글로벌 등록정보에 대한 로컬 보안을 제어합니다.

트랜잭션 내역 로깅 레벨을 설정하려면 다음을 수행합니다.

1. Data Relationship Management 웹 클라이언트에서 **관리**를 선택합니다.
2. **메타데이터**에서 **시스템 환경설정**을 확장하고 **TransactionLevels** 환경설정을 편집합니다.
3. **TransactionLevels**에서 다음 트랜잭션 레벨 유형을 선택합니다.
  - **기록된 작업**은 기본 로깅 정보(예: 사용자 로그인 및 로그아웃)를 기록합니다.
  - **핵심 작업**은 버전, 계층 또는 노드 정보를 변경하는 작업(예: 노드 추가, 등록정보 변경 또는 노드 이동)을 기록합니다.
  - **결과 작업**은 핵심 작업에서 발생하는 작업을 기록합니다. 예를 들어 "아래 모두 지우기" 핵심 작업을 수행하면 개별 노드에서 등록정보가 지워집니다. 개별 노드에서 등록정보를 지우는 작업은 결과 작업입니다.
  - **손실 작업**은 핵심 작업으로 인한 데이터 손실을 기록합니다. 예를 들어 노드를 삭제하면 해당 노드에 대해 정의된 등록정보가 삭제되며, 이는 손실 작업입니다. 손실 작업은 LossLevel 시스템 환경설정에 의해 제어됩니다.

### 주:

손실 작업을 지정하거나 AllowAsOf system 환경설정을 켜면 TransactionLevels 시스템 환경설정에서 설정되지 않은 경우에도 핵심 작업이 추적됩니다.

4. LossLevel 환경설정을 설정합니다.
  - **정의됨** – 노드를 삭제하면 구체적으로 노드에 설정된 값만 추적됩니다.
  - **모든 항목** – 파생된 값, 기본값 및 상속된 값은 LossAction에서 추적됩니다.
5. 애플리케이션을 중지하고 다시 시작하거나 Oracle Hyperion Data Relationship Management 서비스를 다시 시작합니다.

## 변경 승인 설정

Oracle Data Relationship Management의 변경 승인 시스템을 사용하면 승인 그룹을 정의하고 등록정보 또는 특수 작업 세트에 의해 트리거되는 승인 플래그에 연결할 수

있습니다. 이렇게 하면 일반 사용자는 변경 작업을 수행할 수 있고 승인자는 쿼리를 실행한 다음 필요에 따라 승인 플래그를 설정할 수 있습니다.

Data Relationship Management의 변경 승인 동작을 결정하는 시스템 환경설정은 다음과 같습니다.

- UseChangeApproval – 변경 승인 사용을 설정하려면 True로 설정합니다.
- ApprovalGroups – 시스템에서 사용된 승인 그룹의 심표로 구분된 이름 목록입니다.
- ApprovalGroupTrackProperties – UseChangeApproval이 True이면 이 그룹의 승인 플래그를 False로 변경하는 이벤트를 트리거할, 추적되는 등록정보를 정의합니다. 형식은 xxx[a,b,c],yyy[d,e,f]...입니다. 여기서 xxx와 yyy는 ApprovalGroups 환경설정에 정의된 판매 그룹이고 a,b,c,d,e,f는 등록정보 이름입니다(예: Sales[Custom.SalesGroup, {NodeMove}], Treasury[Custom.AccountDescription, {NodeAdd}]).

등록정보 목록에 포함될 수 있는 특수 작업은 다음과 같습니다.

- {NodeAdd} – 추가된 노드에서 승인 필요 메커니즘을 트리거합니다.
- {NodeInactivate} – 비활성화된 노드에서 승인 필요 메커니즘을 트리거합니다.
- {NodeReactivate} – 재활성화된 노드에서 승인 필요 메커니즘을 트리거합니다.
- {NodeInsert} – 삽입된 노드에서 승인 필요 메커니즘을 트리거합니다.
- {NodeRemove} – 제거된 노드에서 승인 필요 메커니즘을 트리거합니다.
- {NodeMove} – 이동된 노드에서 승인 필요 메커니즘을 트리거합니다.
- ApprovalPropertyByApprovalGroup – UseChangeApproval이 True이면 트리거 등록정보가 하나라도 변경되거나 특수 작업이 사용되는 경우 False로 설정할 글로벌 부울 등록정보를 정의합니다. 형식은 xxx:bbbb,yyy:cccc...입니다. 여기서 xxx와 yyy는 ApprovalGroups 환경설정에 정의된 판매 그룹이고 bbbb 및 cccc는 그룹에 대한 승인 플래그를 저장하는 데 사용할 글로벌 부울 등록정보의 이름입니다(예: Sales:Custom.SalesApprovedFlag, Treasury:Custom.TreasuryApprovedFlag).

## 시스템 환경설정 구성

시스템 환경설정을 구성하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **메타데이터**에서 **시스템 환경설정**을 확장합니다.
3. 시스템 환경설정을 선택하고 을 누릅니다.
4. 값을 수정하고 **저장**을 누릅니다.

# 15

## 외부 연결 작업

애플리케이션 관리자는 외부 파일 시스템, 데이터베이스 및 웹 서비스에 대한 공통 연결을 정의하고 구성할 수 있습니다. 임포트, 익스포트 및 장부는 파일 및 데이터베이스 연결을 공유하여 연결 정보의 유지 관리를 최소화할 수 있습니다. 외부 작업으로 데이터베이스 및 웹 서비스 연결을 구성하여 외부 시스템에서 데이터를 록업하거나 데이터 변경사항을 외부 시스템에 커밋할 수 있습니다. 외부 연결을 사용하면 애플리케이션 서버가 이러한 리소스 데이터에 직접 액세스하거나 읽거나 쓸 수 있습니다.



주:

외부 연결을 정의하기 전에 외부 리소스를 설정해야 합니다.

### 외부 작업

웹 서비스 또는 데이터베이스 외부 연결에 대해 외부 작업을 정의할 수 있습니다. 외부 작업은 록업 또는 커밋으로 구성됩니다. 록업 작업은 외부 시스템에서 데이터를 읽습니다. 커밋 작업은 외부 시스템에 데이터를 씁니다. 데이터베이스 및 웹 서비스 연결은 여러 작업을 지원할 수 있습니다. 자세한 내용은 [외부 커밋](#) 및 [외부 록업](#)을 참조하십시오.

## 외부 연결 정의

외부 연결을 정의하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **새로 작성**에서 **외부 연결**을 선택합니다.
3. 이름과 설명을 입력합니다.
4. **객체 액세스**에서 **표준, 시스템** 또는 사용자정의 그룹을 선택합니다.
5. 연결 유형(**서버 파일, FTP, 데이터베이스** 또는 **웹 서비스**)을 선택합니다.
6. 다음 중 하나를 수행합니다.

- **서버 파일**을 선택한 경우 서버의 UNC 경로를 입력하고 을 누릅니다.



주:

Oracle Data Relationship Management 애플리케이션 서버에서 사용하는 Windows 사용자 계정이 자동으로 서버 파일 연결에 사용됩니다. **Oracle DRM 서버 프로세스** Windows 서비스에 사용되는 기본 Windows 사용자 계정은 **로컬 시스템 계정**입니다. 서비스에 사용되는 계정이 적절한 서버 파일 연결의 UNC 경로에 액세스할 수 있어야 합니다. 또한, UNC 경로에 서비스 계정이 파일을 읽고 쓰는 데 적절한 권한이 있어야 합니다.

- **FTP**를 선택한 경우 다음 정보를 입력합니다.
  - 호스트 서버
  - 사용자 ID
  - 사용자 비밀번호
  -  을 누릅니다.
- **데이터베이스**를 선택한 경우:
  - **데이터 액세스 제공자**(Oracle, SqlServer 또는 OleDb)를 선택합니다.
    - \* 데이터베이스 연결 시간 초과 값 입력
    - \* 데이터베이스 명령 시간 초과 값 입력
  - **연결 문자열**을 입력합니다.
  - 사용자 ID 및 비밀번호를 입력합니다

 **주:**

쓰기 가능한 외부 연결을 설정하려면 관리자에게 SELECT, INSERT 및 DELETE 액세스 권한이 있어야 합니다. SELECT 액세스 권한만 있는 사용자는 테이블 및 뷰에 대한 읽기 전용 외부 연결을 설정할 수 있습니다.

-  을 누릅니다.
- **허용된 객체** 탭에서 큰 목록을 필터링하려면 다음 중 하나를 수행합니다.
  - \* 필요한 경우 와일드카드 문자를 사용하여 스키마/소유자를 선택하거나 입력합니다.
  - \* 필요한 경우 와일드카드 문자를 사용하여 객체 이름을 입력합니다.
  - \* **뷰 포함**을 선택하여 SELECT 이상의 권한이 있는 뷰를 포함합니다. 뷰는 항상 읽기 전용입니다.
  - \* **읽기 전용 테이블 포함**을 선택하여 SELECT 이상의 권한이 있지만 INSERT와 DELETE가 둘 다 포함되지 않은 테이블을 포함합니다.
  - \*  을 누르고 **사용가능** 목록에서 객체를 선택합니다. 화살표를 사용하여 객체를 **선택됨** 목록으로 이동합니다.
  - \* **선택사항:** **빠른 추가** 섹션을 사용하려면 추가할 객체의 이름 및 스키마/소유자를 입력하고 화살표를 눌러 **선택됨** 목록으로 이동합니다.
- 외부 작업을 추가하려면 **외부 작업** 탭을 누르고 **추가**를 누른 후 다음을 수행합니다.
  - \* 작업의 **이름**을 입력합니다. 상위 외부 연결에 고유한 이름이어야 합니다.
  - \* 작업의 목적을 설명하는 **설명** 텍스트를 입력합니다.
  - \* **작업 유형**(록업 또는 커밋)을 선택합니다. 이 선택에 따라 외부 록업 및 외부 커밋 기능에서 선택할 수 있는 작업 목록이 필터링됩니다.
  - \* **데이터베이스 작업 유형**(명령문 또는 저장된 프로시저)을 선택합니다.

- \* 명령문을 선택한 경우 **추가**를 누르고 다음을 수행합니다.
  - \* 작업을 호출할 때 전달되는 매개변수를 입력합니다.
    - \* **매개변수 이름** - 매개변수의 이름입니다. 공백은 사용할 수 없습니다.
    - \* **매개변수 설명** - 매개변수에 대한 설명입니다.
    - \* **테스트 값** - 작업 테스트에 사용되는 값입니다. 이 값은 재사용할 수 있도록 저장됩니다.
  - \* **SQL 문 필드**에 실행할 단일 SQL 문을 입력합니다. SQL 문에 대체 매개변수를 사용하여 런타임 값을 전달할 수 있습니다. 대체 매개변수 형식 지정은 <%ParamKey%>입니다. 여기서 <% 및 %>는 대체 매개변수를 나타내고 ParamKey는 대체에 사용할 매개변수의 이름입니다. 예를 들면 <%TopNode%>와 같습니다.
  - \*  를 눌러 작업을 테스트합니다. [롤백] 옵션은 스크립트에 의한 데이터베이스 변경사항을 모두 롤백합니다. [롤백]은 기본적으로 선택됩니다. 작업을 테스트하면 매개변수의 테스트 값이 명령문에 삽입되고 실행됩니다. **결과** 탭을 눌러 테스트 결과를 봅니다.
- \* **저장된 프로시저**를 선택한 경우:
  - \* 실행할 **저장된 프로시저 이름**을 입력합니다. 패키지 이름을 접두어로 포함할 수 있습니다.
  - \* 작업의 **이름**을 입력합니다. 상위 외부 연결에 고유한 이름이어야 합니다.
  - \* 작업의 목적을 설명하는 **설명** 텍스트를 입력합니다.
  - \* 저장된 프로시저의 매개변수 목록을 봅니다. [결과 매개변수]에 대해 True를 선택하여 Data Relationship Management 작업 결과에 매개변수를 반환합니다. 하나의 매개변수만 결과 매개변수로 선택할 수 있습니다. 결과 매개변수는 록업 작업에 대해서만 반환됩니다. 커밋 작업의 경우 성공 또는 실패만 표시됩니다.
  - \* **테스트 값** - 작업 테스트에 사용되는 값입니다. 이 값은 재사용할 수 있도록 저장됩니다.
  - \*  를 눌러 작업을 테스트합니다. [롤백] 옵션은 저장된 프로시저에 의한 데이터베이스 변경사항을 모두 롤백합니다. [롤백]은 기본적으로 선택됩니다. 작업을 테스트하면 매개변수의 테스트 값이 저장된 프로시저에 삽입되고 실행됩니다. **결과** 탭을 눌러 테스트 결과를 봅니다.
- **웹 서비스를 선택한 경우:**
  - **프로토콜**(HTTP 또는 HTTPS)을 선택합니다.
  - **호스트 이름**을 입력합니다.
  - **포트**를 입력합니다. 포트 0을 지정하면 표준 포트 80과 443이 각각 HTTP와 HTTPS에 사용됩니다.
  - **인증 유형**을 선택합니다. [기본]으로 설정하면 사용자 ID와 비밀번호를 저장할 수 있습니다.
  - **사용자 ID 및 비밀번호**를 입력합니다.
  - 외부 작업을 추가하려면 **추가**를 누르고 다음을 수행합니다.
    - \* 작업의 **이름**을 입력합니다. 상위 외부 연결에 고유한 이름이어야 합니다.
    - \* 작업의 목적을 설명하는 **설명** 텍스트를 입력합니다.

- \* **작업 유형**(록업 또는 커밋)을 선택합니다. 이 선택에 따라 외부 록업 및 외부 커밋 기능에서 선택할 수 있는 작업 목록이 필터링됩니다.
- \* **요청 탭**에서 **추가**를 누른 다음 작업을 호출할 때 전달되는 매개변수를 입력합니다.
  - \* **매개변수 이름** - 매개변수의 이름입니다. 공백은 사용할 수 없습니다.
  - \* **매개변수 설명** - 매개변수에 대한 설명입니다.
  - \* **테스트 값** - 작업 테스트에 사용되는 값입니다. 이 값은 재사용할 수 있도록 저장됩니다.
- \* **HTTP 작업**에서 GET, POST, PUT 또는 DELETE을 선택합니다.



주:

POST 및 PUT에서만 HTTP 본문 내용을 전송할 수 있습니다.

- \* 웹 서비스 메시지의 **HTTP URI**를 입력합니다.
- \* **HTTP 헤더**의 원시 콘텐츠를 입력합니다.
- \* **HTTP 본문**의 텍스트 콘텐츠를 입력합니다.
- \* **응답 탭** - 웹 서비스 작업에 대한 전체 송신 및 수신 메시지를 표시합니다. 송신 메시지에 사용된 매개변수의 테스트 값이 요청에 삽입됩니다. 웹 서비스에 의해 반환된 수신 메시지의 HTTP 본문은 XML 또는 JSON 형식이어야 합니다. 외부 록업 작업의 경우 수신 메시지를 외부 록업 등록정보에 사용하기 위해 테이블 형식(행과 열)으로 변환해야 합니다. 이 변환을 처리하기 위해 XPath 표현식을 사용할 수 있습니다. 목록 식별자 표현식 매개변수는 수신 메시지에서 결과 세트의 행이 되는 요소를 식별합니다. 결과 열은 결과 세트에서 열로 표시되는 행 요소의 속성을 식별합니다.

목록 식별자 표현식 및 결과 열 구성 결과를 미리 보려면 **미리보기** 탭을 누릅니다. 결과가 데이터 그리드에 표시됩니다.

URI, HTTP 헤더 및 HTTP 본문에 대체 매개변수를 사용하여 외부 작업에 런타임 값을 전달할 수 있습니다. 대체 매개변수 형식 지정은 `<%ParamKey%>`입니다. 여기서 `<%` 및 `>`는 대체 매개변수를 나타내고 `ParamKey`는 대체에 사용할 매개변수의 이름입니다. 예를 들면 `<%TopNode%>`와 같습니다.

구성을 테스트하려면 를 누릅니다. HTTP 요청이 작성되어 엔드포인트로 전송됩니다. 사용자 인터페이스가 **응답** 탭으로 자동으로 전환되고 전체 송신 메시지와 수신 응답이 표시됩니다. 송신 메시지에 사용된 매개변수의 테스트 값이 요청에 삽입됩니다.

7. 을 눌러 선택한 항목을 검증하고 연결 사용자 이름과 비밀번호를 통해 해당 레벨에서 액세스할 수 있는지 확인합니다.
8. 을 눌러 외부 연결을 저장합니다.

## 외부 연결 편집

외부 연결을 편집하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **메타데이터**에서 **외부 연결**을 확장합니다.
3. 외부 연결을 선택하고 을 누릅니다.
4. 필요에 따라 변경합니다.
5. 을 눌러 외부 연결을 저장합니다.

## 외부 연결 삭제

외부 연결을 삭제하면 해당 연결을 사용한 모든 임포트 및 익스포트 프로파일이 영향을 받습니다.

외부 연결을 삭제하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **메타데이터**에서 **외부 연결**을 확장합니다.
3. 외부 연결을 선택하고 을 누릅니다.
4. **이 항목 삭제**를 선택하여 삭제를 확인합니다.

# 16

## 관리 워크플로우 구성

관리 워크플로우는 노드, 관계 및 등록정보 값에 대한 변경사항의 입력, 승인, 검증 및 커밋을 제어하는 데 사용되는 공식화된 프로세스입니다.

애플리케이션 관리자는 워크플로우 태스크와 워크플로우 모델을 정의하여 비즈니스 사용자가 제출한 변경 요청과 데이터 관리자가 제출한 수정 요청을 관리합니다.

관리 워크플로우 개념에 대한 자세한 내용은 *Oracle Data Relationship Management 사용자 가이드*의 "관리 워크플로우"를 참조하십시오.

## 워크플로우 태스크 관리

워크플로우 태스크는 요청 컨텍스트 내에서 로컬 노드에 대해 사용자가 수행하는 단일 변경 세트입니다. 요청의 요청 항목은 워크플로우 태스크에서 제어됩니다.

워크플로우 태스크는 작업 유형, 사용자에게 대한 지침, 보거나 편집할 등록정보 및 검증으로 구성됩니다. 워크플로우 태스크의 작업 유형은 노드 추가, 이동 또는 업데이트와 같이 수행되는 작업의 기본 유형을 지정합니다. 각 작업 유형은 노드 및 상위 선택, 등록정보 업데이트 적용, 요청이 검증 및 커밋될 때 수행할 작업에 관한 규칙을 정의합니다.



주:

다음 작업은 워크플로우 요청에서 지원되지 않습니다.

- 노드 병합
- 노드 취소
- 노드 재활성화
- 단독 노드 삽입
- 도메인이 상위 멤버와 다른 도메인 노드 추가

## 태스크 등록정보

워크플로우 태스크 등록정보를 구성하여 요청 항목에 대해 표시되는 등록정보, 편집 가능 여부 및 값이 필수인지 여부를 제어할 수 있습니다. 편집 가능한 등록정보는 필수로 구성될 수 있습니다. 작업 유형의 기본 등록정보는 태스크에서 제거할 수 없습니다.

## 태스크 및 등록정보 지침

요청 항목의 생성, 보강 및 승인 과정을 사용자에게 안내하기 위해 요청 페이지에 지침을 추가할 수 있습니다. 워크플로우 태스크 및 해당 등록정보에 대한 지침을 정의할 수 있습니다. 제출, 강화 또는 커밋 단계에서 항목을 보는 경우 요청 항목의 시작 태스크에 대한 태스크 지침이 표시됩니다. 승인 또는 강화 단계에 지정된 업데이트 워크플로우 태스크에 대한 태스크 지침이 시작 태스크 지침 대신 표시됩니다. 개별 요청 항목 등록정보에 대한 태스크 등록정보 지침을 표시할 수 있습니다.

하이퍼링크는 태스크 및 등록정보 지침에 포함할 수 있습니다. URL을 지침 필드에 직접 삽입하거나 URL에 `[url=http_URL]URL_Title[/url]` 구문을 사용할 수 있습니다. 여기서 `http_URL`은 하이퍼링크 텍스트를 지정하고 `URL_Title`은 사용자에게 표시되는 텍스트를 지정합니다. 예를 들어 이 예에서 `[url=http://support.oracle.com]Oracle 지원[/url]`은 등록정보 그리드에 Oracle 지원으로 렌더링됩니다.

## 태스크 검증

태스크 검증은 특정 워크플로우 단계에 대해 요청을 제출 또는 승인하기 전에 요청 항목에 대해 성공적으로 실행되어야 하는 선택적 노드 레벨 검증입니다. 배치 모드로 실행되도록 구성된 검증을 태스크 검증으로 선택할 수 있습니다. 수정해야 할 수 있는 특정 등록정보에 검증 메시지를 연결하기 위해 태스크 등록정보에 태스크 검증을 연결할 수 있습니다.

## 계산된 이름 및 상위 멤버 등록정보

워크플로우 태스크에서 사용되는 이름 및 상위 멤버 등록정보를 통해 변경되는 노드 및 계층 위치를 알 수 있습니다. 이러한 등록정보 값은 주로 사용자가 수동으로 정의하거나 소스 파일에서 로드됩니다. 워크플로우 태스크에 사용 가능한 이름 계산 및 상위 멤버 계산 옵션을 사용하면 값을 명시적으로 정의하거나 로드하는 대신 동적 스크립트를 사용하여 이러한 등록정보 값을 계산할 수 있습니다.

이름 계산 옵션은 리프 추가 또는 림 추가 태스크 유형을 사용하는 워크플로우 태스크에 사용할 수 있습니다. 상위 멤버 계산 옵션은 삽입 및 이동 태스크뿐 아니라 다음 태스크에도 사용할 수 있습니다. 스크립트 계산 논리는 다음 데이터 소스에 액세스할 수 있습니다.

- NextID 함수
- 요청 버전 등록정보
- 버전의 계층 구조 및 해당 등록정보
- 노드 및 해당 등록정보
- 노드 간 계층 관계
- 요청 등록정보
- 요청 항목 및 해당 등록정보
- 요청 항목 태스크 및 해당 태스크 유형

이러한 옵션이 사용되고 있으며 워크플로우 태스크를 사용하여 요청 항목이 추가되는 단계에서 관리 요청이 계산되는 경우 이름 및 상위 멤버 등록정보 계산이 수행됩니다. 요청 항목의 시작 단계나 이러한 등록정보를 다시 계산하도록 구성된 이후 단계에서 값이 다시 계산될 수 있습니다.

### 주:

워크플로우 모델에서 재계산된 태스크 등록정보를 허용하도록 설정된 상태에서 계산된 이름 또는 상위 멤버가 수동으로 재정의되면 해당 단계 또는 이후 단계에서 이름 또는 상위 멤버가 다시 계산되지 않습니다.

## 외부 커밋

워크플로우 태스크에 외부 커밋을 선택적으로 구성하여 요청을 커밋할 때 관리 요청에서 승인된 변경사항을 외부 대상 시스템에 즉시 동기화할 수 있습니다. 예를 들어 외부 작업에서 데이터를 삽입, 업데이트 또는 삭제하는 SQL 문을 실행하거나 SOAP 또는 REST 웹 서비스를 호출하여 외부 시스템에서 데이터를 생성, 업데이트 또는 삭제할 수 있습니다. Oracle Data Relationship Governance에서 외부 커밋을 사용하는 경우 Data Relationship Governance 요청이 성공적으로 커밋된 후 외부 데이터 업데이트를 시작할 수 있습니다. 외부 데이터 소스는 데이터베이스 및 웹 서비스 연결에 대해 정의된 외부 작업을 사용하여 액세스합니다.

Data Relationship Governance 요청이 성공적으로 커밋된 후 각 항목의 태스크(task)에서 구성된 대로 각 항목에 대한 외부 태스크(operation)이 수행됩니다.

- 항목 및 태스크별로 정의된 순서에 따라 태스크가 동기적으로 수행됩니다.
- 요청 항목에 대한 로컬 노드의 컨텍스트에서 태스크(operation)이 수행되므로 출력 매개변수가 태스크(task)에 대해 선택되지 않은 등록정보를 기반으로 할 수 있습니다.
- 외부 작업 중에 오류가 발생하면 오류 메시지가 요청 항목에 외부 커밋 실패로 추가됩니다.
- 각 외부 작업 후에 요청 활동이 성공 또는 실패로 업데이트됩니다.
- 외부 작업에 대해 커밋 상태 등록정보를 정의한 경우 작업이 오류 없이 완료되면 해당 등록정보가 True로 업데이트되고, 작업이 완료되었지만 오류가 있으면 False로 업데이트됩니다.
- 외부 작업이 성공적으로 완료되지 않은 경우 데이터 관리자 및 커밋 단계 참가자가 통지를 받습니다.

## 워크플로우 태스크 생성

워크플로우 태스크를 생성하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **새로 작성**에서 **워크플로우 태스크**를 선택합니다.
3. 워크플로우 태스크의 이름을 입력합니다.
4. **작업 유형**에서 다음 태스크의 작업 유형을 선택합니다.
  - **리프 추가** – 글로벌 및 로컬 등록정보가 포함된 리프 노드를 추가합니다.
  - **림 추가** – 글로벌 및 로컬 등록정보가 포함된 림 노드를 추가합니다.
  - **삭제** – 노드의 글로벌/로컬 등록정보를 업데이트하고 노드를 삭제합니다.
  - **비활성화** – 노드의 글로벌 및 로컬 등록정보를 업데이트하고 노드를 비활성화합니다.
  - **삽입** – 노드를 계층에 삽입하고 글로벌/로컬 등록정보를 업데이트합니다.
  - **이동** – 노드를 다른 상위 멤버로 이동하고 해당 글로벌/로컬 등록정보를 업데이트합니다.
  - **제거** – 노드의 글로벌/로컬 등록정보를 업데이트하고 노드를 제거합니다.
  - **업데이트** – 노드의 글로벌 및 로컬 등록정보를 업데이트합니다.

 주:

사용자가 파일에서 요청으로 항목을 업로드하는 경우 태스크에서 다음 등록정보를 정의(하고 사용자가 파일을 업로드)해야 합니다.

- 추가 작업: 이름, 상위, 설명
- 삽입 작업: 이름, 상위
- 이동 작업: 이름, 상위

- **재활성화** – 노드의 글로벌 및 로컬 등록정보를 업데이트하고 비활성 노드를 재활성화합니다.
5. **선택 사항:** 다음 태스크 중 하나를 수행합니다.
- **지침 필드에 사용자에 대한 텍스트를 입력합니다.**  
URL을 지침 필드에 직접 삽입하거나 URL에 [url=http\_URL]URL\_Title[/url] 구문을 사용할 수 있습니다. 여기서 http\_URL은 하이퍼링크 텍스트를 지정하고 URL\_Title은 사용자에게 표시되는 텍스트를 지정합니다. 예를 들어 이 예에서 [url=http://support.oracle.com]Oracle 지원[/url]은 등록정보 그리드에 Oracle 지원으로 렌더링됩니다.
  - 필터링할 계층 그룹을 선택합니다.

 주:

워크플로우 태스크에 대해 선택한 계층 그룹은 워크플로우 모델에 대해 구성된 계층 그룹 등록정보와 함께 태스크에 선택할 수 있는 계층을 필터링하는 데 사용됩니다.

- 리프 추가 또는 리프 추가 태스크를 위한 노드의 **도메인**을 선택합니다.

 주:

워크플로우 태스크에 대해 구성된 도메인은 해당 태스크를 사용하는 요청 항목의 대상 버전에서 사용되는 도메인과 일치해야 합니다. 버전에서 태스크 도메인이 사용되지 않으면 요청 항목 노드를 해당 버전에 추가할 수 없습니다.

 주:

도메인이 지정된 경우 리프 추가 및 리프 추가 태스크의 설명 등록정보가 필요합니다.

6. 등록정보 탭의 **사용가능** 목록에서 태스크에 지정할 등록정보를 선택합니다. 화살표를 사용하여 등록정보를 **선택됨** 목록으로 이동합니다. 위쪽 및 아래쪽 화살표를 사용하여 등록정보의 순서를 지정합니다.
7. 등록정보에 대한  을 눌러 다음 옵션을 업데이트합니다.

- **편집 가능** – 등록정보를 편집할 수 있게 하려면 선택합니다.
- **필수** – 등록정보를 필수로 설정하려면 선택합니다.
- **계산** – 리프 추가 또는 리프 추가 태스크의 경우 동적 스크립트에서 이름 값을 계산하도록 선택합니다. 선택한 경우 이름 등록정보의 편집 가능 옵션이 False이며 비활성화됩니다. 이 옵션을 선택하면 **이름 계산** 탭을 사용할 수 있게 되며, 이름 값을 계산하기 위한 스크립트를 입력할 수 있습니다.

동적 스크립트에서 리프 추가, 리프 추가, 이동 및 삽입 태스크에서 상위 값을 계산하려면 상위 노드 옆에 있는  을 누르고 **계산**을 선택합니다. 선택된 경우 상위 멤버 등록정보의 편집 가능 옵션이 False이고 비활성화됩니다. 이 옵션을 선택하면 **상위 멤버 계산** 탭을 사용할 수 있게 되며, 상위 멤버 값을 계산하기 위한 스크립트를 입력할 수 있습니다. 동적 스크립트 작성에 대한 자세한 내용은 [동적 스크립트 관리](#)를 참조하십시오.

- **사용자정의 레이블** – **선택 사항:** 등록정보의 대체 레이블을 입력합니다. 이 레이블은 항목 세부정보의 등록정보 레이블 옆에 표시됩니다.
- **등록정보 지침** – **선택 사항:** 등록정보에 대한 특정 지침을 입력합니다. 등록정보가 편집 가능하지 않아도 지침을 추가할 수 있습니다. 지침은 항목 세부정보에서 등록정보 값 위에 표시됩니다.

URL을 지침 필드에 직접 삽입하거나 URL에 `[url=http_URL]URL_Title[/url]` 구문을 사용할 수 있습니다. 여기서 `http_URL`은 하이퍼링크 텍스트를 지정하고 `URL_Title`은 사용자에게 표시되는 텍스트를 지정합니다. 예를 들어 이 예에서 `[url=http://support.oracle.com]Oracle 지원[/url]`은 등록정보 그리드에 Oracle 지원으로 렌더링됩니다.

 을 눌러 변경사항을 저장하거나  을 눌러 변경사항을 취소합니다.

8. **검증** 탭의 **사용가능** 목록에서 태스크에 지정할 검증을 선택합니다. 화살표를 사용하여 검증을 **선택됨** 목록으로 이동합니다.
9. 특정 태스크 등록정보에 검증을 연계하려면  을 누릅니다. 선택한 검증이 실패하면 지정한 등록정보에 대한 검증 메시지가 표시됩니다.

 을 눌러 변경사항을 저장하거나  을 눌러 변경사항을 취소합니다.

10. 이름 또는 상위 멤버를 계산하도록 선택한 경우 **이름 계산** 또는 **상위 멤버 계산** 탭을 선택하고 다음을 수행합니다.
  - 동적 스크립트를 입력하여 이름 또는 상위 멤버를 계산합니다. 동적 스크립트 작성에 대한 자세한 내용은 [동적 스크립트 생성](#)을 참조하십시오.
  - 다음 정보를 입력합니다.
    - **요청 ID**—스크립트를 평가할 때 사용할 요청 ID를 지정합니다.
    - **요청 항목 번호**—스크립트를 평가할 때 사용할 요청 항목 번호를 지정합니다.
    - **스크립트 시간 초과**—스크립트가 시간 초과될 때까지의 초 수입니다.
  - **선택 사항:** **숨김**을 선택하여 계산할 이름 또는 상위 멤버의 숨김 등록정보를 지정합니다. 선택된 경우 계산된 이름 또는 상위 멤버가 요청 항목 세부정보에 표시되지 않습니다.
  - **평가**를 누릅니다. 결과가 스크립트 디자이너의 맨아래에 표시됩니다.
11. **선택사항:** **외부 커밋** 탭을 선택하고 **추가**를 누른 후 다음 설정을 구성합니다.
  - **외부 연결** - 외부 연결을 선택합니다.
  - **작업** - 수행할 외부 작업을 선택합니다.

 주:

작업은 연결에서 커밋 유형 작업으로 정의된 상태여야 합니다.

- 각 외부 작업 매개변수에 대해 다음을 구성합니다.
  - 매개변수 소스 유형 - [리터럴] 또는 [등록정보]를 선택합니다
  - 소스 - 소스 유형에 대해 리터럴을 선택한 경우 [매개변수 소스] 옆에 리터럴 값을 입력합니다. 외부 작업을 호출하면 현재 매개변수에 대한 리터럴 값이 전달됩니다. 소스 유형에 대해 등록정보를 선택한 경우 외부 작업에 대한 매개변수 값을 제공할 등록정보를 선택합니다. 외부 커밋을 실행하면 현재 노드 또는 요청 항목에서 선택한 등록정보의 매개변수 값이 전달됩니다.
- 커밋 상태 등록정보 - 노드에서 외부 커밋 오류가 발생했는지 여부를 나타내는 부울 등록정보를 선택합니다. 이 등록정보는 요청의 대상 버전에 있는 노드에 대해 설정됩니다. 외부 커밋에 실패할 경우 이 등록정보를 사용하여 버전에서 외부 시스템에 성공적으로 커밋되지 않은 변경사항을 파악할 수 있습니다.

12.  을 눌러 워크플로우 태스크를 저장합니다.

## 워크플로우 태스크 편집

워크플로우 태스크에 대한 등록정보 및 검증 목록은 태스크가 생성된 후에 편집할 수 있습니다. 태스크가 저장된 후에는 워크플로우 태스크에 대한 작업 유형을 수정할 수 없습니다.

워크플로우 태스크에 대한 태스크 등록정보를 추가 또는 제거하거나 편집 가능에서 읽기 전용으로 변경하거나 순서를 조정하면 기존 요청에 대한 요청 항목 등록정보가 영향을 받습니다. 제거된 태스크 등록정보는 해당 태스크를 사용하는 요청 항목에 대해 더 이상 표시되지 않습니다. 편집 가능에서 읽기 전용으로 변경된 태스크 등록정보를 사용하여 요청 항목에 대해 정의된 등록정보 값은 무시됩니다.

워크플로우 태스크를 편집하려면 다음을 수행합니다.

1. 홈 페이지에서 관리를 선택합니다.
2. 워크플로우에서 워크플로우 태스크를 확장합니다.
3. 태스크를 선택하고  을 누릅니다.
4. 등록정보 및 검증 탭에서 등록정보 및 검증 선택을 변경합니다.
5.  을 누릅니다.

## 워크플로우 태스크 복사

기존 태스크를 복사하여 워크플로우 태스크를 생성할 수 있습니다. 작업 유형, 등록정보 및 검증이 복사되며, 저장 전에 편집할 수 있습니다.

워크플로우 태스크를 복사하려면 다음을 수행합니다.

1. 홈 페이지에서 관리를 선택합니다.
2. 워크플로우에서 워크플로우 태스크를 확장합니다.
3. 복사할 태스크를 마우스 오른쪽 버튼으로 누르고 복사를 선택합니다.

4. 태스크의 새 이름을 입력합니다.
5. 태스크에 대해 다른 변경을 수행하고  을 눌러 워크플로우 태스크를 저장합니다.

## 워크플로우 태스크 삭제

워크플로우 태스크가 변경 요청에 지정된 모델에 지정되어 있지 않으면 삭제될 수 있습니다. 삭제할 수 없는 모델에 태스크가 지정된 경우에는 태스크를 삭제할 수 없습니다.

워크플로우 태스크를 삭제하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. 워크플로우에서 **워크플로우 태스크**를 확장합니다.
3. 워크플로우 태스크를 선택하고  을 누릅니다.
4. 워크플로우 태스크 삭제를 눌러 삭제를 확인합니다.

## 워크플로우 모델 관리

워크플로우 모델은 해당 모델에 따라 단일 요청에 함께 포함될 수 있는 정의된 유형의 변경 관리 태스크 세트를 정의합니다. 모델은 변경사항이 버전으로 커밋될 수 있기 전에 필요한 승인 및 강화 단계 세트를 정의합니다.

## 워크플로우 단계

워크플로우 단계는 각 워크플로우 모델에 대해 정의되며 워크플로우 모델에 걸쳐 공유될 수 없습니다.

### 단계 유형

단계가 워크플로우 모델에 지정되면 단계 유형 속성이 워크플로우의 해당 단계에 있는 사용자의 참여 유형을 정의합니다.

표 16-1 워크플로우 단계 유형

워크플로우 단계 유형	설명	작업 유형
제출	<p>제출 단계는 요청에 포함할 초기 요청 항목을 정의하는 데 사용됩니다. 여러 워크플로우 태스크를 이 단계 유형에 연결할 수 있습니다. 제출 단계 중에 요청 항목 하나 이상을 요청에 추가해야 합니다.</p> <p>종속 워크플로우 태스크를 사용하여 리프 추가 또는 리프 추가 태스크를 선택적으로 구성할 수 있습니다. 시스템에서 원래 워크플로우 태스크에 대한 요청 항목과 각 종속 태스크에 대한 추가 요청 항목을 추가합니다.</p> <p>기본 태스크는 종속 태스크가 될 수 없습니다. 기본 태스크에 대한 추가 항목의 이름을 계산할 때 기본 태스크와 종속 태스크는 관련 그룹으로 간주됩니다. 이름 계산이 대기 중인 동안 기본 태스크를 삭제하면 추가가 아닌 종속 태스크도 삭제됩니다.</p> <p><b>참고:</b> 각 요청에 하나의 제출 단계만 있습니다. 이 단계의 워크플로우 단계 기준을 정의할 수 없습니다.</p>	<ul style="list-style-type: none"> <li>• 리프 추가</li> <li>• 리프 추가</li> <li>• 업데이트</li> <li>• 비활성화</li> <li>• 삽입</li> <li>• 이동</li> <li>• 제거</li> <li>• 삭제</li> </ul>
강화	<p>강화 단계는 제출 단계에서 추가된 요청 항목을 업데이트하거나 요청 항목을 추가하는 데 사용됩니다. 이 단계의 워크플로우 단계 기준을 정의할 수 있습니다.</p> <p>강화 단계에는 단일 워크플로우 태스크가 연결됩니다. 일반 강화 단계에서는 워크플로우 태스크를 기존 요청 항목에 대한 업데이트 작업과 함께 사용합니다. 그러나 일부 강화 단계에서는 다음과 같이 추가 라인 항목을 생성해야 할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• 단일 노드를 여러 계층 구조에 삽입</li> <li>• 여러 계층 구조의 단일 노드에 대한 로컬 등록정보 업데이트</li> </ul> <p>이 단계는 제출 단계와 커밋 단계 사이에 수행됩니다.</p> <p><b>참고:</b> 워크플로우 모델에 대해 원하는 개수의 강화 단계를 정의할 수 있습니다.</p>	<ul style="list-style-type: none"> <li>• 업데이트(기존 요청 항목)</li> <li>• 삽입(새 항목 추가)</li> <li>• 이동(새 항목 추가)</li> <li>• 제출 단계에 사용할 수 있는 모든 작업 유형</li> </ul>

표 16-1 (계속) 워크플로우 단계 유형

워크플로우 단계 유형	설명	작업 유형
승인	<p>승인 단계는 제출 단계에서 추가되거나 강화 단계 중이 추가 또는 업데이트된 모든 요청 항목을 보고 승인하는 데 사용됩니다. 사용자는 승인 단계 중에 요청 항목을 추가하거나 편집할 수 없습니다. 이 단계의 워크플로우 단계 기준을 정의할 수 있습니다.</p> <p>승인 단계는 단일 워크플로우 태스크를 사용하여 요청이 단계에 있는 동안 등록정보를 보고 요청 항목에 대한 검증을 실행합니다. 업데이트 태스크는 승인 단계에서 읽기 전용 모드로 사용할 수 있습니다. 중간 단계에서 요청 항목의 등록정보를 업데이트하려면 강화 단계 유형을 대신 사용합니다.</p> <p>이 단계는 제출 단계와 커밋 단계 사이에 수행됩니다.</p> <p><b>참고:</b> 워크플로우 모델에 대해 원하는 개수의 승인 단계를 정의할 수 있습니다.</p>	업데이트(기존 요청 항목)
커밋	<p>커밋 단계는 대상 버전에 대한 요청에서 요청 항목의 커밋을 트리거하기 위해 요청의 최종 승인을 제공하는 데 사용됩니다. 커밋하는 사용자는 요청의 모든 요청 항목을 승인해야 합니다. 이 단계의 워크플로우 단계 기준을 정의할 수 있지만 이 단계에서 요청을 분할할 수는 없습니다.</p> <p>커밋 단계에는 연결된 워크플로우 태스크가 없습니다. 대신에 커밋 단계에서는 등록정보의 수퍼 세트를 표시하고 이전 제출 및 강화 단계의 요청 항목에 사용할 수 있는 검증 수퍼 세트를 실행합니다. 커밋 단계의 사용자는 최종 조정을 허용하기 위해 요청 항목에 대해 표시되는 편집 가능한 등록정보를 업데이트할 수 있습니다.</p> <p>최종 워크플로우 단계입니다.</p> <p><b>참고:</b> 각 요청에 하나의 커밋 단계만 있습니다.</p>	해당 없음

### 단계 조건

단계 조건을 사용하여 요청의 항목에 대해 평가되는 지정한 기준을 기반으로 특정 요청의 워크플로우 경로를 변경할 수 있습니다. 단계에 대한 조건을 설정하고 조건이 충족될 경우 수행할 작업(예: 요청이 단계를 시작할 수 있는지 여부 또는 일부 요청 항목이 별도의 요청으로 분할되는지 여부)을 선택합니다. 다음 기준에 따라 워크플로우 단계 조건을 평가할 수 있습니다.

- **등록정보 기준** - 단계에 대한 단계 기준으로 평가할 리터럴 값과 등록정보 쿼리 연산자를 사용합니다.

- **선택한 검증** – 단계에 대한 단계 기준으로 실행할 검증을 하나 이상 선택합니다. 승인, 강화 또는 커밋 단계에 대해 이 옵션을 선택할 수 있습니다.
- **태스크 검증** – 워크플로우 태스크에 지정된 검증 실패입니다. 선택된 경우 태스크에 지정된 검증도 단계에 대한 단계 기준으로 실행됩니다. 승인 또는 강화 단계에 대해 이 옵션을 선택할 수 있습니다. 단계에 지정된 태스크에 검증이 지정되어 있지 않으면 이 옵션을 사용할 수 없습니다.

워크플로우 단계에 대한 단계 조건을 충족하는 요청 항목이 있는 경우 다음 작업 중 하나가 수행될 수 있습니다.

- **단계 입력** – 승인, 강화 또는 커밋 단계의 경우 요청이 단계의 사용자에게 지정됩니다. 요청이 단계를 시작하고 해당 단계에 대해 워크플로우 처리가 계속됩니다.
- **요청 항목 분할** – 승인 또는 강화 단계의 경우 단계 조건을 충족하는 요청이 동일한 워크플로우 모델을 사용하는 별도의 제출된 요청으로 이동됩니다. 새 요청이 워크플로우 단계를 시작하고 단계의 사용자에게 지정됩니다. 단계 조건을 충족하지 않는 항목은 원래 요청에 유지되고 원래 요청에 대해 단계를 건너뛸 것입니다. 모든 요청 항목이 단계 기준을 만족하면 요청이 분할되지 않고 분할 단계가 시작됩니다.

요청 항목이 워크플로우 단계에 대한 단계 조건을 충족하지 않는 경우 단계를 건너뛰며 요청이 워크플로우 모델의 다음 단계로 이동합니다.

### 승인 방법

요청의 단계를 승인해야 하는 사용자를 선택합니다.

- **임의 그룹** – 지정된 노드 액세스 그룹의 모든 사용자가 요청을 다음 워크플로우 단계로 진행하기 위해 승인할 수 있습니다. 현재 단계 유형 이상에 대한 액세스 권한이 있는 계층에 노드 액세스 그룹을 지정해야 합니다. 단계에 지정된 액세스 그룹에 요청에 포함된 요청 항목에 대한 적절한 데이터 액세스가 없는 경우 필요한 값이 제공되고 모든 요청 항목에 대해 검증이 통과되기만 하면 단계를 건너뛸 수 있습니다.
- **모든 그룹** – 모든 지정된 노드 액세스 그룹의 사용자 한 명 이상이 요청을 승인해야 합니다. 단계에 지정된 액세스 그룹에 요청에 포함된 요청 항목에 대한 적절한 데이터 액세스가 없는 경우 요청이 해결을 위해 데이터 관리자에게 상향 조정됩니다.

### 재승인

요청이 이전 단계로 반송되고 요청 항목이 반송되는 동안 수정될 경우 원래 요청에 대한 초기 승인을 제공한 사용자가 요청에 대한 변경을 재승인해야 할 수 있습니다. 이 옵션은 반송 모드 중엔 각 단계에서 수행된 변경을 다른 사용자가 재승인해야 할지 여부를 결정합니다. 다음 옵션 중 하나를 선택합니다.

- **현재** – 이 단계의 요청에 대한 변경을 현재 단계에 대해서만 재승인해야 합니다. 승인 후 이전에 요청을 반송한 사용자에게 요청이 지정됩니다.
- **모두** – 이 단계의 요청에 대한 변경을 후속 단계에 대해 재승인해야 합니다.

### 직무 분리

요청의 다른 단계를 제출 또는 승인하지 않은 별도의 승인 사용자가 필요하도록 워크플로우 단계를 선택적으로 구성할 수 있습니다. [직무 분리] 옵션이 활성화된 경우 다른 워크플로우 단계를 제출 또는 승인한 사용자는 옵션이 활성화된 단계의 요청을 클레임할 수 없습니다. 다음 예외 사항에 유의하십시오.

- 제출자는 제출 단계로 반송된 요청을 클레임할 수 있습니다.
- 단계의 이전 승인자는 승인 또는 강화 단계로 반송된 요청을 클레임할 수 있습니다.

- 데이터 관리자 역할 사용자는 이전 승인에 관계없이 지정된 모든 요청을 클레임할 수 있습니다.

### 통지

통지에는 웹 클라이언트 알림과 전자메일 통지가 모두 포함됩니다. 워크플로우 단계마다 워크플로우 사용자에게 알림과 통지가 전송되는 경우 및 시기를 설정할 수 있습니다. 통지는 단계의 통지 설정과 통지를 트리거한 워크플로우 이벤트 유형에 따라 특정 사용자에게 필터링되어 표시됩니다.



#### 주:

사용자는 본인이 수행하는 작업에 대한 통지를 받지 않습니다.

각 단계에 대해 다음 통지 옵션을 선택하십시오.

- **없음** – 이 워크플로우 단계에서 수행된 작업에 대한 통지가 사용자에게 제공되지 않습니다.
- **담당자** – 지정, 승인, 커밋 또는 거부 작업이 발생하면 요청에 현재 지정된 워크플로우 노드 액세스 그룹에 속하는 사용자에게 통지가 제공됩니다.

[통지] 설정이 [담당자] 또는 [담당자 및 참가자]인 단계에 지정된 워크플로우 액세스 그룹에 속한 담당자에게만 통지가 제공됩니다.

- **참가자**

- 커밋 또는 거부 작업이 발생하면 요청을 제출하거나 클레임한 사용자에게 통지가 제공됩니다.
- 승인 또는 승격 작업이 발생하면 요청을 제출한 사용자에게 통지가 제공됩니다.

[통지] 설정이 [참가자] 또는 [담당자 및 참가자]인 단계에 지정된 워크플로우 노드 액세스 그룹에 속한 참가자에게만 통지가 제공됩니다.

- **담당자 및 참가자** – 담당자와 참가자에게 통지가 제공됩니다.

다음 테이블에서는 각 단계의 통지 설정에 따라 통지를 트리거하는 작업 및 통지 수신자를 보여 줍니다.

**표 16-2** 워크플로우 알림

워크플로우 작업	통지 받는 사람			
	담당자	제출자	참가자	사용자 통지
지정	X			
승인	X	X		X
승격		X		X
에스컬레이션	X			X
거부	X		X	X
커밋	X		X	X

 주:

통지 사용자는 요청 항목에 대한 통지 액세스 권한만 사용하여 단계에 지정된 워크플로우 노드 액세스 그룹의 멤버인 사용자입니다. [통지] 설정이 [담당자] 또는 [담당자 및 참가자]인 경우에만 통지를 받습니다. [통지] 옵션이 [없음] 또는 [참가자]이면 이러한 사용자에게 알리지 않습니다

### 종속 워크플로우 태스크

종속 워크플로우 태스크를 사용하여 다른 태스크가 수행될 때 관리 요청의 워크플로우 태스크를 자동으로 수행할 수 있습니다. 예를 들어 요청을 커밋할 때 모든 계층 구조에서 동기화되도록 노드가 추가될 때 다른 계층 구조에 노드를 삽입할 수도 있습니다. 리프 추가 및 리프 추가 태스크(action) 유형을 사용하여 기본 워크플로우 태스크(task)에 대한 종속 태스크(task)를 구성할 수 있습니다.

요청에 요청 항목을 추가할 때 항목에 대해 선택한 태스크가 기본 태스크입니다. 종속 태스크와 함께 기본 태스크를 구성하면 각 종속 태스크에 대한 추가 요청 항목이 자동으로 요청에 추가됩니다.

## 모델 필터

사용자가 특정 요청 유형에 대해 보고 선택할 수 있는 버전, 계층 구조 및 노드 유형을 제한할 수 있습니다.

- **버전 변수** - 특정 워크플로우 모델의 요청에서 요청 항목에 대한 버전 선택을 제한합니다.
- **계층 그룹 등록정보** - 특정 워크플로우 모델의 요청에서 요청 항목에 대해 노드를 선택할 수 있는 계층을 제한합니다.
- **계층 그룹** - 계층 그룹 등록정보가 지정된 경우 필요합니다.
- **노드 유형** - 특정 워크플로우 모델의 요청에 요청 항목으로 추가할 수 있는 노드를 제한합니다.

## 요청 및 클레임 기간

요청에 대한 워크플로우 모델을 요청 또는 클레임 기간 간격으로 구성하여 특정 요청 유형에 필요한 예상 시간에 따라 관리 워크플로우에 의한 요청의 자동 처리를 제어할 수 있습니다.

- **요청 기간** - 요청을 승인 및 커밋하는 데 걸리는 예상 일수를 나타냅니다. 요청의 경과 기간이 요청 기간을 초과하면 요청이 기한 경과로 표시됩니다.
- **클레임 기간** - 관리 사용자가 워크플로우 단계에 대해 요청을 클레임해야 하는 예상 일수를 나타냅니다. 요청의 경과 기간이 클레임 기간을 초과하면 요청이 자동으로 클레임 취소되어 다른 지정된 사용자가 클레임할 수 있게 됩니다.

 주:

한쪽 옵션의 값이 0이면 기한 경과 및 자동 클레임 취소 기능이 워크플로우 모델에 대해 비활성화됩니다.

## 워크플로우 모델 생성

워크플로우 모델을 생성하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. **새로 작성**에서 **워크플로우 모델**을 선택합니다.
3. 워크플로우 모델의 이름, 레이블 및 설명을 입력합니다.

이름은 워크플로우 모델의 고유 이름입니다. 레이블은 워크플로우 모델의 레이블이며 이름과 같을 수 있습니다. 설명은 선택 사항입니다.

URL을 설명 필드에 직접 삽입하거나 URL에 `[url=http_URL]URL_Title[/url]` 구문을 사용할 수 있습니다. 여기서 `http_URL`은 하이퍼링크 텍스트를 지정하고 `URL_Title`은 사용자에게 표시되는 텍스트를 지정합니다. 예를 들어 이 예에서 `[url=http://support.oracle.com]Oracle 지원[/url]`은 등록정보 그리드에 Oracle 지원으로 렌더링됩니다.

4. **선택사항: 요청 기간 및 클레임 기간**에 대해 일수를 입력합니다.
5. 워크플로우 단계 탭에서 단계를 두 번 누르거나(제출 또는 커밋) **단계 추가**를 누릅니다.
6. 단계 탭에서 다음 옵션을 구성합니다. 이러한 옵션에 대한 자세한 내용은 **워크플로우 단계**를 참조하십시오.
  - **레이블** - 단계의 레이블을 입력합니다. 모델에 대한 요청이 있는 경우에도 언제든지 단계 레이블을 편집할 수 있습니다.
  - **유형** - 단계 유형을 선택합니다. 모델에 대한 요청이 있을 때까지 단계 유형을 편집할 수 있으며, 그 후에는 변경할 수 없습니다.
  - **워크플로우 방법** - 요청의 단계를 승인해야 하는 노드 액세스 그룹을 지정합니다.
  - **재승인** - 현재 단계에서만 수행되었거나 모든 단계에서 수행된 변경사항에 재승인이 필요한지 여부를 지정합니다.
  - **통지** - 통지와 알림을 보낼 대상을 지정합니다.
  - **직무 분리** - 요청의 다른 단계를 제출 또는 승인하지 않은 별도의 승인 사용자가 필요하게 하려면 선택합니다.
  - **태스크 등록정보 재계산** - 외부 룩업 등록정보에 사용하거나 계산된 이름 또는 상위 멤버 값을 재계산할 수 있게 하려면 선택합니다. 이후 워크플로우 단계에서 데이터가 요청 항목의 최종 이름 또는 상위 멤버를 계산하는 데 사용되는 입력인 경우 이 옵션이 필요합니다.

### 주:

워크플로우 모델에서 재계산된 태스크 등록정보를 허용하도록 설정된 상태에서 계산된 이름 또는 상위 멤버가 수동으로 재정의되면 해당 단계 또는 이후 단계에서 이름 또는 상위 멤버가 다시 계산되지 않습니다.

7. 제출 단계 태스크의 경우에만 **태스크** 탭에서 단계 태스크를 구성합니다.
  - 왼쪽 및 오른쪽 화살표 버튼을 사용하여 단계에 지정할 태스크를 선택합니다.
  - 위쪽 및 아래쪽 화살표 버튼을 사용하여 태스크를 원하는 순서대로 배치합니다.

- 태스크가 종속 태스크인 경우 종속되는 기본 태스크를 설정해야 합니다. 종속 태스크에 대해  을 누르고 [기본 태스크] 드롭다운 목록에서 기본 태스크를 선택합니다.

 주:

림 추가 또는 리프 추가 태스크만 기본 태스크로 설정할 수 있습니다. 기본 태스크는 숨길 수 없으며 종속 태스크가 될 수도 없습니다.

- 숨김 - 종속 태스크에 대해 선택하면 [항목 추가] 대화상자에서 해당 태스크가 요청 내에 표시되지 않습니다.

 주:

모델에 대한 요청이 있을 때까지 선택한 태스크를 편집할 수 있으며, 그 후에는 변경할 수 없습니다.

8. **노드 액세스 그룹** 탭에서 워크플로우 단계에 연결할 워크플로우 노드 액세스 그룹을 선택합니다.  
워크플로우 유형의 노드 액세스 그룹만 단계에 지정할 수 있습니다.
9. **선택사항:** 워크플로우 단계에 대한 기준을 추가하려면 **조건** 탭에서 조건 유형을 선택하고 수행할 작업을 선택한 다음  을 누릅니다.
  - 유형
    - **등록정보 기준** - 단계에 대한 단계 기준으로 평가할 등록정보를 하나 이상 선택합니다. **추가**를 눌러 기준 행을 삽입합니다. 행에 대한 **등록정보 및 연산자**를 선택하고 **값**을 입력합니다.
    - **선택한 검증** - 단계에 대한 단계 기준으로 실행할 검증을 하나 이상 선택합니다. 화살표를 눌러 검증을 **선택됨** 목록으로 이동합니다.
    - **태스크 검증** - 단계 기준으로 태스크에 지정된 검증을 실행하려면 선택합니다.
  - **작업** - 단계 기준이 충족될 때 워크플로우 단계에 대해 수행할 작업(단계 입력 또는 요청 항목 분할)을 선택합니다. 자세한 내용은 **단계 조건**를 참조하십시오.
10.  을 눌러 워크플로우 단계를 저장합니다.
11. **선택사항:** **필터** 탭에서 항목을 선택하여 사용자가 특정 요청 유형에 대해 보고 선택할 수 있는 버전, 계층 구조 및 노드 유형을 제한합니다.
12. **선택사항:** **단계 추가**를 눌러 강화 또는 승인 단계를 워크플로우 모델에 추가한 다음 추가된 각 단계에 대해 6-8단계를 수행합니다.
13.  을 눌러 워크플로우 모델을 저장합니다.

## 워크플로우 모델 편집

워크플로우를 처리하는 중 기준 요청이 부정적인 영향을 받지 않고 요청이 완료된 후 해당 콘텐츠가 변경되지 않도록 요청이 생성된 워크플로우 모델은 특정 편집이 제한됩니다. 변경 요청이 있는 모델의 경우 다음과 같은 편집 제한이 적용됩니다.

- 모델에 대한 워크플로우 단계는 추가, 제거 또는 순서를 조정할 수 없습니다.
- 단계에 대한 단계 유형은 변경할 수 없습니다.
- 모델에 대한 워크플로우 단계의 태스크는 변경할 수 없습니다.

워크플로우 모델을 편집하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. 워크플로우에서 **워크플로우 모델**을 확장합니다.
3. 모델을 선택하고  을 누릅니다.
4. 워크플로우 모델을 변경하고  을 누릅니다.

## 워크플로우 모델 복사

기존 모델을 복사하여 워크플로우 모델을 생성할 수 있습니다. 모든 워크플로우 단계, 모델 필터 및 기간 설정이 복사되며, 저장 전에 편집할 수 있습니다. 이후 요청을 다르게 처리하기 위해 현재 요청에 사용되는 기존 워크플로우 모델을 편집해야 하는 경우 모델을 복사할 수 있으며 새 모델을 변경합니다. 그런 다음 편집한 모델 복사본을 새로 생성한 요청에 사용할 수 있습니다.

워크플로우 모델을 복사하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. 워크플로우에서 **워크플로우 모델**을 확장합니다.
3. 복사할 모델을 선택하고  을 누릅니다.
4. 모델의 새 이름을 입력합니다.
5. 모델에 대해 다른 변경을 수행하고  을 눌러 워크플로우 모델을 저장합니다.

## 워크플로우 모델 이름 바꾸기

시간에 따라 다른 워크플로우 요구사항을 지원하려면 워크플로우 모델을 복사하여 구성에 편집 내용을 적용할 수 있습니다. 이 경우 모델 복사본의 이름을 관리 사용자에게 이미 익숙한 원본 워크플로우의 이름과 일치하도록 바꿀 수 있습니다.

워크플로우 모델의 이름을 바꾸려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. 워크플로우에서 **워크플로우 모델**을 확장합니다.
3. 이름을 바꿀 모델을 선택하고  을 누릅니다.
4. 모델의 새 이름을 입력하고  을 누릅니다.

## 워크플로우 모델 숨기기

워크플로우 모델을 숨겨 사용자가 해당 모델을 사용하여 새 요청을 생성하지 못하도록 할 수 있습니다. 워크플로우 모델을 숨기기 전에 생성된 기존 요청은 완료될 때까지 모델을 계속 진행합니다. 원본 모델을 바꾸기 위해 워크플로우 모델을 복사하고 수정하는 경우 하나의 모델 인스턴스만 새 요청에 사용할 수 있도록 원본 모델을 숨길 수 있습니다.

 주:

숨기도록 선택한 워크플로우 모델을 사용한 요청은 완료될 때까지 해당 프로세스 플로우를 계속합니다.

워크플로우 모델을 숨기려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. 워크플로우에서 **워크플로우 모델**을 확장합니다.
3. 숨길 모델을 선택하고  을 누릅니다.
4. **숨김**을 선택하고  을 누릅니다.

## 워크플로우 모델 삭제

연결된 요청(진행 중인 요청 또는 과거 요청 포함)이 없는 경우에만 워크플로우 모델을 삭제할 수 있습니다. 완료된 요청은 요청에 대한 버전이 삭제될 때까지 유지되며, 요청을 보기 위해 워크플로우 모델도 사용할 수 있어야 합니다.

 팁:

워크플로우 모델 숨기기의 정보를 고려하여 이 옵션이 더 적절한지 확인하십시오.

워크플로우 모델을 삭제하려면 다음을 수행합니다.

1. 홈 페이지에서 **관리**를 선택합니다.
2. 워크플로우에서 **워크플로우 모델**을 확장합니다.
3. 모델을 선택하고  을 누릅니다.
4. **워크플로우 모델 삭제**를 눌러 삭제를 확인합니다.

## Data Relationship Management Analytics 관리

Oracle Data Relationship Management Analytics에서는 변경 추적, 증가 분석, 요청 모니터링, 워크플로우 모델 성과, 참가자 및 사용자 그룹 성과와 관련된 대시보드를 제공합니다. Data Relationship Management Analytics 대시보드는 다음과 같습니다.

- **변경** - 시간 경과에 따라 Oracle Data Relationship Management 시스템에서 발생한 변경사항의 집계 뷰를 제공합니다. 이 대시보드의 메트릭은 커밋된 요청 및 모든 대화식 변경사항을 기반으로 합니다. 이 대시보드에는 노드 전체에 적용되는 추가, 업데이트, 이동 및 삭제 등의 변경 작업과 계층, 노드 유형, 등록정보 범주 등에 따른 변경 관점을 제공하는 등록정보 변경사항이 포함되어 있습니다. 사용자는 변경 방법, 대화식 또는 워크플로우별 변경 추세를 파악하여 관리 채택 사항을 승인할 수 있습니다. 사용자는 각 변경사항을 컨텍스트에서 드릴하여 트랜잭션 세부정보를 검사하고 오프라인에서 추가로 분석하기 위해 해당 세부정보를 플랫폼 파일로 내보낼 수 있습니다.
- **증가** - 단독 노드 및 공유 노드 수, 총 노드 수, 이전 버전(선형 버전)에서 증가 또는 감소한 총 노드 수, 비선형 버전에서 지난 30일 동안 증가 또는 감소한 총 노드 수를 표시하여 버전 및 계층 구조가 시간 경과에 따라 어떻게 변경되었는지에 대한 분석을 제공합니다.
- **요청** - 미해결 Oracle Data Relationship Governance 요청과 관련 있는 주요 성과 지표를 표시하여 기한이 경과되었거나 기한이 임박한 병목 지점과 요청을 확인할 수 있게 하며, Data Relationship Governance 요청으로 드릴 백하여 요청을 변경할 수 있는 기능을 제공합니다.
- **모델** - 완료된(커밋 또는 거부된) 요청의 과거 성능을 표시하여 Data Relationship Governance 워크플로우 모델 디자인 분석을 제공하며 참가자 동작 추세, 리소스 워크로드 및 Data Relationship Governance로의 드릴 백 기능을 포함합니다. 워크플로우 모델 분석에서는 서비스 레벨 계약, 달성한 자동화 레벨, 주기 시간, 커밋된 리소스, 요청 워크로드, 처리량 및 참가자 참여를 기준으로 모델 성과를 파악하기 위해 각 워크플로우 모델에서 처리하는 완료된 요청의 성과를 보고합니다.
- **보고서** - 사용자 및 그룹 멤버십, 보안 및 활동을 보는 데 사용됩니다. 사용자 역할 지정, 액세스 그룹 지정 보고서, 사용자 로그인 활동 등의 정보가 제공됩니다.
  - **사용자 역할 지정 보고서** - 라이선스를 가진 사용자 유형 수와 함께 역할별 사용자 목록 또는 사용자별 역할 목록을 제공합니다.
  - **액세스 그룹 멤버십 보고서** - 대화식 및 워크플로우 사용자 그룹별 사용자 목록을 제공합니다.
  - **객체 액세스 그룹 권한 부여 보고서** - 사용자와 사용자 그룹을 특정 Data Relationship Management 객체에 매핑합니다.
  - **계층 액세스 그룹 지정 보고서** - 계층의 노드에 대한 사용자와 그룹의 데이터 권한 부여를 제공합니다.
  - **워크플로우 액세스 그룹 지정 보고서** - 워크플로우 모델 단계에 대한 사용자와 그룹의 데이터 권한 부여를 제공합니다.
  - **사용자 로그인 활동 보고서** - 시간 경과에 따른 사용자 로그인 활동에 대한 추세 보고서를 제공합니다.
  - **메타데이터 객체 사용 보고서** - Data Relationship Management 객체(쿼리, 비교, 임포트, 익스포트, 혼합기, 장부)에 대한 도수 분포와 에이징 정보를 제공합니다.

## Data Relationship Management Analytics 액세스

Oracle Data Relationship Management Analytics를 구성하기 전에 다음 태스크를 완료했는지 확인합니다.

- 분석 URL 설정 - Oracle Data Relationship Management에서 Data Relationship Management Analytics로의 링크를 제공합니다. *Oracle Data Relationship Management 설치 가이드*의 "분석 URL 구성"을 참조하십시오.
- 웹 팜 설정 - Data Relationship Management Analytics에서 Data Relationship Management로 드릴 백할 수 있도록 합니다. *Oracle Data Relationship Management 설치 가이드*의 "웹 서버 구성"을 참조하십시오.
- 버전 선행 설정 - 버전 선행을 사용하면 Data Relationship Management Analytics에서 여러 선행과 버전의 변경사항을 집계할 수 있습니다. *Oracle Data Relationship Management 사용자 가이드*의 "버전 등록정보 편집"을 참조하십시오.
- 계층 및 버전 노드 수를 업데이트할 때 Data Relationship Management에서 설정됩니다. 노드 수는 버전을 열거나 저장하거나 닫을 때와 시스템 환경설정에 지정된 시기에 업데이트됩니다. **시스템 환경설정**의 AnalyticsNodeCountUpdateTime을 참조하십시오.
- 계층 그룹 등록정보를 기본 핵심 등록정보 유형으로 설정합니다. Data Relationship Management Analytics에서는 기본 코어 등록정보 유형만 지원됩니다. **등록정보 생성의 6단계**를 참조하십시오.

Data Relationship Management에서 [분석] 링크를 누릅니다.

### 주:

[분석] 링크는 사용자가 분석 사용자, 관리자, 액세스 관리자, 데이터 관리자, 애플리케이션 관리자 역할에 지정된 경우에만 사용할 수 있습니다.

## 환경설정 작업

실행 계획을 생성하기 전에 환경설정을 구성해야 합니다.

환경설정을 지정하려면 다음을 수행합니다.

1.  을 누릅니다.
2. **선택 사항:** 다음을 수행합니다.
  - **배치 크기** - 배치 크기 값을 입력합니다. 모델 분석에 사용됩니다. 기본값은 250MB이며 필요한 경우에만 변경하십시오. 배치 크기가 클수록 메모리 및 데이터베이스 요구사항도 커집니다.
  - **초기 추출 날짜** - 모든 Oracle Data Relationship Management Analytics 태스크에 대한 데이터를 추출할 시작 날짜를 설정합니다.
3. **저장**을 누릅니다.

## 실행 계획 작업

미리 정의된 태스크는 Oracle Data Relationship Management에서 정보를 추출하여 특정 Oracle Data Relationship Management Analytics 대시보드에 반환하며, 여기서 정보를 필터링하고 검토할 수 있습니다. 작업(job)은 대시보드 관련 태스크(task)로 구성됩니다. 하나의 실행 계획에 여러 작업(job)을 포함할 수 있습니다.

실행 계획은 스케줄과 하나 이상의 작업(job) 및 해당 태스크(task)로 구성됩니다. 실행 계획을 매일, 매주 또는 매월 실행되도록 구성할 수 있으며, 단순(지금 실행 또는 미래 날짜/시간에 실행) 또는 Cron(Cron 표현식을 사용하여 스케줄 정보 지정)으로 실행되도록 예약할 수 있습니다. 실행 계획을 편집, 비활성화(사용하지 않을 경우) 및 삭제(더 이상 필요하지 않을 경우)할 수 있습니다.

표 17-1 작업 태스크

작업	태스크
변경 분석	트랜잭션 팩트 테이블 트랜잭션 집계 트랜잭션 등록정보 집계 버전 선형
사용자 활동 보고서	트랜잭션 팩트 테이블
증가 분석	버전 선형 계층 수 버전 수
모델 분석	모델 분석

## 실행 계획 생성

실행 계획을 생성하려면 다음을 수행합니다.

- Oracle Data Relationship Management Analytics 대시보드에서 **설정**을 선택합니다.
- 생성**을 누르고 다음 정보를 입력합니다.
  - 이름** - 실행 계획의 이름을 입력합니다.
  - 스케줄 유형** - 다음 옵션 중 하나를 선택합니다.
    - 단순** - 시작 날짜와 종료 날짜를 지정하려면 사용합니다.
    - Cron** - Cron 표현식을 지정하려면 사용합니다.
  - 스케줄러 기간** - [지금 실행] 또는 [미래]를 선택합니다.
- 다음**을 누릅니다.
- 다음**을 수행합니다.
  - 단순**을 스케줄 유형으로 선택하고 스케줄러 기간에 대해 **지금 실행**을 선택한 경우 다음을 수행합니다.
    - 선택 사항: 자르기 및 로드**를 선택하여 이 작업과 연관된 테이블을 모두 자르고 시스템의 초기 추출 날짜를 기준으로 재로드합니다. 선택하지 않으면 증분 로드가 실행됩니다.
    - 자르기 및 로드를 수행하려면 **확인**을 누릅니다.

- 단순을 스케줄 유형으로 선택하고 스케줄러 시간에 대해 **미래**를 선택한 경우 다음을 수행합니다.
    - a. 실행 계획을 실행할 빈도(매일, 매주 또는 매월)를 선택합니다.
    - b.  을 눌러 시작 날짜 및 시간을 입력합니다.
    - c. **선택사항:**  을 눌러 종료 날짜 및 시간을 입력합니다.
  - **Cron** 스케줄 유형으로 선택한 경우 스케줄러가 실행되는 시기에 대한 Cron 표현식을 입력합니다.
5. 다음을 누릅니다.
  6. 실행 계획에 추가할 작업(job)을 선택합니다. [이동], [모두 이동], [제거] 및 [모두 제거] 버튼을 사용하여 [사용 가능] 목록에서 [선택됨] 목록으로 작업(job)을 이동합니다.
  7. 다음을 누릅니다.
  8. 실행 계획 설정을 검토한 다음 **스케줄 계획**을 누릅니다.

 **주:**

실행 계획을 실행하려면 스케줄러를 시작해야 합니다. 스케줄러를 시작하려면  을 누르고 **시작**을 선택합니다.

9. **확인**을 눌러 일정 계획을 확인합니다.

## 실행 계획 편집

실행 계획을 편집할 때 계획 이름을 제외한 모든 필드를 편집할 수 있습니다.

실행 계획을 편집하려면 다음을 수행합니다.

1. 편집할 계획을 선택합니다.
2.  을 누르고 **실행 계획 생성**의 2~9단계를 수행하여 계획을 변경합니다.

 **주:**

계획 이름은 변경할 수 없습니다. 계획 이름을 변경해야 하는 경우 계획을 삭제하고 새 계획을 생성합니다.

## 실행 계획 비활성화 및 재활성화

실행 계획을 비활성화하면 스케줄링된 미래 계획이 스케줄러에서 모두 제거되며 계획이 [비활성 계획] 탭으로 이동됩니다. 계획을 재활성화하려면 [비활성 계획] 탭에서 계획을 편집한 후 예약합니다.

실행 계획을 비활성화하려면 다음을 수행합니다.

-  을 선택하고 비활성화할 계획을 선택합니다.
- 계획 이름 옆에 있는  을 누릅니다.  
실행 계획을 재활성화하려면 다음을 수행합니다.
  -  을 선택하고 재활성화할 계획을 선택합니다.
  -  을 누르고 **실행 계획 생성**의 2~9단계를 수행하여 계획을 변경합니다.

**주:**

계획 이름은 변경할 수 없습니다. 계획 이름을 변경해야 하는 경우 계획을 삭제하고 새 계획을 생성합니다.

## 실행 계획 삭제

실행 계획을 삭제하려면 다음을 수행합니다.

- 삭제할 실행 계획을 선택합니다.
- 계획 이름 옆에 있는  을 누릅니다.
- 확인**을 눌러 삭제를 확인합니다.

## 활동 보기

[최근 활동] 섹션에서 실행된 실행 계획의 결과를 볼 수 있습니다. 실행 계획의 시작 시간과 종료 시간, 실행 기간, 처리된 레코드 수 및 실행 상태를 볼 수 있습니다. 동일한 계획에서 여러 작업(job)을 예약하고 다른 작업(job)에서 이미 실행한 태스크(task)가 둘 이상의 작업(job)에 포함된 경우 실행 시 후행 작업(job)에서 해당 태스크(task)를 건너뛰며 실행 계획 결과에 [중복 항목 건너뛴]으로 표시됩니다.

실행된 실행 계획의 결과를 보려면 다음을 수행합니다.

-  을 누르거나  을 누릅니다.
- 계획 이름 왼쪽에 있는 화살표를 눌러 보려는 실행 계획을 확장합니다. 계획 내의 작업(job)을 확장하면 관련 태스크(task)를 검토할 수 있습니다.
- 선택 사항:** 필터 모음을 누르고 필터 옵션을 설정합니다.
  - 시간 프레임** - 계획 활동을 표시할 일수를 입력합니다. 예를 들어 2를 입력하면 최근 2일의 계획 활동이 표시됩니다.
  - 이름** - 모두를 선택하거나 결과에 포함할 실행 계획 이름을 선택합니다.
  - 상태** - 모두를 선택하거나 결과에 포함할 실행 계획 상태를 선택합니다. 계획 상태는 [완료], [부분 실패], [실패] 및 [처리 중]입니다.

 주:

상태 필터 조건은 실행 계획 상태에만 적용되고 작업(job) 또는 태스크(task) 상태에는 적용되지 않습니다.

## 외부 워크플로우 애플리케이션 통합

외부 워크플로우 애플리케이션을 사용하여 외부 소스에서 제안된 Oracle Data Relationship Management 변경사항을 처리할 수 있습니다. 웹 서비스 API는 외부 워크플로우 프로세스 중에 단일 작업 단위로 변경사항을 검증 및 커밋하기 위해 여러 변경사항을 함께 그룹화할 수 있는 외부 요청 인터페이스를 제공합니다. API 사용자가 외부 요청에 참여하려면 워크플로우 사용자 역할이 있어야 합니다. 이 요청 인터페이스는 일반이며 워크플로우 모델, 워크플로우 태스크 또는 웹 클라이언트의 [태스크 목록] 페이지 사용을 지원하지 않습니다. 이 일반 외부 요청은 [요청 내역]에 기록되며 여기서만 액세스할 수 있습니다.

외부 요청의 API 지원에 대한 자세한 내용은 "Oracle Data Relationship Management API Reference"를 참조하십시오.

### 외부 요청

다음과 같은 외부 요청을 생성할 수 있습니다.

- 계층 구조 추가
- 노드 추가
- 노드 삽입 및 이동
- 노드 활성화, 비활성화 및 제거
- 등록정보 업데이트
- 등록정보 값 제거

변경사항을 버전에 즉시 커밋하지 않고 승인을 위해 외부 요청을 초안 상태로 저장한 다음 Oracle Data Relationship Management 버전과 비교하여 검증할 수 있습니다. 이 승인 보류 상태의 외부 요청은 여러 사용자가 각기 다른 시간에 업데이트하고 필요에 따라 재검증할 수 있습니다. 요청이 승인되면 요청의 트랜잭션이 Data Relationship Management 버전에 커밋됩니다.

#### 주:

외부 요청이 승인된 후에는 요청을 수정할 수 없으며, 관련 버전을 삭제할 때까지 요청을 삭제할 수도 없습니다.

외부 요청은 다음과 같은 요소로 구성됩니다.

- 대상 Data Relationship Management 버전
- 요청 소유자 - 적합한 Data Relationship Management 사용자 ID
- 사용자정의 워크플로우 ID - 워크플로우 애플리케이션의 요청 식별자
- 사용자정의 워크플로우 레이블 - 워크플로우 애플리케이션의 요청에 대한 간단한 설명
- 사용자정의 워크플로우 상태 - 워크플로우 애플리케이션의 요청 상태 관리
- 사용자정의 워크플로우 정보 - 워크플로우 애플리케이션에 필요한 추가 정보 저장

- 요청 설명 - 요청에 대한 노트
- 생성자 - 초기 요청을 생성한 사용자
- 생성 날짜 - 요청이 생성된 날짜
- 업데이트자 - 요청을 최종 업데이트한 사용자
- 업데이트 날짜 - 요청이 최종 업데이트된 날짜
- 승인자 - 요청을 승인한 사용자
- 승인 날짜 - 요청이 승인된 날짜
- 검증됨 플래그 - 요청이 최종 업데이트된 이후 검증되었는지 여부를 나타냄
- 승인됨 플래그 - 요청이 승인되었는지 여부를 나타냄
- 검증 또는 승인 작업 중에 요청의 작업에만 적용되어야 하는 추가 बै치 검증
- 현재 요청에 대한 계층 구조 및 노드에 영향을 주는 작업 항목 목록

# Data Relationship Management 메타데이터 마이그레이션

Oracle Data Relationship Management 마이그레이션 유틸리티를 사용하면 애플리케이션 관리자가 Data Relationship Management 애플리케이션 간에 메타데이터 객체 유형을 이동할 수 있습니다.

마이그레이션 유틸리티에서 다음 작업을 할 수 있습니다.

- Data Relationship Management 애플리케이션에서 XML 파일로 메타데이터 객체 유형을 추출하고 결과에서 HTML 보고서를 생성합니다.
- XML 파일에서 Data Relationship Management 애플리케이션으로 메타데이터를 로드합니다.
- 두 소스 간의 메타데이터 차이를 비교하고 차이가 포함된 XML 파일을 생성한 다음 결과에서 HTML 보고서를 생성합니다.
- XML 파일의 메타데이터를 보고 파일에서 HTML 보고서를 생성합니다.

다음 유형의 메타데이터를 추출, 로드, 비교 및 조회할 수 있습니다.

- 등록정보 정의
- 등록정보 범주
- 검증
- 노드 유형
- 그래픽 기호
- 노드 액세스 그룹
- 계층 그룹
- 쿼리(표준, 시스템 및 사용자정의)
- 비교(표준, 시스템 및 사용자정의)
- 도메인
- 버전 변수(표준, 시스템 및 사용자정의)
- 익스포트(표준, 시스템 및 사용자정의)
- 익스포트 장부(표준, 시스템 및 사용자정의)
- 임포트(표준, 시스템 및 사용자정의)
- 혼합기(표준, 시스템 및 사용자정의)
- 시스템 환경설정
- 외부 연결(표준, 시스템 및 사용자정의)

외부 연결은 연결 이름만 표시합니다. 객체 액세스 그룹 이름 접두어는 추가되지 않습니다.

 주:

마이그레이션 로드 및 추출 시 연결 문자열, 사용자 ID 및 비밀번호는 함께 마이그레이션되지 않습니다.

- 객체 액세스 그룹
- 워크플로우 태스크
- 워크플로우 모델

### 핵심 등록정보 구성 및 설정 마이그레이션

메타데이터 마이그레이션 유틸리티를 사용하여 동일한 릴리스의 Data Relationship Management 인스턴스 간에 마이그레이션할 수 있는 핵심 등록정보 구성 및 설정은 다음과 같습니다.

- Core.DefaultDisplayBy [기본 표시 등록정보]
- Core.DefaultPasteProps [기본 붙여넣기 등록정보]
- Core.DefaultSynchBy [기본 일치 기준]
- Core.EnableSharedNodes [공유 노드 사용]
- Core.HierarchyNodeType [계층 노드 유형]
- Core.IDLengthLeafProp [ID 길이 리프 등록정보]
- Core.IDLengthLimbProp [ID 길이 림 등록정보]
- Core.PrefillLeafProp [리프 등록정보 사전 채우기]
- Core.PrefillLimbProp [림 등록정보 사전 채우기]
- Core.SortOrder [정렬 순서]
- Core.StandardHierSort [표준 계층 정렬]

## 마이그레이션 유틸리티 열기

기본적으로 마이그레이션 유틸리티는 다음 위치에 설치됩니다.

```
MIDDLEWARE_HOME\EPMSysstem11R1\products\DataRelationshipManagement\client
```

마이그레이션 유틸리티를 열려면 **Data Relationship Management 마이그레이션 유틸리티**를 두 번 누릅니다.

## 메타데이터 추출

Oracle Data Relationship Management 애플리케이션에서 추출할 메타데이터 유형을 선택할 수 있습니다. 정보를 XML 파일로 추출한 다음 파일을 보거나, 다른 Data Relationship Management 애플리케이션으로 로드하거나, 다른 XML 파일과 비교하거나, 다른 Data Relationship Management 애플리케이션과 비교할 수 있습니다. 이 파일을 백업, 저장 및 감사 목적으로 사용할 수도 있습니다.

생성된 XML 파일의 정보에서 보고서를 생성할 수 있습니다.

Data Relationship Management 애플리케이션에서 메타데이터를 추출하려면 다음을 수행합니다.

1. 기본 메뉴에서 **추출**을 누릅니다.
2. Data Relationship Management 연결 정보를 입력하고 **로그인**을 누릅니다.
3. 추출할 객체 유형 또는 객체를 선택하고 **다음**을 누릅니다.

 **주:**

계층 트리에서 더하기 기호를 눌러 객체를 표시합니다. 객체 유형의 확인란을 선택하여 객체 유형 및 해당 객체를 모두 선택하거나 추출할 객체의 확인란을 선택합니다. 객체 이름을 눌러 새 창에 객체 유형 정의를 표시합니다.

4. **선택 사항:** 찾기를 눌러 메타데이터 객체 유형 또는 객체를 검색합니다.

 **주:**

입력한 텍스트가 포함된 모든 객체 유형이 반환됩니다. 결과에서 특정 객체로 이동하려면 [이동] 링크를 누릅니다.

5. 요약 정보를 검토합니다.

 **주:**

마이그레이션 유틸리티는 종속성이 있는 객체 유형에 대해 추가 확인을 수행합니다. 예를 들어 익스포트가 등록정보 정의에 따라 달라지거나 등록정보 정의에서 다른 등록정보 정의를 참조할 수 있습니다. 요약에 누락된 종속성이 있는 경우 포함할 특정 종속성을 선택할 수 있습니다. 제외된 모든 종속성을 포함하거나 모든 종속성을 포함할 수 있습니다.

 **주:**

페이지 크기를 늘리면 페이지에서 보려는 객체 유형 수를 정의할 수 있습니다.

6. **선택 사항:** 이 추출에 대한 메타데이터 세부정보를 입력합니다.  
다음 정보를 입력할 수 있습니다.
  - **제목** - 최대 255자
  - **목적** - 형식 지정 메모
  - **사용** - 형식 지정 메모
  - **애플리케이션 버전** - 최대 20자
  - **파일 버전** - 최대 20자
7. **추출 실행**을 누릅니다.
8. 다음 중 하나를 수행합니다.
  - **메타데이터 파일 다운로드**를 눌러 XML 파일을 열거나 저장합니다.

- **메타데이터 파일 보기**를 눌러 XML 파일 세부정보를 봅니다.
- **메타데이터 파일 로드**를 눌러 XML 파일을 Data Relationship Management 애플리케이션으로 로드합니다. 자세한 내용은 [메타데이터 로드](#)을 참조하십시오.
- **메타데이터 파일에 대한 보고서 생성**을 눌러 XML 파일에서 보고서를 생성합니다. 자세한 내용은 [보고서 생성](#)를 참조하십시오.

## 메타데이터 로드

Oracle Data Relationship Management XML 형식이 포함된 파일만 Data Relationship Management 애플리케이션으로 로드할 수 있습니다. 로그 파일은 로드가 수행된 후 생성되며 감사, 정보, 경고, 오류 메시지 등의 데이터 심각도를 표시합니다.

### 주:

메타데이터 파일을 로드하기 전에 이전 구성으로 되돌리려는 경우를 위해 기존 메타데이터를 추출하는 것이 좋습니다. 특히 마이그레이션 파일을 프로덕션 환경으로 로드하는 경우 메타데이터를 로드하기 전에 데이터베이스를 백업하는 것이 좋습니다.

XML 파일에서 Data Relationship Management 애플리케이션으로 메타데이터를 로드하려면 다음을 수행합니다.

1. 기본 메뉴에서 **로드**를 누릅니다.
2. **찾아보기**를 누르고 로드할 XML 파일을 선택한 다음 **업로드**를 누릅니다.

### 주:

마이그레이션 파일은 UTF-8 인코딩이어야 합니다.

3. 업로드된 파일 정보를 검토하고 **다음**을 누릅니다.
4. Data Relationship Management 연결 정보를 입력하고 **로그인**을 누릅니다.
5. 로드할 객체 유형 또는 객체를 선택하고 **다음**을 누릅니다.

### 주:

계층 트리에서 더하기 기호를 눌러 객체를 표시합니다. 객체 유형의 확인란을 선택하여 객체 유형 및 해당 객체를 모두 선택하거나 로드할 객체의 확인란을 선택합니다. 객체 이름을 눌러 새 창에 객체 유형 정의를 표시합니다.

6. 요약 정보를 검토하고 **다음**을 누릅니다.

 주:

페이지 크기를 사용하면 페이지에서 보려는 객체 유형 수를 정의할 수 있습니다.

7. **선택사항:** 오류가 발생한 경우에도 로드를 계속하려면 **오류 후 로드 계속**을 선택합니다.
8. **로드 실행**을 누릅니다.
9. 로드 결과를 검토합니다.

감사, 정보, 경고, 오류 등 표시할 세부정보 심각도를 선택하여 로그 파일 뷰를 변경할 수 있습니다. 로그 파일을 저장하려면 **다운로드**를 누릅니다.

 주:

열 머리글 링크를 사용하여 임의 열을 기준으로 로그 항목을 정렬할 수 있습니다.

## 메타데이터 비교

두 메타데이터 소스를 비교할 수 있습니다. 두 Oracle Data Relationship Management 애플리케이션 간, 두 XML 파일 간 또는 Data Relationship Management 애플리케이션과 XML 파일 간에 메타데이터 차이를 비교할 수 있습니다. 두 메타데이터 소스 간의 차이가 포함된 XML 파일을 생성할 수 있습니다. 결과를 사용하여 데이터를 복원하거나, 권한 없는 변경을 실행취소하거나, 잘못된 객체 유형 구성을 찾을 수 있습니다.

생성된 XML 파일의 정보에서 보고서를 생성할 수 있습니다.

메타데이터를 비교하려면 다음을 수행합니다.

1. 기본 메뉴에서 **차이**를 누릅니다.
2. **소스 #1** 드롭다운 목록에서 소스 유형(서버 연결 또는 XML 파일)을 선택합니다.
3. 다음 중 하나를 수행합니다.
  - **서버 연결**을 선택한 경우 Data Relationship Management 연결 정보를 입력하고 **로그인**을 누릅니다.
  - **XML 파일**을 선택한 경우 **찾아보기**를 누르고 비교에 사용할 XML 파일을 선택한 다음 **업로드**를 누릅니다.
4. 파일을 업로드한 경우 업로드된 파일 정보를 검토하고 **다음**을 누릅니다. 그렇지 않으면 다음 단계로 건너뜁니다.
5. 소스 #2에 대해 2-4단계를 반복합니다.
6. **다음**을 누릅니다.
7. 다음 작업을 사용하여 차이 파일에 포함할 객체 유형을 선택합니다.
  - 필터를 선택합니다.
  - >를 눌러 소스 #1에서 객체 유형을 선택합니다.
  - <를 눌러 소스 #2에서 객체 유형을 선택합니다.
  - X를 눌러 객체 유형을 선택 취소합니다.
  - 왼쪽 열 머리글을 눌러 선택한 필터를 기준으로 소스 #1에서 모든 객체를 선택합니다.

- 오른쪽 열 머리글을 눌러 선택한 필터를 기준으로 소스 #2에서 모든 객체를 선택합니다.
- 가운데 열 머리글을 눌러 선택한 필터를 기준으로 모든 객체를 선택 취소합니다.
- 비교 결과 맨위에 있는 페이지 링크를 눌러 다른 페이지로 전환합니다.

 주:

페이지 크기를 사용하면 페이지에서 보려는 객체 유형 수를 정의할 수 있습니다.

8. 차이 파일 생성을 누릅니다.
9. 다음 중 하나를 수행합니다.
  - 메타데이터 차이 파일 다운로드를 눌러 XML 파일을 열거나 저장합니다.
  - 메타데이터 차이 파일 보기를 눌러 XML 파일 세부정보를 봅니다.
  - 메타데이터 차이 파일 로드를 눌러 파일을 Data Relationship Management 애플리케이션으로 로드합니다. 자세한 내용은 [메타데이터 로드](#)를 참조하십시오.
  - 메타데이터 파일에 대한 보고서 생성을 눌러 XML 파일에서 보고서를 생성합니다. 자세한 내용은 [보고서 생성](#)를 참조하십시오.

## 메타데이터 보기

메타데이터 파일을 보고 파일의 정보에서 보고서를 생성할 수 있습니다.

XML 파일의 메타데이터를 보려면 다음을 수행합니다.

1. 기본 메뉴에서 **파일 보기**를 누릅니다.
2. **찾아보기**를 누르고 보려는 XML 파일을 선택한 다음 **업로드**를 누릅니다.
3. 업로드된 파일 정보를 검토하고 **다음**을 누릅니다.
4. 계층 트리에서 더하기 기호를 눌러 메타데이터 객체를 봅니다.
5. **선택사항: 찾기**를 눌러 파일에서 항목을 검색합니다.

 주:

텍스트가 포함된 모든 객체 유형이 반환됩니다. 결과에서 특정 객체로 이동하려면 [\[이동\]](#) 링크를 누릅니다.

6. **선택 사항: 보고서** 탭을 눌러 파일에서 HTML 보고서를 생성합니다.

## 메타데이터 파일 제한 사항

마이그레이션 유틸리티에서 업로드된 파일의 기본 제한은 4MB입니다. 마이그레이션 유틸리티를 사용하여 큰 메타데이터 파일을 로드하거나 보는 경우 파일 크기가 구성된 제한을 초과하면 다음 오류가 발생할 수 있습니다.

"예상치 못한 오류입니다. 요청을 처리하는 중 예상치 못한 오류가 발생했습니다. 최대 요청 길이를 초과했습니다."

큰 파일 크기 구성에 대한 자세한 내용은 *Oracle Data Relationship Management 설치 가이드*의 "마이그레이션 유틸리티 구성"을 참조하십시오.

## 보고서 생성

추출 후에 생성된 XML 파일, 차이 보고서 및 보고 있는 메타데이터 파일에서 HTML 보고서를 생성할 수 있습니다.

HTML 보고서를 생성하려면 다음을 수행합니다.

1. 다음 중 하나를 수행합니다.
  - 메타데이터를 추출하거나 차이 보고서를 생성한 후 **메타데이터 파일에 대한 보고서 생성**을 누릅니다.
  - 메타데이터 파일을 본 후 **보고서**를 누릅니다.
2. 다음 중 하나를 수행합니다.
  - **보고서 보기**를 눌러 보고서를 표시합니다.
  - **보고서 다운로드**를 눌러 보고서를 저장합니다.